



**PENGEMBANGAN APLIKASI READO: APLIKASI
PEMBELAJARAN BAHASA INGGRIS BAGI ANAK PADA
KEGIATAN SOSIAL KOMUNITAS KATALIS PENDIDIKAN
DENGAN METODE *DESIGN SPRINT***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rachmalia Dewi
NIM: 165150201111236



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2020

PENGESAHAN

PENGEMBANGAN APLIKASI READO: APLIKASI PEMBELAJARAN BAHASA INGGRIS
BAGI ANAK PADA KEGIATAN SOSIAL KOMUNITAS KATALIS PENDIDIKAN DENGAN
METODE *DESIGN SPRINT*

SKRIPSI

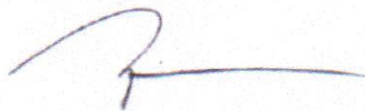
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rachmalia Dewi
NIM: 165150201111236

Skripsi ini telah diuji dan dinyatakan lulus pada
4 Mei 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Agi Putra Kharisma, S.T., M.T.
NIK: 201304 860430 1 001

Dosen Pembimbing 2



Candra Dewi, S.Kom, M.Sc.
NIP: 19771114 200312 2 001

Mengetahui

Ketua Jurusan Teknik Informatika



Astoro Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

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, 4 Mei 2020



Rachmalia Dewi

NIM: 165150201111236

PRAKATA

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Aplikasi Reado: Aplikasi Pembelajaran Bahasa Inggris Bagi Anak pada Kegiatan Sosial Komunitas Katalis Pendidikan dengan Metode Design Sprint” ini dapat terselesaikan.

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

1. Bapak Agi Putra Kharisma, S.T., M.T. dan Ibu Candra Dewi, S.Kom, M.Sc. selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika,
4. Bapak Reza Andria Siregar, S.T., M.Kom. selaku dosen Penasihat Akademik yang selalu memberikan nasihat kepada penulis selama menempuh masa studi,
5. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikan skripsi ini,
6. Seluruh civitas akademik Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

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

Malang, 4 Mei 2020

Rachmalia Dewi

rachmaliadew@student.ub.ac.id

ABSTRAK

Rachmalia Dewi, Pengembangan Aplikasi Reado: Aplikasi Pembelajaran Bahasa Inggris Bagi Anak pada Kegiatan Sosial Komunitas Katalis Pendidikan dengan Metode *Design Sprint*

Pembimbing: Agi Putra Kharisma, S.T., M.T. dan Candra Dewi, S.Kom., M.Sc.

Indonesia adalah salah satu negara yang menerapkan penggunaan bahasa Inggris sebagai mata pelajaran pada pendidikan formal. Namun, nilai dari indeks kecakapan bahasa Inggris di Indonesia menempati peringkat 61 dari 100 negara di dunia. Indonesia memiliki nilai indeks yaitu 50,06 dan berada di bawah nilai rata-rata yaitu 53, sehingga Indonesia ditempatkan dalam kategori rendah untuk kecakapan bahasa Inggris dari 100 negara. Hal ini dikarenakan pembelajaran bahasa Inggris yang saat ini diterapkan masih belum dapat berjalan dengan efektif dan efisien, sehingga diperlukan adanya perbaikan untuk meningkatkan pembelajaran dalam bahasa Inggris. Pembelajaran bahasa Inggris bagi anak sangatlah memiliki peluang yang besar untuk kemampuan anak kedepannya melalui komunikasi. Pembelajaran bahasa Inggris dapat dilakukan dengan menggunakan strategi pengajaran salah satunya yaitu pembelajaran melalui permainan edukatif dengan menggunakan perangkat elektronik untuk pembelajaran kata dalam bahasa Inggris. Selain itu, berdasarkan data statistik menunjukkan bahwa sebagian besar masyarakat yaitu menggunakan aplikasi perangkat bergerak *mobile* berbasis Android. Maka dari itu, dalam penelitian skripsi ini dilakukan pengembangan permainan edukatif bagi anak melalui perangkat bergerak *mobile* berbasis Android yang dinamakan dengan aplikasi Reado. Pengembangan aplikasi Reado dilakukan dengan menggunakan metode *design sprint* dan hasil pengembangan ini diujikan pada anak-anak dari kegiatan sosial Komunitas Katalis Pendidikan yang terdiri dari 22 anak. Dari penerapan metode *design sprint* pada pengembangan aplikasi Reado, berdasarkan hasil pengujian *black box* yaitu 10 pengujian dari kebutuhan fungsional aplikasi Reado telah lulus dan sudah terverifikasi sehingga dinyatakan bahwa aplikasi Reado sudah memenuhi kebutuhan dari pengguna. Selain itu, berdasarkan hasil pengujian *usability* yang dilakukan pada tingkatan Sekolah Dasar (SD) dari setiap masing-masing kelas yaitu kelas 1 sampai 6, didapatkan hasil yaitu nilai dari aspek efisiensi memiliki rata-rata 93,5% yaitu “baik”, nilai dari aspek efektivitas adalah 0,06 yaitu “sangat baik” dan nilai dari aspek kepuasan pengguna adalah 80,37 yaitu “baik”.

Kata kunci: *design sprint*, aplikasi perangkat bergerak, Android, permainan edukatif, pembelajaran bahasa inggris, anak-anak



ABSTRACT

Indonesia is one of the countries that applied English as a subject in formal education. However, the value of the English proficiency index in Indonesia listed at the 61st position of 100 countries in the world. Indonesia has an index value of 50.06 that is below the average value of 53 and Indonesia is placed in the low category for English proficiency of 100 countries. This is because learning English that is currently being implemented is still not able to run effectively and efficiently, so it is necessary to make improvements to improve learning in English. Learning English for children really has great opportunities for children's abilities in the future through communication. Learning English can be done by using teaching strategy which is learning through educational games using electronic devices for learning words in English. In addition, based on statistical data shows that the majority of the public is using an Android-based mobile application. Therefore in this research, the development of educational game for children through Android-based mobile devices is called Reado application. The development of the Reado application was carried out by using the design sprint method and the results of this development were tested on children from the social activities of the Katalis Pendidikan community which consisted of 22 children. From the application of the design sprint method to the development of the Reado application, based on the result of the black box testing that is 10 tests of the functional requirements of the Reado application have passed and have been verified, so it is stated that the Reado application has fulfill user's specific needs. In addition, based on the results of usability testing that conducted at the elementary school level of each class that is grades 1 to 6, the results obtained that the value of the efficiency aspect has an average of 93.5% which is "good", the value of the effectiveness aspect is 0.06 which is "very good" and the value of the user satisfaction aspect is 80.37 which is "good".

Keywords: design sprint, mobile application, Android, educational game, english learning, children



DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	ii
PRAKATA.....	vi
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xvii
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 Perangkat Bergerak.....	7
2.3 Pengembangan Aplikasi Perangkat Bergerak dengan <i>Design Sprint</i>	8
2.4 Android.....	10
2.5 Pengujian Perangkat Lunak.....	11
2.5.1 Pengujian <i>Black Box</i>	11
2.5.1.1 <i>Equivalence Partitioning</i>	11
2.5.2 Pengujian <i>Usability</i>	12
2.5.2.1 Efisiensi.....	13
2.5.2.2 Efektivitas.....	14
2.5.2.3 Kepuasan Pengguna.....	14



BAB 3 METODOLOGI PENELITIAN.....	18
3.1 Tipe Penelitian	18
3.2 Strategi dan Rancangan Penelitian.....	18
3.2.1 Strategi atau Metode Secara Umum	18
3.2.2 Subjek Penelitian.....	20
3.2.3 Lokasi Penelitian.....	22
3.2.4 Teknik Pengumpulan Data.....	22
3.2.4.1 Observasi.....	22
3.2.4.2 Wawancara.....	23
3.2.4.3 Kuesioner.....	23
3.2.5 Desain Penelitian.....	23
3.2.5.1 Analisis Permasalahan.....	23
3.2.5.2 Studi Literatur.....	24
3.2.5.3 Pengumpulan Data.....	24
3.2.5.4 Analisis Kebutuhan dan Perancangan.....	24
3.2.5.5 Implementasi.....	24
3.2.5.6 Pengujian.....	26
3.2.5.7 Penarikan Kesimpulan dan Saran.....	26
BAB 4 ANALISIS KEBUTUHAN.....	27
4.1 Identifikasi Aktor.....	27
4.2 Identifikasi Kebutuhan Perangkat Lunak.....	27
4.2.1 Menentukan <i>Long-Term Goal</i>	27
4.2.2 Menentukan <i>Sprint Questions</i>	28
4.2.3 Membuat <i>Project Map</i>	28
4.2.4 Melakukan <i>Ask The Expert</i>	30
4.2.5 Membuat Daftar <i>How Might We</i> (HMW).....	31
4.2.6 Melakukan <i>Affinity Mapping</i>	33
4.2.7 Melakukan <i>Zen Voting</i>	34
4.2.8 Menentukan Metrik-Metrik.....	35
4.3 Analisis Kebutuhan Perangkat Lunak.....	36
4.3.1 Kebutuhan Fungsional.....	37
4.3.2 Kebutuhan Non Fungsional.....	38



4.4 Pemodelan Kebutuhan	39
4.4.1 Diagram <i>Use Case</i>	39
4.4.2 Skenario <i>Use Case</i>	41
BAB 5 PERANCANGAN DAN IMPLEMENTASI	50
5.1 Perancangan Perangkat Lunak	50
5.1.1 Perancangan Arsitektur Aplikasi	50
5.1.2 Perancangan Diagram	51
5.1.2.1 Diagram <i>Sequence</i>	51
5.1.2.2 Diagram <i>Class</i>	63
5.1.3 Perancangan Basis Data	65
5.1.4 Perancangan Algoritma	66
5.1.5 Perancangan Antarmuka	78
5.1.5.1 Melakukan <i>Lightning Demos</i>	79
5.1.5.2 Membuat Sketsa	79
5.1.5.3 Melakukan <i>Sticky Decision</i>	80
5.1.5.4 Membuat <i>Storyboard</i>	80
5.1.5.5 Membuat Prototipe	93
5.1.5.6 Melakukan Pengujian Prototipe	99
5.2 Implementasi Perangkat Lunak	99
5.2.1 Spesifikasi Perangkat Keras	99
5.2.2 Spesifikasi Perangkat Lunak	100
5.2.3 Batasan Implementasi	100
5.2.4 Implementasi Basis Data	101
5.2.5 Implementasi Algoritma	102
5.2.6 Implementasi Antarmuka	126
BAB 6 PENGUJIAN	132
6.1 Pengujian <i>Black Box</i>	132
6.2 Pengujian <i>Usability</i>	142
6.2.1 Data Pengujian	144
6.2.2 Aspek Efisiensi	148
6.2.3 Aspek Efektivitas	151
6.2.4 Aspek Kepuasan Pengguna	154



BAB 7 PENUTUP.....

157

7.1 Kesimpulan.....

157

7.2 Saran.....

158

DAFTAR REFERENSI.....

159

LAMPIRAN.....

161



DAFTAR TABEL

Tabel 3.1 Jumlah Anak-Anak dari Setiap Kelas	21
Tabel 3.2 Jumlah Anak-Anak yang Akan Diujikan dari Setiap Kelas.....	22
Tabel 4.1 Aktor dari Aplikasi Reado	27
Tabel 4.2 <i>Sprint Questions</i> dari Pengembangan Aplikasi Reado	28
Tabel 4.3 Hasil Wawancara dari Pengajar.....	30
Tabel 4.4 Daftar <i>How Might We</i> dari Tim Pengembang Aplikasi Reado	32
Tabel 4.5 Daftar <i>How Might We</i> dari Hasil <i>Affinity Mapping</i>	34
Tabel 4.6 Daftar <i>How Might We</i> dari Hasil <i>Zen Voting</i>	35
Tabel 4.7 Hasil Penentuan Metrik dengan Menggunakan <i>HEART Metric</i>	36
Tabel 4.8 Kebutuhan Fungsional dari Aplikasi Reado	37
Tabel 4.9 Skenario <i>Use Case</i> Mendeteksi Kata	42
Tabel 4.10 Skenario <i>Use Case</i> Menampilkan Kata-Kata dari Hasil Deteksi Kata ..	43
Tabel 4.11 Skenario <i>Use Case</i> Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	43
Tabel 4.12 Skenario <i>Use Case</i> Menentukan Tahapan Permainan	44
Tabel 4.13 Skenario <i>Use Case</i> Menentukan <i>Level</i> Permainan	45
Tabel 4.14 Skenario <i>Use Case</i> Menampilkan Permainan Tahapan Latihan.....	45
Tabel 4.15 Skenario <i>Use Case</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	46
Tabel 4.16 Skenario <i>Use Case</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	47
Tabel 4.17 Skenario <i>Use Case</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	48
Tabel 4.18 Skenario <i>Use Case</i> Menyediakan Mekanisme untuk Jawaban dari Permainan	48
Tabel 5.1 Perancangan Basis Data dari Entitas <i>Quizzes</i>	65
Tabel 5.2 Perancangan Basis Data dari Entitas <i>Stages</i>	65
Tabel 5.3 Perancangan Algoritma Mendeteksi Kata	67
Tabel 5.4 Perancangan Algoritma Menampilkan Kata-Kata dari Hasil Deteksi Kata	68
Tabel 5.5 Perancangan Algoritma Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	69



Tabel 5.6 Perancangan Algoritma Menentukan Tahapan Permainan.....	69
Tabel 5.7 Perancangan Algoritma Menentukan Level/ Permainan.....	71
Tabel 5.8 Perancangan Algoritma Menampilkan Permainan Tahapan Latihan	72
Tabel 5.9 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	73
Tabel 5.10 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	74
Tabel 5.11 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	75
Tabel 5.12 Perancangan Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan	76
Tabel 5.13 Aplikasi Kompetitor dari Pengembangan Aplikasi Reado	79
Tabel 5.14 Penjelasan Tampilan Aplikasi untuk Logo Aplikasi Reado	81
Tabel 5.15 Penjelasan Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado ..	81
Tabel 5.16 Penjelasan Tampilan Aplikasi untuk Halaman Deteksi Kata	82
Tabel 5.17 Penjelasan Tampilan Aplikasi untuk Halaman Kata	83
Tabel 5.18 Penjelasan Tampilan Aplikasi untuk Halaman Alfabet.....	83
Tabel 5.19 Penjelasan Tampilan Aplikasi untuk Halaman Tahapan Permainan ...	84
Tabel 5.20 Penjelasan Tampilan Aplikasi untuk Halaman Level Permainan.....	85
Tabel 5.21 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Latihan.....	87
Tabel 5.22 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar.....	87
Tabel 5.23 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara.....	88
Tabel 5.24 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Video	90
Tabel 5.25 Spesifikasi Perangkat Keras Laptop.....	99
Tabel 5.26 Spesifikasi Perangkat Keras <i>Smartphone</i>	99
Tabel 5.27 Spesifikasi Perangkat Lunak	100
Tabel 5.28 Implementasi Basis Data Aplikasi Reado.....	101
Tabel 5.29 Implementasi Algoritma Mendeteksi Kata.....	102
Tabel 5.30 Implementasi Algoritma Menampilkan Kata	105
Tabel 5.31 Implementasi Algoritma Menampilkan Alfabet.....	107



Tabel 5.32 Implementasi Algoritma Menentukan Tahapan Permainan.....	109
Tabel 5.33 Implementasi Algoritma Menentukan Level Permainan.....	113
Tabel 5.34 Implementasi Algoritma Menampilkan Permainan Tahapan Latihan	116
Tabel 5.35 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	117
Tabel 5.36 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	118
Tabel 5.37 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	120
Tabel 5.38 Implementasi Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan	123
Tabel 6.1 Pengujian <i>Black Box</i> dengan Menggunakan Teknik <i>Equivalence Partitioning</i> pada Aplikasi Reado	133
Tabel 6.2 Kebutuhan Fungsional dari Hasil Pengujian <i>Black Box</i>	142
Tabel 6.3 Subjek Penelitian dari Pengujian Aplikasi Reado	143
Tabel 6.4 Tugas Pengguna untuk Pengujian Aplikasi Reado	143
Tabel 6.5 Keberhasilan Pengguna Dalam Menyelesaikan Tugas	144
Tabel 6.6 Waktu yang Dibutuhkan untuk Menyelesaikan Tugas	145
Tabel 6.7 Jumlah Kesalahan Pengguna dari Setiap Tugas	146
Tabel 6.8 Hasil Kuesioner	148
Tabel 6.9 Hasil Nilai Efisiensi Relatif Keseluruhan dari Setiap Kelas	150
Tabel 6.10 Total Kesempatan untuk Menyelesaikan Tugas Pengguna	151
Tabel 6.11 Hasil Nilai Tingkat Kesalahan dari Setiap Kelas	154
Tabel 6.12 Nilai SUS	155
Tabel 6.13 Hasil Nilai SUS dari Setiap Kelas	156



DAFTAR GAMBAR

Gambar 2.1 Tingkat Kepercayaan Aplikasi.....	14
Gambar 2.2 Kuesioner <i>System Usability Scale</i>	15
Gambar 2.3 Kuesioner <i>System Usability Scale</i> (Terjemahan).....	16
Gambar 2.4 <i>Adjective Rating</i> Berdasarkan Skor SUS (<i>System Usability Scale</i>).....	17
Gambar 3.1 Alur Penelitian Metodologi Pengembangan Sistem	25
Gambar 4.1 <i>Project Map</i> dari <i>User Journey</i> Anak-anak dalam Mempelajari Kata Bahasa Inggris.....	29
Gambar 4.2 Diagram <i>Use Case</i> dari Aplikasi Reado.....	41
Gambar 5.1 Skema Perancangan Arsitektur Aplikasi Reado	50
Gambar 5.2 Diagram <i>Sequence</i> Mendeteksi Kata.....	51
Gambar 5.3 Diagram <i>Sequence</i> Menampilkan Kata-Kata dari Hasil Deteksi Kata	53
Gambar 5.4 Diagram <i>Sequence</i> Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	54
Gambar 5.5 Diagram <i>Sequence</i> Menentukan Tahapan Permainan.....	55
Gambar 5.6 Diagram <i>Sequence</i> Menentukan <i>Level</i> Permainan	56
Gambar 5.7 Diagram <i>Sequence</i> Menampilkan Permainan Tahapan Latihan	57
Gambar 5.8 Diagram <i>Sequence</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	58
Gambar 5.9 Diagram <i>Sequence</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	59
Gambar 5.10 Diagram <i>Sequence</i> Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video.....	61
Gambar 5.11 Diagram <i>Sequence</i> Menyediakan Mekanisme untuk Jawaban dari Permainan	62
Gambar 5.12 Perancangan Diagram <i>Class</i> Aplikasi Reado	64
Gambar 5.13 Perancangan ERD (<i>Entity Relationship Diagram</i>) Aplikasi Reado ...	66
Gambar 5.14 Tampilan Aplikasi untuk Logo Aplikasi Reado.....	80
Gambar 5.15 Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado.....	81
Gambar 5.16 Tampilan Aplikasi untuk Halaman Deteksi Kata.....	82
Gambar 5.17 Tampilan Aplikasi untuk Halaman Kata dari Hasil Deteksi Kata	83
Gambar 5.18 Tampilan Aplikasi untuk Halaman Alfabet dari Hasil Deteksi Kata	84



Gambar 5.19 Tampilan Aplikasi untuk Halaman Tahapan Permainan	85
Gambar 5.20 Tampilan Aplikasi untuk Halaman <i>Level</i> Permainan	86
Gambar 5.21 Tampilan Aplikasi untuk Halaman Permainan Tahapan Latihan	86
Gambar 5.22 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar	88
Gambar 5.23 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara	89
Gambar 5.24 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Video	90
Gambar 5.25 <i>Storyboard</i> Aplikasi Reado	92
Gambar 5.26 Prototipe Tampilan Aplikasi untuk Logo Aplikasi Reado	93
Gambar 5.27 Prototipe Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado	94
Gambar 5.28 Prototipe Tampilan Aplikasi untuk Halaman Deteksi Kata	94
Gambar 5.29 Prototipe Tampilan Aplikasi untuk Halaman Kata	95
Gambar 5.30 Prototipe Tampilan Aplikasi untuk Halaman Alfabet	95
Gambar 5.31 Prototipe Tampilan Aplikasi untuk Halaman Tahapan Permainan	96
Gambar 5.32 Prototipe Tampilan Aplikasi untuk Halaman <i>Level</i> Permainan	96
Gambar 5.33 Prototipe Tampilan Aplikasi untuk Halaman Tahapan Latihan	97
Gambar 5.34 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Gambar	97
Gambar 5.35 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Suara	98
Gambar 5.36 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Video	98
Gambar 5.37 Implementasi Antarmuka Tampilan Aplikasi untuk Logo Aplikasi Reado	126
Gambar 5.38 Implementasi Antarmuka Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado	127
Gambar 5.39 Implementasi Antarmuka Tampilan Aplikasi Halaman Deteksi Kata	127
Gambar 5.40 Implementasi Antarmuka Tampilan Aplikasi Halaman Kata	128
Gambar 5.41 Implementasi Antarmuka Tampilan Aplikasi Halaman Alfabet	128
Gambar 5.42 Implementasi Antarmuka Tampilan Aplikasi Tahapan Permainan	129
Gambar 5.43 Implementasi Antarmuka Tampilan Aplikasi <i>Level</i> Permainan	129
Gambar 5.44 Implementasi Antarmuka Tampilan Aplikasi Tahapan Latihan	130
Gambar 5.45 Implementasi Antarmuka Tampilan Tebak Kata dari Gambar	130



Gambar 5.46 Implementasi Antarmuka Tampilan Tebak Kata dari Suara 131

Gambar 5.47 Implementasi Antarmuka Tampilan Tebak Kata dari Video 131



DAFTAR LAMPIRAN

Lampiran A Pelaksanaan Pengembangan Aplikasi.....	161
A.1 Dokumentasi Pelaksanaan Pengembangan Aplikasi Reado untuk Analisis Kebutuhan	161
A.2 Dokumentasi Pelaksanaan Pengembangan Aplikasi Reado untuk Perancangan	161
Lampiran B Hasil Wawancara.....	162
B.1 Hasil Wawancara dari Pengajar yaitu Guru Sekolah	162
B.2 Hasil Wawancara dari Pengajar yaitu Relawan Mengajar.....	164
Lampiran C Pelaksanaan Pengujian Aplikasi	166
C.1 Dokumentasi Pelaksanaan Pengujian <i>Usability</i>	166
Lampiran D Surat Keterangan Pengujian	167
D.1 Surat Permohonan Data Skripsi	167
D.2 Surat Permohonan Pengisian Kuesioner.....	168
D.3 Surat Keterangan Pelaksanaan Pengujian.....	169



BAB 1 PENDAHULUAN

Bagian pertama pada penelitian skripsi ini adalah pendahuluan yang menjelaskan permasalahan yang diangkat di dalam penelitian ini untuk diteliti dan diselesaikan yang terdiri atas, latar belakang dari permasalahan, rumusan masalah, tujuan dari penelitian, manfaat dari penelitian, batasan-batasan dari permasalahan yang akan diselesaikan serta sistematika pembahasan dalam penelitian skripsi ini.

1.1 Latar belakang

Indonesia merupakan negara yang menerapkan penggunaan bahasa Inggris sebagai pembelajaran pada pendidikan formal baik dari jenjang sekolah dasar sampai perguruan tinggi. Namun, berdasarkan hasil studi yang dilakukan oleh Education First (EF) yaitu indeks kecakapan bahasa Inggris dari 100 negara di dunia, Indonesia menempati peringkat 61 dalam kategori rendah (EF EPI, 2019). Peringkat di tahun 2019 ini, mengalami penurunan di tahun sebelumnya yaitu peringkat 51. Selain itu berdasarkan dari hasil studi tersebut, negara-negara Asia memiliki rata-rata indeks kecakapan bahasa Inggris yaitu 53, sedangkan Indonesia memiliki indeks dengan nilai 50,06. Maka dari itu, didapatkan bahwa kecakapan bahasa Inggris di Indonesia saat ini mengalami penurunan dan berada di bawah rata-rata angka indeks kecakapan bahasa Inggris dari negara-negara di Asia.

Pembelajaran bahasa Inggris yang diterapkan pada anak sangatlah berpengaruh besar terhadap kemampuan anak kedepannya dalam merespon tuntutan kemajuan zaman melalui komunikasi (Rabbianty, E.N., 2015). Kemampuan komunikasi ini diharapkan mampu menjelaskan pemikiran, perasaan, serta untuk bersosialisasi dengan orang lain. Perkembangan anak yang terjadi dalam kurun waktu 4 tahun pertama sama besarnya dengan perkembangan yang akan terjadi pada kurun waktu 14 tahun berikutnya, dan sampai perkembangan otak akan mengalami stagnasi (Djalal, F. dan Supriadi, D., 2002). Namun, pembelajaran bahasa Inggris bagi anak yang masih belum efektif dan efisien menimbulkan anak-anak sulit dalam mempelajari bahasa Inggris. Maka dari itu, diperlukan adanya perbaikan di dalam pembelajaran bahasa Inggris yang saat ini diterapkan pada pendidikan formal sehingga mampu meningkatkan pembelajaran demi tercapainya tujuan yang diinginkan.

Dalam mempelajari bahasa Inggris, hal terpenting yang harus dilakukan yaitu mempelajari kata-kata dalam bahasa Inggris. Pembelajaran kata merupakan salah satu strategi dalam mempelajari bahasa Inggris yang mana harus dilakukan dengan adanya pengulangan terhadap kata yang telah dipelajari (Quan, Z., 2016). Pembelajaran kata dalam bahasa Inggris bagi anak meliputi cara membaca kata tersebut dalam bahasa Inggris dan penulisan dari kata. Kata-kata tersebut dapat dijumpai di dalam buku, poster, majalah dan media lainnya. Selain itu, pembelajaran kata dengan menebak kata-kata dalam bahasa Inggris secara insidental mampu menciptakan keterampilan yang penting bagi pelajar (Quan, Z., 2016).



Bentuk implementasi untuk menciptakan media pembelajaran yang lebih efektif dan efisien yaitu dengan menggunakan metode pengajaran yang interaktif melalui permainan edukatif. Permainan edukatif mampu menciptakan motivasi diri bagi pelajar untuk dapat lebih kompetitif, kooperatif dan mencari solusi serta informasi secara aktif (Prensky, M., 2003). Permainan edukatif dengan menggunakan perangkat eletronik memiliki potensi yang besar sebagai alat pembelajaran (Mayo, M.J., 2007). Maka dari itu, dengan menerapkan metode pengajaran melalui permainan edukatif pada perangkat elektronik mampu menciptakan media pembelajaran bahasa Inggris yang efektif dan efisien bagi pelajar. Namun penerapan metode pengajaran ini, masih harus dibawah pengawasan orang tua ataupun guru, sehingga mampu menghindari permasalahan yaitu ketergantungan anak terhadap perangkat elektronik.

Saat ini, penggunaan perangkat eletronik yaitu *smartphone* di Indonesia semakin meningkat. Berdasarkan data statistik, sebagian besar masyarakat menggunakan perangkat bergerak berbasis *mobile* atau *smartphone* yaitu sebanyak 92,8 juta yang merupakan penggunaan terbanyak dibandingkan dengan penggunaan laptop ataupun komputer (Asosiasi Penyelenggara Jasa Internet Indonesia, 2016). Selain itu, sebagian besar perangkat bergerak berbasis *mobile* atau *smartphone* menggunakan sistem operasi Android yaitu sebanyak 72,9% penggunaannya secara global (We Are Social, 2017).

Di dalam penelitian ini, pengembangan permainan edukatif melalui perangkat bergerak *mobile* berbasis Android akan dilakukan dengan menggunakan metode *Design Sprint*. Metode *Design Sprint* merupakan metode yang dapat menyelesaikan permasalahan secara kreatif dan produktif dalam konteks pembelajaran dengan mempertimbangkan kebutuhan dari pengguna akhir (Ferreira, V.G. dan Canedo, E.D., 2019). Metode ini dilakukan dalam waktu 5 hari dengan tujuan untuk menciptakan produk yang memiliki nilai bisnis dan dikembangkan secara efektif dan efisien (Knapp, et al., 2016). Pengimplementasian metode ini dilakukan untuk menyelesaikan permasalahan dari pengguna dengan hasil keluaran yaitu berupa prototipe aplikasi yang fungsional. Tahap dari pengembangan dengan menggunakan metode *Design Sprint*, melibatkan guru dan orang tua sebagai pembimbing anak dalam belajar untuk memvalidasi keseluruhan proses pengembangan ini.

Diharapkan dengan adanya penelitian ini, hasil dari prototipe yang sudah tervalidasi dengan menggunakan metode *Design Sprint* mampu dikembangkan menjadi sebuah aplikasi untuk permainan edukatif anak dengan nama aplikasi yaitu Reado. Hasil dari penelitian akan diuji cobakan dengan melakukan pengujian *black box* dan studi kegunaan aplikasi. Pengujian *black box* dilakukan untuk mengetahui bahwa aplikasi yang dikembangkan sudah memenuhi kebutuhan dari pengguna. Sedangkan, studi kegunaan aplikasi dilakukan untuk mengetahui tingkat kegunaan aplikasi dengan menggunakan parameter penilaian yaitu efektivitas, efisiensi dan kepuasan pengguna (ISO 9241-210, 2010). Pengujian dilakukan pada anak-anak dari kegiatan sosial komunitas Katalis Pendidikan.



1.2 Rumusan masalah

Dari latar belakang tersebut didapatkan rumusan masalah yang akan dibahas pada penelitian ini, antara lain:

1. Bagaimana hasil analisis kebutuhan dari pengembangan aplikasi Reado?
2. Bagaimana hasil rancangan dari pengembangan aplikasi Reado?
3. Bagaimana hasil implementasi dari pengembangan aplikasi Reado?
4. Bagaimana hasil pengujian dari pengembangan aplikasi Reado?
5. Bagaimana tingkat kegunaan aplikasi Reado dengan menggunakan metode *Design Sprint* dari hasil pengujian kegunaan aplikasi atau *usability* berdasarkan parameter pengukuran efisiensi, efektivitas, dan kepuasan pengguna?

1.3 Tujuan

Tujuan yang didapatkan dengan adanya penelitian ini, antara lain yaitu:

1. Mengetahui hasil analisis kebutuhan dari pengembangan aplikasi Reado.
2. Mengetahui hasil rancangan dari pengembangan aplikasi Reado.
3. Mengetahui hasil implementasi dari pengembangan aplikasi Reado.
4. Mengetahui hasil pengujian dari pengembangan aplikasi Reado.
5. Mengetahui tingkat kegunaan aplikasi Reado dengan menggunakan metode *Design Sprint* dari hasil pengujian kegunaan aplikasi atau *usability* berdasarkan parameter pengukuran efektivitas, efisiensi, dan kepuasan pengguna.

1.4 Manfaat

Manfaat dari penelitian ini dapat diuraikan sebagai berikut:

1. Manfaat bagi penulis
Dapat menganalisis, merancang, mengembangkan dan menguji aplikasi perangkat bergerak berbasis *mobile* menggunakan sistem operasi Android sesuai dengan materi yang sudah dipelajari di perkuliahan.
2. Manfaat bagi peneliti selanjutnya
Dapat dijadikan sebagai acuan dasar di dalam pengembangan aplikasi dengan menerapkan metode *Design Sprint* untuk aplikasi perangkat bergerak *mobile* berbasis Android dan pengujian aplikasi dengan menggunakan pengujian *black box* dan studi kegunaan aplikasi berdasarkan parameter pengukuran yaitu efektivitas, efisiensi dan kepuasan pengguna.
3. Manfaat bagi pengguna



Dapat memiliki media untuk permainan edukatif bagi anak yang mampu membantu anak untuk mempelajari Bahasa Inggris dengan efektif dan efisien.

1.5 Batasan masalah

Di dalam penelitian ini dapat diuraikan menjadi beberapa batasan masalah, antara lain sebagai berikut:

1. Aplikasi ini hanya ditujukan untuk mata pelajaran Bahasa Inggris.
2. Aplikasi ini dikembangkan untuk anak-anak sebagai media permainan edukatif yang berfokus pada pembelajaran kata.
3. Aplikasi ini dikembangkan untuk anak-anak pada kegiatan sosial Komunitas Katalis Pendidikan.
4. Aplikasi merupakan aplikasi perangkat bergerak berbasis *mobile*.
5. Aplikasi ini dikembangkan dengan penerapan metode *design sprint*.

1.6 Sistematika pembahasan

Sistematika pembahasan di dalam penyusunan skripsi ini terbagi atas beberapa bab yang terdiri dari:

BAB 1 PENDAHULUAN

Menguraikan beberapa sub bab yang terdiri atas latar belakang, rumusan masalah, tujuan dari penelitian, manfaat dari penelitian, batasan-batasan masalah di dalam penelitian serta sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Dalam bab ini dijelaskan pembahasan terkait dengan kajian-kajian pada penelitian sebelumnya serta teori-teori yang berkaitan dengan penelitian ini yang mana menjadi acuan dasar di dalam analisis, perancangan dan pengembangan pada penelitian ini. Adapun landasan kepastakaan di dalam penelitian ini adalah penjelasan terkait dengan Aplikasi Perangkat Bergerak, Metode *Design Sprint*, Android, Pengujian *Black Box* yaitu *Equalivalence Partitioning*, Pengujian *Usability* atau Kegunaan Aplikasi berdasarkan parameter pengukuran efisiensi, efektivitas dan kepuasan pengguna.

BAB 3 METODOLOGI PENELITIAN

Menguraikan penjelasan terkait dengan cara sistematika untuk menyelesaikan permasalahan yang terdiri atas tipe penelitian dan strategi dan rancangan penelitian yang terdiri atas strategi atau metode secara umum, subjek penelitian, lokasi penelitian, teknik pengumpulan data dan desain penelitian.

BAB 4 ANALISIS KEBUTUHAN



Membahas terkait dengan tahapan dalam penggalian kebutuhan pengguna serta analisis kebutuhan pengguna untuk pengembangan dari aplikasi Reado. Analisis kebutuhan pada penelitian ini, dilakukan dengan menerapkan metode *Design Sprint* pada hari pertama yaitu tahapan *understand*. Di dalam bagian ini juga terdapat pembahasan terkait dengan identifikasi aktor, analisis kebutuhan perangkat lunak, identifikasi kebutuhan yang terdiri atas kebutuhan fungsional dan non fungsional serta pemodelan dari kebutuhan.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Membahas terkait dengan tahapan dalam perancangan serta implementasi aplikasi berdasarkan hasil dari kebutuhan pengguna yang sudah di analisis sebelumnya. Perancangan aplikasi pada penelitian ini, dilakukan dengan menerapkan metode *Design Sprint* yang dilakukan pada hari kedua sampai kelima yaitu tahapan *ideate*, *decide*, *prototype* dan *test*. Tahapan-tahapan ini dilakukan untuk mendapatkan prototipe yang tervalidasi dan selanjutnya akan diimplementasikan menjadi aplikasi yang dapat digunakan oleh pengguna.

BAB 6 PENGUJIAN

Membahas terkait dengan hal-hal yang berkaitan dengan tahapan pengujian dalam penelitian ini yang terdiri atas pengujian *black box* dan pengujian *usability* atau kegunaan aplikasi. Pengujian *usability* dilakukan yang menggunakan teknik *equivalence partitioning* dan pengujian *usability* dilakukan berdasarkan parameter pengukuran dari aspek efisiensi, efektivitas dan kepuasan pengguna.

BAB 7 PENUTUP

Membahas terkait dengan kesimpulan yang didapatkan berdasarkan penelitian yang telah dilakukan serta saran-saran yang mungkin bermanfaat di dalam pengembangan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Bagian kedua pada penelitian skripsi ini adalah landasan kepustakaan yang menjelaskan kajian-kajian pada penelitian sebelumnya serta teori-teori yang berkaitan dengan penelitian yang terdiri atas penjelasan Aplikasi Perangkat Bergerak, Metode *Design Sprint*, Android, Pengujian *Black Box* yaitu *Equalivalence Partitioning*, Pengujian *Usability* atau Kegunaan Aplikasi berdasarkan parameter pengukuran efisiensi, efektivitas dan kepuasan pengguna.

2.1 Kajian Pustaka

Beberapa acuan dari penelitian-penelitian sebelumnya yang berkaitan dengan penelitian ini akan digunakan sebagai pembandingan dan acuan dasar dari pengembangan aplikasi Reado. Penelitian-penelitian sebelumnya antara lain yaitu penelitian yang membahas tentang pembelajaran Bahasa Inggris, perkembangan anak dalam belajar, permainan edukatif, metode pengembangan aplikasi yaitu metode *Design Sprint* dan studi kegunaan aplikasi.

Pembelajaran bahasa Inggris merupakan salah satu hal yang memiliki pengaruh besar terhadap kemampuan pada anak (Rabbianty, E.N., 2015). Bahkan beberapa penelitian menganggap pembelajaran bahasa Inggris merupakan salah satu langkah untuk membantu anak dalam merespon tuntutan kemajuan zaman melalui komunikasi. Pembelajaran Bahasa Inggris bagi anak sangatlah penting, berdasarkan hasil penelitian menunjukkan bahwa perkembangan anak dalam kurun waktu 4 tahun pertama sama besarnya dengan perkembangan yang akan terjadi pada 14 tahun berikutnya dan sampai perkembangan otak anak akan mengalami stagnasi (Djalal, F., 2002). Berdasarkan penelitian sebelumnya, maka pembelajaran bahasa Inggris sangat penting untuk diterapkan bagi anak sehingga pengembangan pada penelitian skripsi ini akan terfokus terhadap pembelajaran bahasa Inggris bagi anak.

Pembelajaran bahasa Inggris memiliki beberapa metode yang dapat diterapkan. Namun, strategi untuk mempelajari bahasa Inggris salah satunya yaitu pembelajaran kata dalam bahasa Inggris yang dilakukan dengan adanya pengulangan terhadap kata yang telah dipelajari (Quan, Z., 2016). Berdasarkan penelitian sebelumnya, terdapat beberapa cara untuk menerapkan strategi dari pembelajaran kata ini salah satunya yaitu dengan menebak kata-kata dalam Bahasa Inggris secara insidental. Maka dari itu, didapatkan bahwa pembelajaran bahasa Inggris bagi anak dapat dilakukan salah satunya yaitu dengan menggunakan strategi untuk pembelajaran kata dalam bahasa Inggris dan disertai dengan adanya pengulangan terhadap kata yang telah dipelajari seperti menebak kata-kata secara rutin.

Permainan edukatif merupakan salah satu metode pengajaran yang mampu menciptakan motivasi untuk dapat lebih kompetitif, kooperatif dan aktif (Prensky, M., 2003). Permainan edukatif juga memiliki peluang yang sangat besar untuk diterapkan dalam pendidikan. Pembelajaran pada pendidikan saat ini semakin



mengikuti perkembangan zaman untuk lebih meningkatkan sistem pembelajaran yang lebih baik. Bahkan berdasarkan penelitian sebelumnya, Pembelajaran dalam bentuk permainan edukatif melalui perangkat elektronik memiliki potensi yang sangat besar (Mayo, M.J., 2007). Maka dari itu, hal ini dapat disimpulkan bahwa pembelajaran melalui permainan edukatif dengan menggunakan perangkat elektronik dapat memiliki peluang yang sangat besar untuk dapat menciptakan lingkungan pembelajaran yang lebih baik.

Berdasarkan penelitian sebelumnya, metode pengembangan aplikasi yang dapat digunakan untuk menyelesaikan permasalahan secara kreatif dan produktif ialah metode *Design Sprint* (Ferreira, V.G. dan Canedo, E.D., 2019). Metode *Design Sprint* merupakan metode pengembangan yang mampu mempertimbangkan kebutuhan-kebutuhan dari pengguna untuk menentukan solusi yang akan dikembangkan. Metode *Design Sprint* juga mampu untuk menciptakan produk yang memiliki nilai bisnis dan dikembangkan secara efektif dan efisien (Knapp, et al., 2016). Maka dari itu, pengembangan aplikasi dengan menggunakan metode *Design Sprint* mampu membantu peneliti untuk mengembangkan aplikasi secara efektif dan efisien yang mana memiliki nilai bisnis dan dapat digunakan untuk menyelesaikan permasalahan dengan mempertimbangkan kebutuhan dari pengguna.

2.2 Aplikasi Perangkat Bergerak

Perangkat bergerak atau *mobile device* merupakan teknologi yang saat ini sudah sangat berkembang dan sangat meningkat penggunaannya di seluruh dunia. Peningkatan dari penggunaan perangkat bergerak ini ditandai dengan ditemukannya perangkat cerdas atau *smartphone* di tahun 2008. Saat ini berdasarkan data statistik, sebagian besar masyarakat menggunakan perangkat bergerak berbasis *mobile* atau *smartphone* yaitu sebanyak 92,8 juta yang merupakan penggunaan terbanyak dibandingkan dengan penggunaan laptop ataupun komputer (Asosiasi Penyelenggara Jasa Internet Indonesia, 2016). Sebagian besar perangkat bergerak berbasis *mobile* atau *smartphone* menggunakan sistem operasi Android yaitu sebanyak 72,9% penggunaannya secara global (We Are Social, 2017). Perangkat cerdas atau *smartphone* menghadirkan banyak inovasi-inovasi dari berbagai aplikasi dan fungsional-fungsional pada perangkat bergerak. Aplikasi perangkat bergerak adalah program yang berjalan pada perangkat cerdas atau *smartphone*, tablet ataupun perangkat bergerak yang sejenis. Berbagai aplikasi dan fungsional-fungsional pada perangkat bergerak menunjukkan bahwa kebutuhan-kebutuhan pengguna akan aplikasi dan fungsional-fungsional tersebut pada perangkat bergerak sangatlah tinggi.

Aplikasi perangkat bergerak memiliki perbedaan-perbedaan dibandingkan dengan aplikasi yang lainnya. Perbedaan ini dapat ditunjukkan dengan adanya karakteristik-karakteristik dari aplikasi perangkat bergerak tersebut. Adapun karakteristik-karakteristik dari aplikasi perangkat bergerak, antara lain sebagai berikut (Tolle, H., et al., 2017):

1. Kemampuan Komputasi



Kemampuan komputasi dari aplikasi perangkat bergerak memiliki kemampuan yang lebih rendah dibandingkan dengan aplikasi lainnya seperti aplikasi *desktop*. Kemampuan komputasi yang lebih rendah ini disebabkan karena arsitektur pada perangkat bergerak mampu menurunkan performa pada perangkat bergerak yang mana memiliki orientasi pada efisiensi energi.

2. Komunikasi Nirkabel

Komunikasi nirkabel merupakan komunikasi untuk pertukaran data dan informasi yang menerapkan konsep nirkabel atau tidak terhubung dengan penghantar listrik. Komunikasi ini dapat berupa WiFi, Bluetooth, GPS (*Global Positioning System*), internet dan lain-lain.

3. Dimensi Fisik

Dimensi fisik dari perangkat bergerak memiliki ukuran yang relatif lebih kecil dibandingkan dengan perangkat lainnya sehingga mampu memudahkan di dalam penggunaannya yang mana bersifat *portable* atau mudah untuk dibawa.

4. Catu Daya

Catu daya atau baterai pada perangkat bergerak memiliki daya tahan atau kapasitas yang baik dan memiliki peningkatan yang signifikan seiring dengan perkembangan teknologi.

2.3 Pengembangan Aplikasi Perangkat Bergerak dengan Metode *Design Sprint*

Design Sprint merupakan metodologi di dalam pengembangan aplikasi dengan desain yang dikembangkan oleh Google Ventures dengan tujuan untuk pengembangan yang lebih cepat, efisien dan lebih dinamis. Metode *Design Sprint* merupakan metode yang dapat menyelesaikan permasalahan secara kreatif dan produktif dalam konteks pembelajaran dengan mempertimbangkan kebutuhan dari pengguna akhir (Ferreira, V.G. dan Canedo, E.D., 2019). Metode *Design Sprint* bertujuan untuk mengembangkan aplikasi yang hanya terfokus untuk menemukan, memahami dan mendefinisikan permasalahan-permasalahan yang dihadapi oleh pengguna. Metode ini dilakukan dalam waktu 5 hari dengan tujuan untuk menciptakan produk yang memiliki nilai bisnis dan dikembangkan secara efektif dan efisien (Knapp, et al., 2016).

Pada metode *Design Sprint* memiliki 6 fase pengembangan yang dilakukan yaitu *understand*, *define*, *diverge*, *decide*, *prototype* dan *validate*. Dari beberapa fase-fase pengembangan ini akan dilakukan dalam waktu 5 hari pengembangan. Maka dari itu, pengembangan aplikasi yang akan dilakukan dengan menggunakan metode *Design Sprint* akan dilaksanakan dalam 5 tahapan yang dilakukan pada 5 hari pengembangan. Tahapan-tahapan dengan metode *Design Sprint* yaitu tahapan *understand*, *ideate*, *decide*, *prototype* dan *test*. Adapun penjelasan lebih rinci dari tahapan yang dilakukan dalam metode *Design Sprint*, antara lain sebagai berikut (Knapp, et al., 2016):

1. Hari Pertama: Tahapan *understand*



Tahapan *understand* dilakukan untuk fase pengembangan yaitu *understand* dan *define*. Fase *understand* merupakan fase pengembangan yang dilakukan untuk mendapatkan kebutuhan-kebutuhan dari produk, layanan ataupun ide bisnis dari aplikasi yang akan dikembangkan. Selain itu, fase *define* adalah fase pengembangan untuk menentukan tujuan dan spesifikasi dari produk dan layanan yang akan dikembangkan untuk menciptakan metrik, posisi ataupun fokus utama dari pengembangan aplikasi. Adapun untuk melakukan tahapan *understand* ini dari fase *understand* dan *define* dilakukan melalui beberapa langkah, antara lain yaitu:

Fase *Understand* :

- Menentukan *long-term goal* atau tujuan jangka panjang dari pengembangan aplikasi yang dilakukan.
- Menentukan *sprint questions* atau kemungkinan-kemungkinan dan tantangan yang akan menyebabkan kegagalan dari pengembangan aplikasi yang dilakukan.
- Membuat *project map* atau peta proyek berdasarkan pengalaman dari pengguna dengan tujuan untuk membuat *user journey maps* dari pengguna. *User journey map* digunakan untuk mengetahui langkah-langkah yang dilakukan pengguna untuk dapat menyelesaikan permasalahan.
- Melakukan *ask the expert* atau tanya ahli dengan tujuan untuk eksplorasi kebutuhan dari pengguna. Dalam eksplorasi kebutuhan ini dapat menggunakan beberapa metode antara lain yaitu *360 degree lightning talks*, *user interviews*, mengunjungi pengguna di tempat dimana produk digunakan, *stakeholder map*, *competitive overview* dan *summarize the learnings*.
- Menentukan kebutuhan-kebutuhan dari pengguna dengan membuat daftar *How Might We* (HMW) dari setiap masing-masing anggota tim pengembang.
- Melakukan *affinity mapping* atau pemetaan afinitas terhadap daftar *How Might We* (HMW) untuk mengelompokkan daftar HMW berdasarkan persamaan atau kemiripan.
- Melakukan *zen voting* untuk memilih kebutuhan-kebutuhan yang menjadi prioritas untuk dikembangkan.

Fase *Define* :

- Menetapkan metrik-metrik dari pengembangan aplikasi. Salah satu metode yang dapat digunakan untuk menetapkan metrik pengembangan yaitu dengan menggunakan metode *Heart Metric*.

2. Hari Kedua: Tahapan *Ideate*

Tahapan *ideate* dilakukan untuk fase pengembangan yaitu *diverge*. Fase *diverge* yaitu fase pengembangan untuk menggali solusi-solusi dalam bentuk *sketch* dalam pengembangan dari aplikasi. Adapun untuk melakukan tahapan *ideate* ini dari fase *diverge* dilakukan melalui beberapa langkah, antara lain yaitu:



Fase *Diverge* :

- Melakukan *lightning demos* bertujuan untuk merepresentasikan referensi dari kompetitor ataupun referensi lainnya.
- Membuat sketsa yaitu menggambarkan sketsa pada kertas untuk solusi pengembangan dari aplikasi.

3. Hari Ketiga: Tahapan *Decide*

Tahapan *decide* dilakukan untuk fase pengembangan yaitu *decide*. Fase *decide* yaitu fase pengembangan untuk menentukan *storyboard* secara utuh untuk aplikasi yang akan dikembangkan. Adapun untuk melakukan tahapan *decide* ini dari fase *decide* dilakukan melalui beberapa langkah, antara lain yaitu:

Fase *Decide* :

- Melakukan *sticky decision* bertujuan untuk merepresentasikan *sketch* solusi yang sudah dibuat dan menentukan sketsa solusi yang akan dibuat menjadi prototipe.
- Membuat *storyboard* secara utuh berdasarkan sketsa yang sudah ditentukan.

4. Hari Keempat: Tahapan *Prototype*

Tahapan *prototype* dilakukan untuk fase pengembangan yaitu *prototype*. Fase *prototype* yaitu fase pengembangan untuk membuat prototipe dengan tujuan untuk membuat model dari aplikasi yang akan dikembangkan berupa *mockup*, video, ataupun produk fisik dari aplikasi yang akan dikembangkan. Adapun untuk melakukan tahapan *prototype* ini dari fase *prototype* dilakukan melalui beberapa langkah, antara lain yaitu:

Fase *Prototype* :

- Melakukan *prototyping* berdasarkan hasil dari tahapan yang sudah dilakukan sebelumnya.

5. Hari Kelima: Tahapan *Test*

Tahapan *test* dilakukan untuk fase pengembangan yaitu *validate*. Fase *validate* yaitu fase pengembangan untuk menentukan apakah prototipe yang sudah dibuat sebelumnya sudah memenuhi kebutuhan-kebutuhan dari pengguna. Adapun untuk melakukan tahapan *test* ini dari fase *validate* dilakukan melalui beberapa langkah, antara lain yaitu:

Fase *Validate* :

- Melakukan *testing* dengan melakukan interview kepada *stakeholder*.

2.4 Android

Android adalah sebuah sistem operasi yang dikeluarkan oleh Google dengan berbagai fitur yang sangat bagus dengan pemanfaatan tampilan yang menarik. Perangkat bergerak ini juga memiliki beberapa sensor, seperti *gyroscope*,



accelerometer, *proximity* dan sensor lainnya. Bukan hanya itu, Android juga menarik perhatian berbagai perusahaan di bidang teknologi yang mencari alternatif untuk dapat meminimalisir pengeluaran biaya serta kostumisasi sistem operasi untuk teknologi yang dikembangkan oleh perusahaan tersebut. Dari sisi *developer*, sistem operasi Android merupakan proyek *open source* sehingga *developer* dapat mengembangkan aplikasi berbasis Android secara aman dan gratis. Sistem operasi Android memiliki beberapa versi, yaitu Alpha, Beta, Cupcake, Donnut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jellybean, Kitkat, lollipop, Marshmallow, Nougat, Oreo, Pie dan terbaru Android 10.

2.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak bertujuan untuk memvalidasi apakah aplikasi yang sudah dikembangkan sudah memenuhi kebutuhan-kebutuhan dari pengguna. Teknik pengujian yang dapat dilakukan untuk tahapan pengujian perangkat lunak yaitu seperti pengujian *black box* dan pengujian *usability* atau kegunaan aplikasi.

2.5.1 Pengujian Black Box

Pengujian *black box* adalah salah satu pengujian perangkat lunak yang dapat dilakukan. Pengujian ini bertujuan untuk memvalidasi berdasarkan kebutuhan-kebutuhan fungsional dari aplikasi yang akan dikembangkan. Pengujian *black box* juga merupakan sebuah uji coba untuk melakukan tahap pengujian dari aplikasi yang sedang dikembangkan berdasarkan fungsional-fungsional dari proses bisnis aplikasi (Pressman, 2010). Di dalam pengujian ini, penguji tidak perlu mengetahui isi dari kode program aplikasi, hal ini dikarenakan pengujian dilakukan hanya untuk mengetahui apakah kebutuhan-kebutuhan fungsional dari aplikasi yang dikembangkan sudah sesuai dengan kebutuhan pengguna. Pengujian *black box* dilakukan untuk mencari informasi kesalahan terkait dengan hal-hal berikut (Mustaqbal, et al., 2015):

1. Fungsional aplikasi yang salah atau tidak ditemukan.
2. Kesalahan dari antarmuka atau *interface errors*.
3. Kesalahan dari struktur dan akses basis data.
4. Kesalahan kinerja.
5. Kesalahan dari terminasi dan inisialisasi.

Terdapat beberapa teknik yang dilakukan dalam pengujian *black box* terhadap aplikasi yang sedang dikembangkan, salah satunya adalah *equivalence partitioning*. Adapun penjelasan secara rinci dari teknik pengujian *black box*, antara lain sebagai berikut:

2.5.1.1 Equivalence Partitioning

Equivalence partitioning merupakan sebuah teknik pengujian dari pengujian *black box* yang mana memisahkan atau membagi domain masukan ke dalam kelas-kelas data dari *test case* yang diujikan untuk dapat diturunkan (Pressman, 2010).



Domain masukan dari aplikasi pada umumnya berupa nilai numerik, satu set nilai, rentang nilai ataupun kondisi boolean. *Equivalence partitioning* dapat didefinisikan berdasarkan beberapa kondisi berikut:

1. Jika masukan menentukan rentang nilai, maka satu valid dan dua tidak valid, kelas *equivalence* didefinisikan.
2. Jika masukan membutuhkan nilai tertentu, maka satu valid dan dua tidak valid, kelas *equivalence* didefinisikan.
3. Jika masukan menentukan satu set nilai, maka satu valid dan dua tidak valid, kelas *equivalence* didefinisikan.
4. Jika masukan adalah boolean, maka satu valid dan satu tidak valid, kelas *equivalence* didefinisikan.

2.5.2 Pengujian *Usability*

Pengujian *usability* atau kegunaan aplikasi merupakan sebuah metode atau teknik yang digunakan untuk melakukan evaluasi terhadap sebuah aplikasi ataupun produk yang mana terdiri atas beberapa proses pengujian baik melibatkan pengguna secara langsung maupun secara tidak langsung (Rubin, et al., 2008). Pengujian *usability* dilakukan berdasarkan partisipan yang menggunakan aplikasi. Adapun jumlah partisipan dari pengujian ini didapatkan dari perhitungan estimasi besar sampel dan menggunakan teknik sampling. Perhitungan estimasi besar sampel dihitung berdasarkan perhitungan rumus Slovin dapat dilihat pada persamaan 2.1, yaitu sebagai berikut (Umar, H., 2003):

$$n = \frac{N}{N(d)^2 + 1} \quad (2.1)$$

Keterangan:

n = jumlah estimasi besar sampel

N = jumlah populasi yang ada secara keseluruhan

d = presisi yaitu sebesar 10% atau 0,1

Berdasarkan hasil perhitungan estimasi besar sampel, selanjutnya untuk menentukan jumlah partisipan dilakukan dengan perhitungan untuk teknik sampling. Teknik sampling dapat dihitung dengan menggunakan *proportional random sampling* yaitu dilakukan pengambilan mengambil partisipan dari setiap kelompok atau wilayah (Arikunto, 2006). Adapun rumus perhitungan teknik sampling tersebut dapat dilihat pada persamaan 2.2, antara lain sebagai berikut (Sugiono, 2007):

$$n_i = \frac{x_i}{N} \times E \quad (2.2)$$

Keterangan:

n = jumlah partisipan yang diujikan kelas ke- i

x_i = jumlah populasi kelas ke- i



N = jumlah populasi yang ada secara keseluruhan

E = jumlah estimasi besar sampel

Usability atau kegunaan aplikasi adalah parameter untuk mengukur suatu tingkat penggunaan dari sebuah produk yang digunakan oleh pengguna untuk tujuan tertentu dengan melibatkan faktor efisiensi, efektivitas dan kepuasan pengguna (ISO, 1998). Pengujian *usability* juga dilakukan untuk mengetahui tingkat kegunaan aplikasi dengan menggunakan parameter penilaian yaitu efektivitas, efisiensi dan kepuasan pengguna (ISO 9241-210, 2010). Adapun parameter-parameter yang menjadi parameter penilaian untuk pengujian *usability* atau kegunaan aplikasi berdasarkan pengertian yang dikemukakan oleh ISO (1998), antara lain sebagai berikut:

2.5.2.1 Efisiensi

Parameter penilaian efisiensi merupakan suatu parameter penilaian yang digunakan dalam melakukan pengukuran tingkat kegunaan aplikasi dari aspek efisiensi yaitu dari segi waktu pengerjaan *task* yang diberikan, waktu dapat berupa dalam satuan detik ataupun menit yang diberikan untuk menyelesaikan satu tugas (Mifsud, 2015). Parameter penilaian efisiensi juga dapat diukur melalui pengukuran efisiensi relatif keseluruhan. Pengukuran efisiensi relatif keseluruhan di dalam perhitungannya membutuhkan rasio dari waktu yang digunakan pengguna untuk menyelesaikan tugas dan berhubungan dengan total waktu keseluruhan yang dihabiskan oleh keseluruhan pengguna (Sergeev, 2010). Berikut merupakan persamaan untuk menghitung efisiensi relatif keseluruhan (Sergeev, 2010):

$$\text{Efisiensi relatif keseluruhan} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \% \quad (2.3)$$

Keterangan:

R = jumlah partisipan atau subjek penelitian

N = jumlah tugas

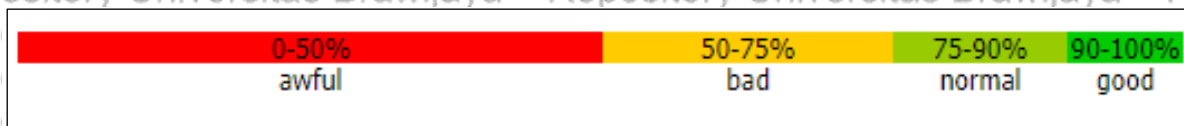
n_{ij} = keberhasilan pengguna pada tugas ke- i oleh partisipan ke- j

t_{ij} = waktu yang dibutuhkan untuk menyelesaikan tugas ke- i oleh partisipan ke- j dalam satuan detik

Pada persamaan 2.3 ditunjukkan bahwa untuk menghitung efisiensi relatif keseluruhan yaitu dengan menjumlahkan seluruh waktu yang dibutuhkan untuk menyelesaikan *task* yang berhasil lalu dibagi dengan waktu total dari keseluruhan dan terakhir dikalikan dengan 100% yang mana $R_{j=1}$ merupakan pengguna ke- j yang melakukan pengujian *usability* atau kegunaan aplikasi, $N_{i=1}$ merupakan tugas ke- i , n_{ij} merupakan nilai yang menunjukkan berhasil atau tidaknya tugas ke- i yang diselesaikan oleh pengguna ke- j yang mana jika bernilai 1 maka dinyatakan berhasil menyelesaikan tugas dan jika bernilai 0 maka dinyatakan tidak berhasil untuk menyelesaikan tugas, dan untuk t_{ij} merupakan waktu yang dibutuhkan untuk menyelesaikan tugas ke- i oleh pengguna ke- j dalam satuan detik.



Tingkat kegunaan aplikasi berdasarkan parameter penilaian yaitu efisiensi dari hasil perhitungan efisiensi relatif keseluruhan dapat dilihat dari tingkat kepercayaan aplikasi. Tingkat kepercayaan aplikasi terdiri atas beberapa tingkatan-tingkatan berdasarkan persentase dari hasil perhitungan efisiensi relatif keseluruhan. Tingkatan-tingkatan tersebut terdiri atas tingkatan *awful* atau mengerikan, *bad* atau buruk, normal dan *good* atau baik (Sergeev, 2010). Adapun tingkat kepercayaan aplikasi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tingkat Kepercayaan Aplikasi

2.5.2.2 Efektivitas

Efektivitas merupakan suatu parameter penilaian yang menunjukkan kelengkapan dan ketepatan pengguna untuk mencapai tujuan tertentu. Pengukuran tingkat efektivitas dari aplikasi dapat dilakukan dengan salah satu pengukuran yaitu pengukuran tingkat kesalahan (Mifsud, 2015). Pengukuran tingkat kesalahan dilakukan dengan jumlah kesalahan yang ditimbulkan oleh pengguna ketika melakukan penyelesaian tugas. Tingkat kesalahan pengguna dapat diukur melalui perhitungan dari banyaknya total kesalahan yang dilakukan pengguna dan total kesempatan yang merupakan jumlah kesempatan atau langkah yang dilakukan oleh pengguna. Perhitungan dari tingkat kesalahan pengguna dapat dilihat pada persamaan 2.4 berikut (Mifsud, 2015):

$$\text{Tingkat kesalahan} = \frac{\text{Total Kesalahan}}{\text{Total Kesempatan} \times \text{Total Partisipan}} \quad (2.4)$$

2.5.2.3 Kepuasan Pengguna

Kepuasan pengguna merupakan parameter penilaian yang mana penilainnya ditentukan secara subyektif oleh pengguna dengan cara memberikan kuisioner formal setelah pengguna menggunakan aplikasi yang sedang diujikan (Mifsud, 2015). Adapun parameter penilaian yang digunakan dalam menentukan parameter kepuasan pengguna ini, antara lain sebagai berikut (Baharrudin, et al., 2018):

1. Tingkat kenyamanan pengguna dalam menggunakan aplikasi
2. Tingkat kemudahan dari penggunaan aplikasi oleh pengguna

Tingkat kegunaan aplikasi berdasarkan aspek kepuasan pengguna dapat ditentukan dengan menggunakan kuisioner *System Usability Scale* (SUS). *System Usability Scale* (SUS) merupakan suatu skala yang terdiri atas 10 pernyataan yang mana skala tersebut menjelaskan terkait dengan penilaian subyektif terhadap *usability* atau kegunaan aplikasi dari aplikasi. SUS menggunakan penilaian skala *likert* dengan skala poin 1 sampai 5. Untuk mengukur tingkat kepuasan dari pengguna terhadap aplikasi, diperlukan penggunaan kuisioner SUS sehingga didapatkan pengukuran secara kuantitatif. Kuisioner SUS ini dapat dilakukan jika

responden dari kuesioner sungguh-sungguh untuk berpartisipasi (Rummel, 2015). Pernyataan yang digunakan dalam kuesioner SUS dapat dilihat pada Gambar 2.2 dan hasil terjemahan untuk pernyataan yang digunakan dalam kuesioner SUS dapat dilihat pada Gambar 2.3.

	Strongly Disagree					Strongly Agree
1. I think that I would like to use this system frequently	☐	☐	☐	☐	☐	
	1	2	3	4	5	
2. I found the system unnecessarily complex	☐	☐	☐	☐	☐	
	1	2	3	4	5	
3. I thought the system was easy to use	☐	☐	☐	☐	☐	
	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
5. I found the various functions in this system were well integrated	☐	☐	☐	☐	☐	
	1	2	3	4	5	
6. I thought there was too much inconsistency in this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly	☐	☐	☐	☐	☐	
	1	2	3	4	5	
8. I found the system very cumbersome to use	☐	☐	☐	☐	☐	
	1	2	3	4	5	
9. I felt very confident using the system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	

Gambar 2.2 Kuesioner System Usability Scale

Sumber: Brooke, J. (1986)



1. Saya berpikir, saya akan sering menggunakan sistem ini.
2. Saya merasa, sistem ini tidak semestinya rumit.
3. Saya merasa, sistem ini mudah untuk digunakan.
4. Saya akan membutuhkan bantuan dari orang lain untuk dapat menggunakan sistem ini.
5. Saya merasa, fitur-fitur di dalam sistem ini dapat berjalan dengan baik.
6. Saya merasa, ada terlalu banyak ketidakkonsistenan di dalam sistem ini.
7. Saya merasa, orang lain akan memahami dengan cepat dalam menggunakan sistem ini.
8. Saya merasa sistem ini sangat rumit untuk digunakan.
9. Saya merasa sangat percaya diri atau yakin dalam menggunakan sistem ini.
10. Saya berpikir, saya perlu belajar banyak hal untuk membiasakan diri sebelum menggunakan sistem ini.

Sangat Tidak Setuju

Sangat Setuju

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

--	--	--	--	--

1 2 3 4 5

Gambar 2.3 Kuesioner System Usability Scale (Terjemahan)

Adapun keterangan-keterangan untuk menentukan nilai dari aspek kepuasan pengguna dengan menggunakan kuesioner SUS, antara lain sebagai berikut:

K = pernyataan dari kuesioner SUS

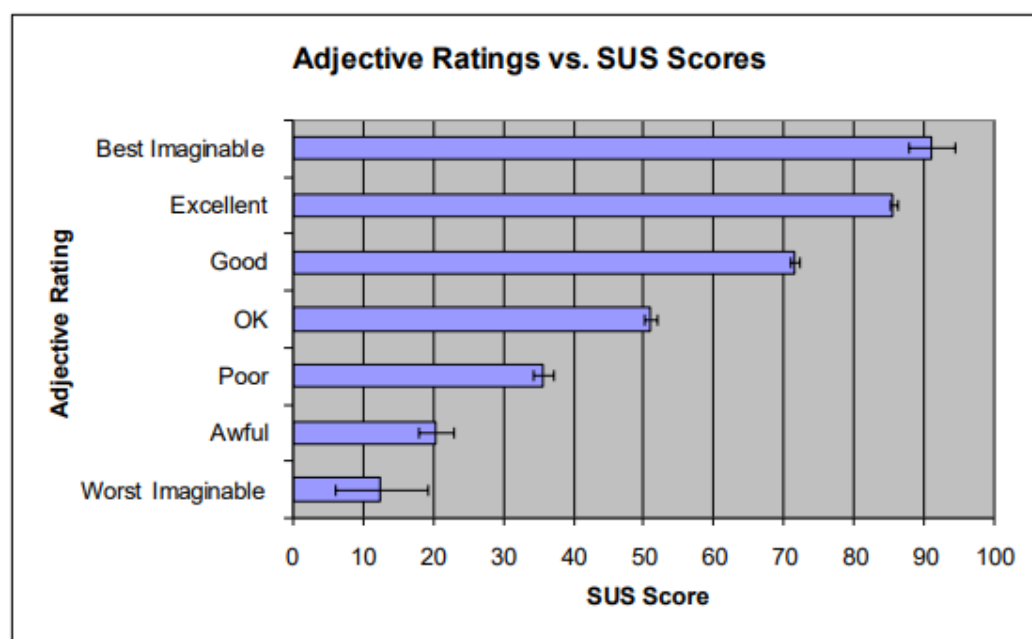
Berikut merupakan perhitungan nilai SUS dalam menentukan nilai SUS berdasarkan nilai dari pernyataan kuesioner SUS dapat dilihat pada persamaan 2.5, antara lain sebagai berikut (Brooke, 1986):

$$\text{Nilai SUS} = \sum_{i=1}^{ganjil=5} (K_{ganjil=i} - 1) + \sum_{j=1}^{genap=5} (5 - K_{genap=j}) \times 2,5 \quad (2.5)$$



Dari hasil kuesioner ini, akan didapatkan nilai berupa skor yang mana nilai ini mewakili dari *usability* atau kegunaan aplikasi keseluruhan aplikasi dengan rata-rata nilai SUS yang dapat disimpulkan berdasarkan Gambar 2.4. Berikut merupakan cara atau langkah yang digunakan untuk menghitung skor berdasarkan kuesioner SUS (Brooke, 1986):

1. Kontribusi skor dari setiap item kuesioner memiliki rentang 1 sampai 5.
2. Untuk item kuesioner nomor 1, 3, 5, 7 dan 9, kontribusi skor dihitung dengan posisi skala dikurangi 1.
3. Untuk item kuesioner nomor 2, 4, 6, 8 dan 10, kontribusi skor dihitung dengan 5 dikurangi posisi skala.
4. Kalikan jumlah skor dengan 2,5 untuk mendapatkan nilai secara keseluruhan.
5. Skor pada SUS memiliki kisaran yaitu 0 sampai 100.



Gambar 2.4 Adjective Rating Berdasarkan Skor SUS (System Usability Scale)

Sumber: Bangor, A., et al. (2009)



BAB 3 METODOLOGI PENELITIAN

Bagian metodologi penelitian pada penelitian skripsi ini menjelaskan cara sistematis untuk menentukan solusi sehingga mampu menyelesaikan permasalahan. Metodologi penelitian ini dijelaskan berdasarkan subbab-subbab yang terdiri atas tipe penelitian, dan strategi dan rancangan penelitian.

3.1 Tipe Penelitian

Tipe penelitian dari penelitian ini merupakan implementatif yang memiliki hasil akhir yaitu membuat produk utama. Produk digunakan sebagai solusi dari penyelesaian permasalahan berdasarkan latar belakang permasalahan dari penelitian ini. Dari latar belakang permasalahan, penelitian ini merupakan salah satu bentuk dari pengembangan atau *development* perangkat lunak. Pengembangan ini akan menghasilkan produk dengan menerapkan prinsip-prinsip dari rekayasa perangkat lunak, yang meliputi analisis, perancangan, implementasi, dan pengujian.

3.2 Strategi dan Rancangan Penelitian

Strategi dan rancangan penelitian pada penelitian skripsi ini terdiri atas strategi atau metode secara umum, subjek penelitian, lokasi penelitian, teknik pengumpulan data dan desain penelitian.

3.2.1 Strategi atau Metode Secara Umum

Metode yang digunakan di dalam penelitian ini yaitu metode *Design Sprint*. Metode *Design Sprint* merupakan metode yang digunakan untuk mengembangkan aplikasi yang hanya terfokus untuk menemukan, memahami dan mendefinisikan permasalahan-permasalahan yang dihadapi oleh pengguna. Metode ini difokuskan untuk mendapatkan prototipe tervalidasi yang mana prototipe ini akan dikembangkan menjadi aplikasi layak pakai bagi pengguna. Penerapan dari metode *Design Sprint* ini, dilakukan selama 5 hari pengerjaan bersama dengan tim pengembang. Tim pengembang untuk pengerjaan ini terdiri atas 9 orang dengan tujuan untuk mampu menemukan solusi dalam menyelesaikan permasalahan pengguna.

Adapun penerapan dari metode ini dilakukan melalui beberapa tahapan yaitu *understand*, *ideate*, *decide*, *prototype*, dan *test*. Penjelasan detail terkait dengan langkah-langkah pada tahapan pengerjaan yang akan dilakukan selama 5 hari, antara lain sebagai berikut:

1. Hari Pertama : Tahapan *Understand*

- Tim pengembang menentukan *long-term goal* atau tujuan jangka panjang dari pengembangan aplikasi yang akan dilakukan.



- Tim pengembang menentukan *sprint questions* atau kemungkinan-kemungkinan dan tantangan yang akan menyebabkan kegagalan dari pengembangan aplikasi yang dilakukan.
 - Tim pengembang membuat *project map* atau peta proyek berdasarkan pengalaman dari pengguna dengan tujuan untuk membuat *user journey map* dari pengguna. *User journey map* digunakan untuk mengetahui langkah-langkah yang dilakukan pengguna untuk dapat menyelesaikan permasalahan.
 - Tim pengembang melakukan *ask the expert* atau tanya ahli dengan tujuan untuk eksplorasi kebutuhan dari pengguna. Tahapan ini dilakukan dengan menggunakan teknik pengumpulan data yaitu wawancara atau *user interview* kepada subjek penelitian yaitu pengajar.
 - Tim pengembang menentukan kebutuhan-kebutuhan dari pengguna dengan membuat daftar *How Might We* (HMW) dari setiap masing-masing anggota tim pengembang.
 - Tim pengembang melakukan *affinity mapping* atau pemetaan afinitas terhadap daftar *How Might We* (HMW).
 - Tim pengembang melakukan *zen voting* untuk memilih kebutuhan-kebutuhan yang menjadi prioritas untuk dikembangkan.
 - Tim pengembang menetapkan metrik-metrik dengan menggunakan metode *Heart Metric*.
2. Hari Kedua : Tahapan *Ideate*
- Tim pengembang melakukan *lightning demos* yang bertujuan untuk merepresentasikan referensi dari kompetitor ataupun referensi lainnya.
 - Tim pengembang membuat sketsa untuk solusi dari aplikasi yang akan dikembangkan.
3. Hari Ketiga : Tahapan *Decide*
- Tim pengembang melakukan *sticky decision* yang bertujuan untuk merepresentasikan sketsa solusi yang sudah dibuat dan menentukan sketsa solusi yang akan dibuat menjadi prototipe.
 - Tim pengembang membuat *storyboard* secara utuh berdasarkan sketsa yang sudah ditentukan.
4. Hari Keempat : Tahapan *Prototype*
- Tim pengembang membuat prototipe yang *high-fidelity* berdasarkan hasil dari tahapan yang sudah dilakukan sebelumnya.
5. Hari Kelima : Tahapan *Test*
- Tim pengembang melakukan pengujian prototipe dengan menggunakan teknik pengumpulan data yaitu wawancara atau *user interview* kepada subjek penelitian yaitu pengajar. Pengujian ini dilakukan untuk



menentukan prototipe yang sudah tervalidasi untuk dapat diimplementasikan menjadi aplikasi layak pakai.

3.2.2 Subjek Penelitian

Subjek penelitian ini terdiri atas 2 subjek yaitu anak-anak dan pengajar. Anak-anak pada kegiatan sosial ini merupakan anak-anak terpilih untuk dapat disekolahkan di sekolah yang lebih layak melalui dana dari kegiatan sosial Komunitas Katalis Pendidikan. Anak-anak tersebut merupakan subjek penelitian yang merupakan pengguna akhir untuk menggunakan aplikasi yang akan dikembangkan di dalam penelitian ini. Selain itu, pengajar di dalam kegiatan sosial ini merupakan relawan mengajar dan guru yang mengajarkan anak-anak dalam meningkatkan kemampuan akademis belajar di sekolah. Pengajar tersebut merupakan subjek penelitian yang memberikan informasi dan memvalidasi aplikasi untuk dapat digunakan oleh anak dalam belajar secara efektif dan efisien.

Dalam menentukan subjek penelitian, penelitian ini menggunakan 2 teknik pengambilan sampel. Teknik pengambilan sampel terhadap pengajar dilakukan dengan menggunakan teknik pengambilan sampel acak sederhana atau *simple random sampling* sehingga memiliki kesempatan yang sama yaitu dengan mengambil 1 orang dari relawan mengajar pada Komunitas Katalis Pendidikan dan 1 orang guru dari SD Islam Mohammad Hatta. Selain itu, teknik pengambilan sampel terhadap 30 anak-anak pada kegiatan sosial dilakukan dengan menggunakan teknik pengambilan sampel yaitu sampel acak secara proporsi atau *propotional random sampling*. Perhitungan dari teknik perhitungan sampel untuk subjek penelitian yaitu anak-anak terdiri dari perhitungan estimasi besar sampel dan perhitungan teknik sampling. Adapun keterangan-keterangan untuk menentukan estimasi besar sampel, antara lain sebagai berikut:

$$N = 30$$

$$d = 0,1$$

Berikut merupakan perhitungan estimasi besar sampel, antara lain sebagai berikut:

$$n = \frac{N}{N(d)^2 + 1}$$

$$n = \frac{30}{30(0,1)^2 + 1}$$

$$n = \frac{30}{0,3 + 1}$$

$$n = \frac{30}{1,3}$$

$$n = 23,076 \approx 23$$



Adapun jumlah populasi setiap kelas pada anak-anak dari kegiatan sosial Komunitas Katalis Pendidikan dapat dilihat pada Tabel 3.1. Selain itu, keterangan-keterangan untuk menentukan jumlah partisipan yang akan diujikan dengan perhitungan teknik sampling, antara lain sebagai berikut:

$$N = 30$$

$$E = 23$$

Tabel 3.1 Jumlah Anak-Anak dari Setiap Kelas

Kelas	Jumlah Pengguna
1	7
2	7
3	5
4	4
5	4
6	3
Total	30

Berikut merupakan perhitungan teknik sampling untuk menentukan jumlah partisipan yang akan diujikan dengan menggunakan nilai estimasi besar sampel yang sudah dihitung sebelumnya, antara lain sebagai berikut:

$$n_1 = \frac{7}{30} \times 23 = 5,367 \approx 5$$

$$n_2 = \frac{7}{30} \times 23 = 5,367 \approx 5$$

$$n_3 = \frac{5}{30} \times 23 = 3,833 \approx 4$$

$$n_4 = \frac{4}{30} \times 23 = 3,067 \approx 3$$

$$n_5 = \frac{4}{30} \times 23 = 3,067 \approx 3$$

$$n_6 = \frac{3}{30} \times 23 = 2,3 \approx 2$$

Berdasarkan hasil perhitungan yang sudah dilakukan sebelumnya, didapatkan bahwa jumlah partisipan yaitu anak-anak pada kegiatan sosial Komunitas Katalis Pendidikan pada setiap kelasnya yaitu kelas 1 sampai kelas 6 terdiri dari 22 orang yang dapat dilihat pada Tabel 3.2.



Tabel 3.2 Jumlah Anak-Anak yang Akan Diujikan dari Setiap Kelas

Kelas	Jumlah Pengguna
1	5
2	5
3	4
4	3
5	3
6	2
Total	22

3.2.3 Lokasi Penelitian

Penelitian ini dilakukan di 2 lokasi yaitu lokasi dari Komunitas Katalis Pendidikan dan lokasi sekolah anak-anak pada kegiatan sosial Komunitas Katalis Pendidikan. Lokasi Komunitas Katalis Pendidikan terletak di Jl. Simpang Candi Panggung, Perumahan Garden Palma Inside Kavling 3, Mojolangu, Kota Malang, Jawa Timur. Selain itu, untuk lokasi sekolah yaitu Yayasan Bina Insan Kamil SD Islam Mohammad Hatta, Jl. Simpang Flamboyan No.30, Lowokwaru, Kota Malang, Jawa Timur.

3.2.4 Teknik Pengumpulan Data

Teknik pengumpulan data dilakukan untuk mendapatkan informasi yang dibutuhkan pada penelitian ini dimulai pada tahap awal sampai tahap akhir penelitian. Teknik pengumpulan data dari penelitian ini menggunakan 3 cara yaitu observasi, wawancara dan kuesioner.

3.2.4.1 Observasi

Observasi dilakukan untuk mendapatkan informasi pada 2 tahapan di dalam penelitian ini. Tahapan pertama dilakukannya observasi yaitu untuk mendapatkan informasi awal untuk analisis permasalahan dan studi literatur terkait dengan latar belakang permasalahan, landasan kepustakaan dari penelitian dan informasi dari penggunaan aplikasi perangkat bergerak berbasis *mobile*. Tahapan kedua yaitu observasi dilakukan pada tahap pengujian aplikasi yang mana bertujuan untuk mendapatkan informasi terkait dengan interaksi pengguna secara langsung terhadap aplikasi yang telah dikembangkan. Teknik pengumpulan data pada tahapan kedua ini dilakukan dengan menggunakan rekaman video pada saat pengguna berinteraksi secara langsung dengan aplikasi. Teknik pengumpulan data yaitu rekaman video dari penggunaan aplikasi dilakukan bersama dengan subjek penelitian yaitu anak-anak dengan langkah-langkah, antara lain sebagai berikut:

1. Menjelaskan informasi terkait dengan aplikasi yang dikembangkan dan langkah-langkah dari observasi yang akan dilakukan.



2. Memberikan informasi terkait dengan tugas-tugas yang harus diselesaikan dari aplikasi yang dikembangkan.

3. Anak-anak menyelesaikan tugas dari tugas pertama sampai tugas terakhir secara bertahap.

4. Pada saat observasi berlangsung, aktivitas dari anak-anak dalam penggunaan aplikasi yang dikembangkan akan direkam.

3.2.4.2 Wawancara

Wawancara dilakukan untuk memperoleh informasi secara langsung terhadap subjek penelitian. Pada penelitian ini, wawancara hanya dilakukan bersama dengan subjek penelitian yaitu pengajar secara individual. Hal ini bertujuan untuk mendapatkan hasil wawancara yang objektif dan proses wawancara yang berjalan dengan baik. Wawancara dilakukan dengan menanyakan beberapa pertanyaan yang bersifat *open-ended questions* yaitu pertanyaan-pertanyaan yang ditujukan kepada narasumber untuk mendapatkan jawaban secara luas. Terdapat dua tahapan dalam melakukan wawancara, tahap pertama dilakukan pada tahapan untuk menentukan permasalahan dan tahapan validasi prototipe dengan melibatkan subjek penelitian yaitu pengajar.

3.2.4.3 Kuesioner

Kuesioner dari penelitian ini dilakukan pada tahapan pengujian yaitu studi kegunaan aplikasi yang bertujuan untuk mendapatkan tanggapan pengguna terhadap aplikasi yang dikembangkan. Teknik pengumpulan data ini digunakan untuk mengukur tingkat kegunaan aplikasi berdasarkan aspek kepuasan pengguna setelah menggunakan aplikasi secara langsung. Kuesioner yang digunakan pada penelitian ini adalah *System Usability Scale* (SUS). Kuesioner SUS ini dibagikan kepada subjek penelitian yaitu anak-anak setelah menjalankan seluruh tugas untuk menggunakan aplikasi yang dikembangkan. Setelah itu, hasil dari kuesioner akan dihitung dan hasil perhitungannya akan dianalisis sehingga peneliti dapat mengetahui tingkat kegunaan aplikasi berdasarkan aspek kepuasan pengguna.

3.2.5 Desain Penelitian

Desain penelitian dari penelitian ini direpresentasikan melalui diagram blok yang mana merupakan alur dari penelitian yang akan dilakukan. Alur penelitian ini terdiri atas analisis permasalahan, studi literatur, pengumpulan data, analisis kebutuhan dan perancangan, implementasi, pengujian, dan penarikan kesimpulan dan saran. Alur penelitian dapat dilihat pada Gambar 3.1 yang memiliki penjelasan secara lengkap, antara lain sebagai berikut:

3.2.5.1 Analisis Permasalahan

Analisis permasalahan bertujuan untuk mengidentifikasi permasalahan yang terjadi terhadap pengguna. Analisis permasalahan difokuskan untuk permasalahan pembelajaran bahasa Inggris bagi anak-anak yang masih belum



dilaksanakan secara efektif dan efisien. Data-data untuk analisis permasalahan ini didapatkan melalui teknik pengumpulan data yaitu observasi dan wawancara.

3.2.5.2 Studi Literatur

Studi Literatur bertujuan untuk mengumpulkan informasi-informasi serta teori-teori yang mana dalam hal ini menjadi suatu pendukung di dalam menyelesaikan penelitian. Studi literatur didapatkan melalui referensi-referensi seperti, jurnal-jurnal penelitian sebelumnya, buku maupun artikel yang terdapat di internet. Adapun studi literatur yang digunakan yaitu membahas terkait dengan hal pembelajaran bahasa Inggris bagi anak melalui permainan edukatif, metode *Design Sprint*, Android, pengujian validasi, pengujian kegunaan aplikasi atau *usability testing* berdasarkan parameter pengukuran yaitu efisiensi, efektivitas dan kepuasan pengguna.

3.2.5.3 Pengumpulan Data

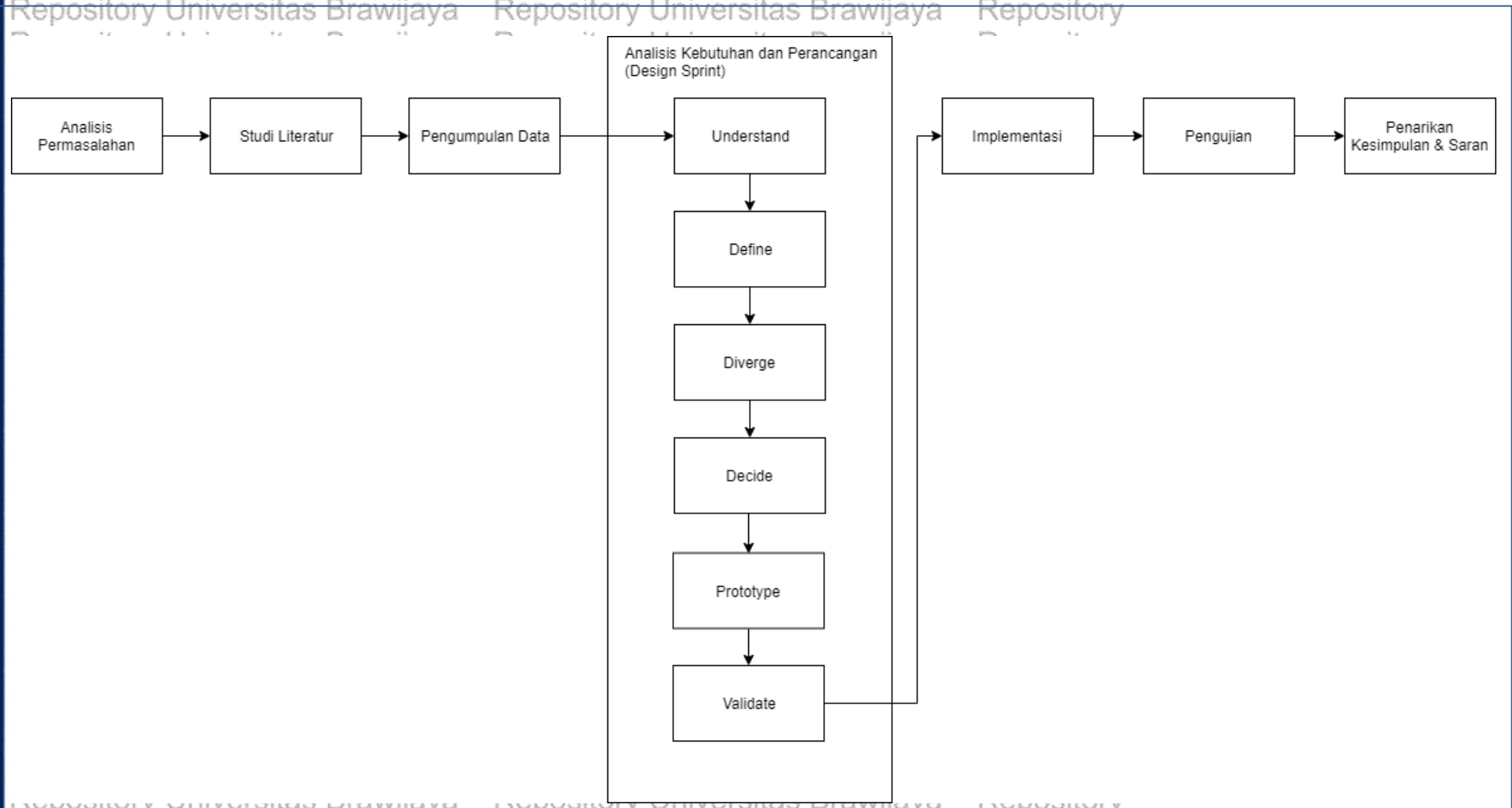
Tahapan pengumpulan data dilakukan dengan tujuan untuk mengumpulkan data yang nantinya akan dilakukan analisis terhadap data-data tersebut. Pada tahapan ini, peneliti menggunakan 3 teknik pengumpulan data yaitu observasi, wawancara dan kuesioner. Tahapan pengumpulan data ini dilakukan bersama dengan subjek penelitian yaitu pengajar dan anak-anak.

3.2.5.4 Analisis Kebutuhan dan Perancangan

Tahapan analisis kebutuhan dan perancangan dilakukan dengan tujuan untuk mampu menganalisis data-data yang didapatkan dari tahapan pengumpulan data untuk dijadikan kebutuhan-kebutuhan dari aplikasi yang akan dikembangkan. Selain itu, peneliti akan melakukan perancangan berdasarkan hasil analisis yang sudah didapatkan. Hasil dari tahapan ini ialah berupa prototipe yang sudah tervalidasi sehingga selanjutnya dapat diimplementasikan untuk menjadi aplikasi yang layak pakai. Pada tahapan analisis kebutuhan dan perancangan, terdiri atas 6 fase yaitu *understand*, *define*, *sketch*, *decide*, *prototype*, dan *validate*. Fase-fase tersebut merupakan penerapan dari metode *Design Sprint* yang pengerjaannya dilakukan selama 5 hari.

3.2.5.5 Implementasi

Pada tahapan implementasi, hasil analisis dan perancangan di tahapan sebelumnya yaitu berupa prototipe yang sudah tervalidasi akan digunakan sebagai model pengembangan selanjutnya untuk dapat diimplementasikan. Tahap implementasi ini dilakukan dengan mengubah hasil perancangan menjadi implementasi basis data, algoritma dan antarmuka. Implementasi algoritma dan implementasi antarmuka dilakukan dengan menggunakan bahasa pemrograman Java untuk mengatur alur logika dan XML untuk implementasi antarmuka dari aplikasi. Implementasi ini dilakukan dengan menggunakan aplikasi untuk pengembangan yaitu IDE (*Integrated Development Environment*) dari Android Studio 3.4.1.



Gambar 3.1 Alur Penelitian Metodologi Pengembangan Sistem



3.2.5.6 Pengujian

Pada tahapan pengujian ini merupakan tahapan yang dilakukan untuk pengujian *black box* dan studi kegunaan aplikasi. Pengujian *black box* dilakukan untuk mengetahui apakah kebutuhan-kebutuhan fungsional dari aplikasi yang dikembangkan sudah sesuai dengan kebutuhan pengguna. Pengujian *black box* akan dilakukan dengan menggunakan teknik *Equivalence Partitioning*, sedangkan studi kegunaan aplikasi dilakukan dengan pengujian *usability* atau kegunaan aplikasi yang dilakukan untuk mengukur tingkat kegunaan aplikasi berdasarkan parameter pengukuran efektivitas, efisiensi dan kepuasan pengguna.

Pengujian *black box* dilakukan terhadap kebutuhan-kebutuhan fungsionalitas dari aplikasi yang sudah dikembangkan. Hal ini bertujuan untuk mengetahui apakah seluruh kebutuhan-kebutuhan fungsionalitas aplikasi sudah berjalan dengan baik, sehingga hal ini menunjukkan bahwa aplikasi sudah memenuhi seluruh kebutuhan-kebutuhan dari pengguna. Pengujian ini dilakukan dengan teknik *Equivalence Partitioning* yaitu melakukan partisi-partisi terhadap *input* dari pengujian yang akan dilakukan. Ketika kebutuhan-kebutuhan fungsionalitas sudah diujikan maka pengujian tersebut memiliki status yaitu sudah tervalidasi.

Pengujian *usability* atau kegunaan aplikasi dilakukan untuk mengetahui tingkat kegunaan aplikasi yang dikembangkan berdasarkan studi kegunaan aplikasi yang dilakukan. Dalam mengukur tingkat kegunaan aplikasi ini, pengujian diukur berdasarkan hasil observasi yang diujikan kepada pengguna akhir yaitu berupa video dari penggunaan aplikasi dan hasil dari kuesioner. Studi kegunaan aplikasi dilakukan dengan menggunakan parameter pengukuran yaitu efektivitas, efisiensi dan kepuasan pengguna. Parameter pengukuran efektivitas dan efisiensi diukur berdasarkan hasil rekaman video, sedangkan parameter pengukuran kepuasan pengguna diukur dengan menggunakan hasil kuesioner.

3.2.5.7 Penarikan Kesimpulan dan Saran

Tahapan penarikan kesimpulan akan dilakukan ketika semua tahapan yaitu analisis permasalahan, studi literatur, pengumpulan data, analisis kebutuhan dan perancangan, implementasi, dan pengujian aplikasi sudah selesai dilakukan. Selain itu, saran merupakan pernyataan yang diberikan oleh peneliti dengan tujuan untuk mampu mengevaluasi, memperbaiki serta menjadikannya pertimbangan untuk penelitian selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

Pada bagian analisis kebutuhan pada penelitian skripsi ini akan dijelaskan terkait dengan tahapan-tahapan yang dilakukan untuk menganalisis kebutuhan-kebutuhan dari aplikasi yang akan dikembangkan. Analisis kebutuhan dilakukan dengan menggunakan metode *Design Sprint* pada hari pertama yaitu tahapan *understand*. Di dalam bagian ini juga terdapat pembahasan terkait dengan identifikasi aktor, analisis kebutuhan perangkat lunak, identifikasi kebutuhan yang terdiri atas kebutuhan fungsional dan non fungsional serta pemodelan dari kebutuhan.

4.1 Identifikasi Aktor

Dari tahap identifikasi aktor ini memiliki tujuan untuk memberikan gambaran terkait dengan pengguna akhir yang akan menggunakan aplikasi Reado. Pengidentifikasian ini terdiri atas nama aktor dan deskripsi dari aktor. Adapun aktor di dalam pengembangan aplikasi ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 Aktor dari Aplikasi Reado

Nama Aktor	Deskripsi Aktor
Anak-anak	Aktor umum yang menggunakan sistem. Aktor dapat masuk ke dalam sistem tanpa melakukan <i>login</i> atau tanpa harus terintegrasi dengan <i>email</i> .

4.2 Identifikasi Kebutuhan Perangkat Lunak

Di dalam mengidentifikasikan kebutuhan-kebutuhan perangkat lunak, penelitian ini menggunakan metode *Design Sprint* yang dilakukan pada hari pertama yaitu tahapan *understand*. Tahapan ini dilakukan untuk menggali atau mendapatkan kebutuhan-kebutuhan terhadap aplikasi yang akan dikembangkan berdasarkan kebutuhan dari pengguna. Adapun penjelasan terkait dengan langkah-langkah pada tahapan *understand* yang dilakukan untuk mengidentifikasikan kebutuhan perangkat lunak pada pengembangan aplikasi Reado, antara lain sebagai berikut:

4.2.1 Menentukan *Long-Term Goal*

Pada tahapan awal dalam menentukan kebutuhan-kebutuhan dari pengguna, dibutuhkan suatu tujuan sebagai acuan dasar dari pengembangan aplikasi ini. Di dalam metode *Design Sprint*, tujuan dari pengembangan aplikasi disebut sebagai *long-term goal* atau tujuan jangka panjang. *Long-term goal* dari pengembangan aplikasi Reado ini adalah membangun aplikasi untuk permainan edukatif bagi anak melalui perangkat bergerak berbasis *mobile* atau *smartphone* yang mana mampu untuk membantu anak dalam mempelajari Bahasa Inggris dengan efektif dan efisien.



4.2.2 Menentukan *Sprint Questions*

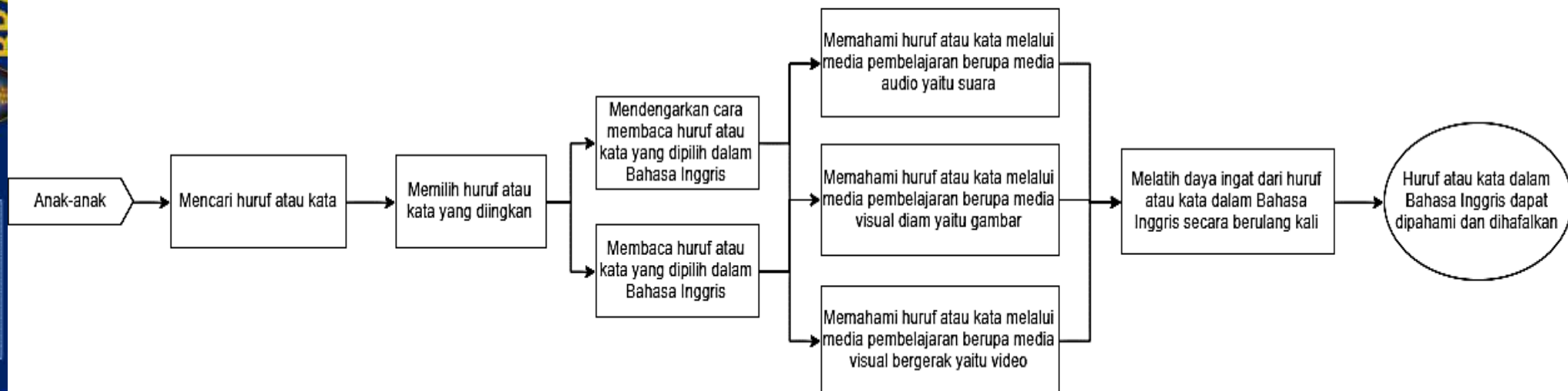
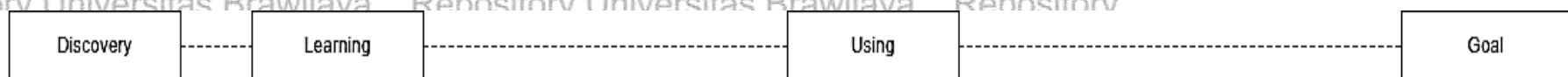
Tahapan selanjutnya yaitu menentukan *sprint questions* atau tantangan-tantangan yang dapat memungkinkan terjadinya kegagalan dari pengembangan aplikasi Reado ini yang mana dituliskan dalam bentuk pertanyaan-pertanyaan. Tahapan ini dilakukan berdasarkan hasil diskusi bersama dengan tim pengembang. Adapun *sprint questions* dari pengembangan aplikasi Reado dapat dilihat pada Tabel 4.2.

Tabel 4.2 *Sprint Questions* dari Pengembangan Aplikasi Reado

No.	<i>Sprint Questions</i>
1.	Dapatkah kita untuk membuat aplikasi yang dapat membantu anak-anak untuk mempelajari Bahasa Inggris secara efektif dan efisien?
2.	Dapatkah kita untuk membuat aplikasi permainan edukatif bagi anak untuk mempelajari Bahasa Inggris?
3.	Dapatkah kita untuk membuat aplikasi yang dapat digunakan oleh anak-anak untuk menarik kemauan anak dalam mempelajari Bahasa Inggris?
4.	Dapatkah kita untuk membuat aplikasi yang tidak menimbulkan efek negatif bagi anak seperti ketergantungan anak dalam bermain melalui aplikasi perangkat bergerak berbasis <i>mobile</i> atau <i>smartphone</i> ?

4.2.3 Membuat *Project Map*

Tahapan selanjutnya yaitu membuat *project map* atau peta proyek berdasarkan pengalaman dari pengguna dengan tujuan untuk membuat *user journey map* dari pengguna. *User journey map* digunakan untuk mengetahui langkah-langkah yang dilakukan pengguna untuk dapat menyelesaikan permasalahan. *Project maps* terbagi menjadi empat bagian yaitu *discovery*, *learning*, *using* dan *goal*. Bagian *discovery* merupakan langkah awal pengguna pertama kali dalam melakukan aktivitas sesuai dengan tujuan untuk memperoleh informasi yang diinginkan yaitu dari pengembangan aplikasi ini adalah mencari kata-kata dalam Bahasa Inggris. *Learning* merupakan langkah pengguna untuk mempelajari sebelum melakukan atau menggunakan sesuatu demi tercapainya tujuan yang ingin dicapai yaitu dari pengembangan ini adalah melakukan seleksi atau memilih kata-kata dalam Bahasa Inggris. Bagian *using* merupakan langkah utama yang harus dilakukan pengguna untuk meraih tujuan yang ingin dicapai. Bagian yang terakhir yaitu *goal* merupakan tujuan akhir yang ingin diraih atau dicapai oleh pengguna. Tujuan akhir atau goal dari pengguna di dalam pengembangan aplikasi Reado ini adalah kata-kata yaitu huruf atau kata dalam Bahasa Inggris dapat dipahami dan dihafalkan oleh anak. Adapun *project maps* dari pengembangan aplikasi Reado dapat direpresentasikan di dalam gambar yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Project Map dari User Journey Anak-anak dalam Mempelajari Kata Bahasa Inggris



4.2.4 Melakukan *Ask The Expert*

Tahapan untuk mendapatkan kebutuhan-kebutuhan dari pengguna yaitu melakukan *ask the expert* atau tanya ahli dengan tujuan untuk eksplorasi kebutuhan pengguna. Tahapan ini merupakan tahapan untuk mendapatkan informasi dari pengguna melalui wawancara atau *user interview* kepada subjek penelitian yaitu pengajar. Wawancara dilakukan dengan menanyakan pertanyaan terkait dengan permasalahan yang dihadapi oleh anak-anak dalam mempelajari Bahasa Inggris dan metode atau cara yang saat digunakan untuk mengajarkan anak-anak. Adapun kesimpulan berdasarkan hasil wawancara yang dilakukan kepada pengajar yaitu dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Wawancara dari Pengajar

No.	Subjek Penelitian (Pengajar)	Hasil Wawancara
1.	Guru Sekolah dari SDI Moh Hatta	1. Anak-anak sulit untuk memahami kata-kata dalam Bahasa Inggris. 2. Anak-anak sulit untuk membaca kata-kata dalam Bahasa Inggris. 3. Anak-anak sering kali mendengarkan atau membaca kata-kata Bahasa Inggris dalam permainan atau game sehingga mengetahui dan memahami kata tersebut. 4. Materi pembelajaran yang diajarkan yaitu membaca, menulis, dan belajar komunikasi dalam Bahasa Inggris. 5. Metode Pengajaran melalui buku, gambar-gambar dan permainan. 6. Sebuah permainan sangat membantu anak dalam mempelajari Bahasa Inggris, karena dengan bermain anak-anak dapat lebih cepat untuk mempelajarinya. 7. Pengajaran kepada anak juga menggunakan perangkat elektronik seperti komputer dan smartphone. Pengajaran ini bertujuan untuk membangun karakter pelajar yang melek terhadap teknologi, untuk meningkatkan pemahaman anak terhadap pembelajaran, untuk meningkatkan rasa penasaran yang tinggi pada anak untuk belajar hal yang baru.



Tabel 4.3 Hasil Wawancara dari Pengajar (lanjutan)

No.	Subjek Penelitian (Pengajar)	Hasil Wawancara
2.	Relawan Mengajar dari Komunitas Katalis Pendidikan	1. Kegiatan sosial yang diadakan oleh Komunitas Katalis Pendidikan bertujuan untuk mengajarkan dan mendidik anak-anak yang membutuhkan sehingga dapat mendapatkan dana bantuan untuk sekolah di sekolah yang terbaik. 2. Bagi anak-anak, mata pelajaran yang sulit untuk dipahami yaitu Bahasa Inggris karena bukan merupakan bahasa ibu seperti Bahasa Indonesia ataupun Bahasa Jawa sehingga sulit untuk mempelajari dan memahaminya. 3. Anak-anak sulit untuk membaca dan memahami arti dari kata-kata dalam Bahasa Inggris. 4. Materi pembelajaran yang digunakan yaitu membaca cerita-cerita seperti dongeng yang dituliskan dalam Bahasa Inggris dan video-video pembelajaran Bahasa Inggris. 5. Metode pengajaran yang diterapkan yaitu belajar melalui buku, menonton video pembelajaran dan permainan. 6. Pihak Komunitas tidak menggunakan perangkat elektronik dikarenakan kekurangan perangkat elektronik yang dibutuhkan sehingga diharapkan anak-anak dapat bersekolah di sekolah terbaik yang memiliki fasilitas yang lengkap dan menerapkan sistem <i>smart class</i> untuk anak-anak melalui perangkat elektronik.

4.2.5 Membuat Daftar *How Might We* (HMW)

Berdasarkan hasil observasi dan wawancara yang dilakukan sebelumnya, maka dari itu kebutuhan-kebutuhan aplikasi berdasarkan kebutuhan pengguna dapat didefinisikan oleh tim pengembang. Kebutuhan-kebutuhan ini dibuat menjadi sebuah daftar *How Might We* (HMM) yang didefinisikan oleh masing-masing dari tim pengembang. Hal ini bertujuan untuk melakukan eksplorasi kebutuhan-kebutuhan aplikasi berdasarkan hasil observasi dan wawancara dengan melalui ide-ide kreatif berdasarkan hasil pemikiran masing-masing dari tim pengembang. Dari penelitian ini, daftar HMW yang didefinisikan oleh masing-masing anggota tim



pengembang terdiri dari 27 hasil HMW. Adapun daftar HMW yang didefinisikan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Daftar *How Might We* dari Tim Pengembang Aplikasi Reado

No.	<i>How Might We</i> (HMW)
1.	Visualisasi pembelajaran kata dalam Bahasa Inggris dengan menggunakan gambar dan video pembelajaran.
2.	Pembelajaran kata dalam Bahasa Inggris dibuat dalam bentuk permainan atau <i>game</i> dengan menebak kata-kata dalam Bahasa Inggris.
3.	Pembelajaran kata dalam Bahasa Inggris dapat dilakukan dengan menangkap kata-kata yang dapat dijumpai oleh anak melalui buku, poster, majalah dan media lainnya.
4.	Permainan edukatif untuk menebak kata berdasarkan suara pelafalan dari kata tersebut.
5.	Permainan edukatif untuk menebak kata berdasarkan visualisasi berupa gambar dari kata tersebut.
6.	Aplikasi yang dapat menyimpan kata-kata dalam Bahasa Inggris yang sudah dipelajari oleh anak-anak.
7.	Anak-anak dapat mempelajari cara pelafalan atau pembacaan dari kata yang dipelajari.
8.	Aplikasi yang dapat merekam kata-kata dalam Bahasa Inggris yang terdapat di lingkungan sekitar anak.
9.	Aplikasi yang dapat menyimpan kata Bahasa Inggris dan dapat dilakukan secara berulang sehingga anak dapat paham dan hafal dengan kata tersebut.
10.	Aplikasi dapat menuliskan kata-kata dalam Bahasa Inggris.
11.	Aplikasi yang dapat menunjukkan cara pembacaan dari kata Bahasa Inggris yang ingin dipelajari oleh anak.
12.	Kata-kata dalam Bahasa Inggris yang sudah dipelajari oleh anak dapat digunakan untuk permainan edukatif tebak kata dalam Bahasa Inggris.
13.	Permainan edukatif yang menampilkan video pembelajaran kata berupa video untuk menebak kata-kata dalam video tersebut.
14.	Permainan edukatif yang dapat menampilkan suara berupa cara pembacaan atau pelafalan dari kata-kata tertentu dalam Bahasa Inggris.
15.	Permainan edukatif yang menampilkan gambar yaitu gambar yang dapat mewakili kata-kata tertentu dalam Bahasa Inggris.
16.	Aplikasi dapat melatih anak untuk menuliskan kata dalam Bahasa Inggris.
17.	Aplikasi dapat melakukan <i>scan</i> atau deteksi kata yang terdapat di lingkungan sekitar anak, di dalam buku, poster ataupun majalah.



Tabel 4.4 Daftar *How Might We* dari Tim Pengembang Aplikasi Reado (lanjutan)

No.	<i>How Might We</i> (HMW)
18.	Aplikasi dapat menyimpan kata-kata yang sudah dideteksi sehingga anak dapat mengulang untuk mempelajari kata tersebut.
19.	Aplikasi yang menyediakan cara pembacaan dalam Bahasa Inggris bagi anak untuk melatih pemahaman anak terhadap cara pelafalan kata-kata dalam Bahasa Inggris.
20.	Aplikasi menyediakan cara pembacaan dalam bentuk suara dan tulisan pelafalan dari kata-kata yang dipelajari oleh anak.
21.	Aplikasi menyediakan permainan berdialog untuk melatih komunikasi anak dalam Bahasa Inggris.
22.	Permainan edukatif menggunakan visualisasi berupa suara, gambar dan video.
23.	Permainan edukatif diberikan beberapa <i>level</i> kesulitan untuk melatih ketertarikan anak terhadap kata-kata dalam Bahasa Inggris.
24.	Aplikasi menggunakan teknologi <i>Augmented Reality</i> (AR) untuk memvisualisasikan kata-kata dalam Bahasa Inggris sehingga anak dapat lebih tertarik untuk mempelajari Bahasa Inggris.
25.	Aplikasi menggunakan gambar-gambar yang menarik perhatian anak yaitu berupa gambar-gambar dengan banyak warna yang menarik.
26.	Aplikasi menyediakan daftar kata-kata dalam Bahasa Inggris yang sudah dipelajari oleh anak.
27.	Daftar kata-kata yang sudah dipelajari dapat diulang kembali dengan memainkan permainan untuk mengulang kata tersebut.

4.2.6 Melakukan *Affinity Mapping*

Tahapan selanjutnya yaitu tim pengembang melakukan *affinity mapping* terhadap daftar *How Might We* (HMW). Tahapan ini merupakan tahapan untuk mengelompokkan hasil-hasil *How Might We* (HMW) yang memiliki kemiripan atau kesamaan. Pengelompokkan daftar *How Might We* (HMW) yang sudah didefinisikan sebelumnya dapat memudahkan tim pengembang untuk menentukan kebutuhan-kebutuhan yang akan dikembangkan pada aplikasi Reado. Tahapan ini dilakukan dengan memaparkan daftar *How Might We* (HMW) dan berdiskusi bersama tim pengembang untuk mengelompokkan daftar *How Might We* (HMW) tersebut. Berdasarkan hasil *How Might We* (HMW) dari tahapan *affinity mapping* ini, didapatkan dua belas (12) daftar hasil *How Might We* (HMW) yang sudah digabungkan atau dikelompokkan berdasarkan kemiripan atau kesamaan dari masing-masing daftar *How Might We* (HMW) yang mana dapat dilihat pada Tabel 4.5.

Tabel 4.5 Daftar *How Might We* dari Hasil *Affinity Mapping*

No.	<i>How Might We</i> (HMW)
1.	Pembelajaran kata dalam Bahasa Inggris dapat dilakukan dengan mendeteksi atau menangkap kata-kata yang dapat dijumpai oleh anak melalui lingkungan sekitar, buku, poster, majalah dan media lainnya.
2.	Permainan edukatif untuk menebak kata berdasarkan suara pelafalan dari kata tersebut.
3.	Permainan edukatif untuk menebak kata berdasarkan visualisasi berupa gambar yang dapat mewakili dari kata tersebut.
4.	Aplikasi yang dapat menyimpan kata-kata dalam Bahasa Inggris yang sudah dideteksi dan dapat dilakukan secara berulang sehingga anak dapat paham dan hafal dengan kata tersebut.
5.	Aplikasi yang dapat menunjukkan cara pembacaan dari kata dalam Bahasa Inggris yang ingin dipelajari oleh anak dalam bentuk suara dan tulisan.
6.	Aplikasi dapat menuliskan kata-kata dalam Bahasa Inggris.
7.	Kata-kata dalam Bahasa Inggris yang sudah dipelajari oleh anak dapat digunakan untuk permainan edukatif tebak kata dalam Bahasa Inggris.
8.	Permainan edukatif yang menampilkan video pembelajaran kata berupa video untuk menebak kata-kata dalam video tersebut.
9.	Aplikasi menyediakan permainan berdialog untuk melatih komunikasi anak dalam Bahasa Inggris.
10.	Permainan edukatif diberikan beberapa <i>level</i> kesulitan untuk melatih ketertarikan anak terhadap kata-kata dalam Bahasa Inggris.
11.	Aplikasi menggunakan teknologi <i>Augmented Reality</i> (AR) untuk memvisualisasikan kata-kata Bahasa Inggris sehingga anak dapat lebih tertarik untuk mempelajari Bahasa Inggris.
12.	Aplikasi menggunakan gambar-gambar yang menarik perhatian anak yaitu berupa gambar-gambar dengan banyak warna yang menarik.

4.2.7 Melakukan *Zen Voting*

Tahapan dalam melakukan *zen voting* ini merupakan tahapan untuk menyeleksi dan menentukan kebutuhan-kebutuhan aplikasi yang akan dikembangkan berdasarkan daftar HMW yang sudah didefinisikan sebelumnya. Masing-masing dari tim pengembang akan melakukan voting untuk menentukan hasil HWM yang memiliki prioritas untuk dikembangkan di dalam aplikasi. Tahapan ini dilakukan dengan cara berdiskusi bersama tim pengembang untuk menentukan kebutuhan-kebutuhan dari aplikasi. Hasil tahapan *zen voting* ini adalah kebutuhan-kebutuhan pengguna yang akan dianalisis untuk dijadikan kebutuhan-kebutuhan dari aplikasi *Reado* yang akan dikembangkan. Adapun hasil *zen voting* yaitu dapat dilihat pada Tabel 4.6.



Tabel 4.6 Daftar How Might We dari Hasil Zen Voting

No.	How Might We (HMW)
1.	Pembelajaran kata dalam Bahasa Inggris dapat dilakukan dengan mendeteksi dan menangkap kata-kata yang dapat dijumpai oleh anak melalui lingkungan sekitar, buku, poster ataupun majalah.
2.	Permainan edukatif untuk menebak kata berdasarkan suara pelafalan dari kata tersebut.
3.	Permainan edukatif untuk menebak kata berdasarkan visualisasi berupa gambar yang dapat mewakili dari kata tersebut.
4.	Aplikasi yang dapat menyimpan kata-kata dalam Bahasa Inggris yang sudah dideteksi dan dapat dilakukan secara berulang sehingga anak dapat paham dan hafal dengan kata tersebut.
5.	Aplikasi yang dapat menunjukkan cara pembacaan dari kata dalam Bahasa Inggris yang ingin dipelajari oleh anak dalam bentuk suara dan tulisan.
6.	Aplikasi dapat menuliskan kata-kata dalam Bahasa Inggris.
7.	Kata-kata dalam Bahasa Inggris yang sudah dipelajari oleh anak dapat digunakan untuk permainan edukatif tebak kata dalam Bahasa Inggris.
8.	Permainan edukatif yang menampilkan video pembelajaran kata berupa video untuk menebak kata-kata dalam video tersebut.
9.	Permainan edukatif diberikan beberapa <i>level</i> kesulitan untuk melatih ketertarikan anak terhadap kata-kata dalam Bahasa Inggris.

4.2.8 Menentukan Metrik-Metrik

Pada tahapan menentukan metrik ini, dilakukan dengan menggunakan metode *HEART metric*. Metode *HEART metric* ini memiliki lima (5) kategori yaitu *Happiness* (Kebahagiaan), *Engagement* (Keterikatan), *Adoption* (Adopsi), *Retention* (Penyimpanan) dan *Task Success* (Tugas Sukses). Tahapan ini bertujuan untuk mendapatkan metrik-metrik keberhasilan yang menjadi tolak ukur untuk pengembangan aplikasi *Reado* yang didapatkan atau ditentukan berdasarkan pengalaman pengguna. Tahapan ini juga dilakukan dengan berdiskusi bersama dengan tim pengembang untuk menentukan metrik-metrik keberhasilan tersebut. Dari setiap masing-masing kategori akan dispesifikasikan berdasarkan parameter-parameter tertentu yaitu *Goals* (Tujuan), *Signals* (Penanda) dan *Metrics* (Metrik). Parameter *goals* atau tujuan merupakan parameter pengukuran untuk mengukur tujuan yang ingin dicapai dari pengembangan aplikasi yang dilakukan. Selain itu, parameter *signals* atau penanda digunakan untuk menandakan bahwa tujuan yang didefinisikan yaitu sudah tercapai, sedangkan parameter *metrics* merupakan data-data yang dapat dianalisis untuk menentukan keberhasilan dari pengembangan aplikasi *Reado* yang dilakukan. Adapun metrik-metrik keberhasilan pada pengembangan aplikasi *Reado* dengan menggunakan metode *HEART Metric* ini dapat dilihat pada Tabel 4.7.


Tabel 4.7 Hasil Penentuan Metrik dengan Menggunakan *HEART Metric*

Kategori	Goals (Tujuan)	Signals	Metrics (Metrik)
Happiness (Kebahagiaan)	Pengguna dapat dengan mudah menggunakan aplikasi	Tingkat kegunaan aplikasi dari parameter kepuasan pengguna bernilai tinggi	Hasil kuesioner SUS (<i>System Usability Scale</i>)
Engagement (Keterikatan)	Pengguna dapat tertarik untuk menggunakan aplikasi	Tingkat kegunaan aplikasi dari parameter kepuasan pengguna bernilai tinggi	Hasil kuesioner SUS (<i>System Usability Scale</i>)
Adoption (Adopsi)	Pengguna dapat tertarik untuk menggunakan aplikasi	Tingkat kegunaan aplikasi dari parameter efisiensi bernilai tinggi	Hasil perhitungan dari tingkat kegunaan aplikasi dari parameter efisiensi
Retention (Penyimpanan)	Pengguna dapat tertarik untuk menggunakan aplikasi kembali	Tingkat kegunaan aplikasi dari parameter kepuasan pengguna bernilai tinggi	Hasil kuesioner SUS (<i>System Usability Scale</i>)
Task Success (Tugas Sukses)	Pengguna dapat mencapai tujuan dengan menggunakan aplikasi	Tingkat kegunaan aplikasi dari parameter efektivitas bernilai tinggi	Hasil perhitungan dari tingkat kegunaan aplikasi dari parameter efektivitas

4.3 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak ini dilakukan untuk menentukan kebutuhan-kebutuhan perangkat lunak yang akan dikembangkan di dalam aplikasi Reado. Dalam menentukan kebutuhan-kebutuhan dari aplikasi Reado, penentuan ini dilakukan berdasarkan hasil identifikasi aktor dan identifikasi kebutuhan perangkat lunak sebelumnya. Adapun kebutuhan-kebutuhan berdasarkan hasil

analisis terbagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional di dalam sistem merupakan kebutuhan-kebutuhan yang mendeskripsikan apa saja yang terdapat di dalam sistem yaitu berupa layanan ataupun kemampuan dari sistem. Setiap kebutuhan-kebutuhan fungsional ditandai dengan kode penomoran yang mana kode tersebut terdiri atas 3 format. Format pertama yaitu berupa huruf SRS yang merupakan singkatan dari *Software Requirement Specifications*, format kedua yaitu berupa huruf RD yang merupakan singkatan untuk aplikasi Reado dan format ketiga yaitu berupa nomor yang merupakan urutan dari kebutuhan fungsional sistem. Adapun kebutuhan-kebutuhan fungsional dari aplikasi Reado dapat dilihat pada Tabel 4.8.

Tabel 4.8 Kebutuhan Fungsional dari Aplikasi Reado

No.	Kode	Nama Fungsi	Deskripsi Kebutuhan
1.	SRS-RD-01	Mendeteksi Kata	Sistem harus mampu menyediakan mekanisme untuk mendeteksi kata-kata yang dapat ditangkap oleh kamera secara <i>realtime</i> atau waktu sebenarnya.
2.	SRS-RD-02	Menampilkan Kata-Kata dari Hasil Deteksi Kata	Sistem harus mampu menampilkan kata-kata dari hasil deteksi kata dan ditampilkan dalam bentuk tulisan serta cara pembacaan kata-kata tersebut dalam bentuk suara.
3.	SRS-RD-03	Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	Sistem harus mampu menampilkan alfabet dari kata berdasarkan hasil deteksi kata yang dipilih dan ditampilkan dalam bentuk tulisan serta cara pembacaan alfabet tersebut dalam bentuk suara dan tulisan.
4.	SRS-RD-04	Menentukan Tahapan Permainan	Sistem harus mampu menyediakan mekanisme untuk menentukan tahapan permainan yang terdiri atas latihan, tebak kata berdasarkan gambar, tebak kata berdasarkan suara dan tebak kata berdasarkan video.
5.	SRS-RD-05	Menentukan Level Permainan	Sistem harus mampu menyediakan mekanisme untuk menentukan <i>level</i> permainan berdasarkan pilihan aktor yang terdiri atas <i>level</i> pada tahapan permainan tebak kata berdasarkan gambar, suara dan video.



Tabel 4.8 Kebutuhan Fungsional dari Aplikasi Reado (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi Kebutuhan
6.	SRS-RD-06	Menampilkan Permainan Tahapan Latihan	Sistem harus mampu menampilkan permainan tahapan latihan. Tahapan latihan untuk mengulang kata-kata yang sudah dideteksi sebelumnya dalam bentuk permainan tebak kata berdasarkan suara dari pelafalan kata yang harus ditebak.
7.	SRS-RD-07	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan gambar. Tahapan tebak kata berdasarkan gambar untuk menebak kata dari gambar yang memvisualisasikan kata yang harus ditebak.
8.	SRS-RD-08	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan suara. Tahapan tebak kata berdasarkan suara untuk menebak kata dari suara pelafalan kata yang harus ditebak.
9.	SRS-RD-09	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan video. Tahapan tebak kata berdasarkan video untuk menebak kata dari video yang memvisualisasikan kata yang harus ditebak.
10.	SRS-RD-10	Menyediakan Mekanisme untuk Jawaban dari Permainan	Sistem harus mampu menyediakan mekanisme untuk jawaban dan menentukan jawaban yang benar atau salah dari permainan disetiap tahapan. Jawaban yang harus dipilih terdiri atas alfabet-alfabet dari kata yang harus ditebak.

4.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari pengembangan aplikasi Reado ini merupakan kebutuhan-kebutuhan yang bukan merupakan layanan atau kemampuan yang berhubungan secara langsung dengan pengguna. Namun, kebutuhan ini dapat menentukan kepuasan dan kualitas pengguna terhadap penggunaan dari aplikasi.



Adapun kebutuhan non fungsional dari pengembangan sistem ini yaitu *usability* atau kegunaan aplikasi. *Usability* atau kegunaan aplikasi diukur melalui studi kegunaan aplikasi dengan menggunakan parameter-parameter penilaian. Parameter penilaian yang digunakan yaitu efektivitas, efisiensi dan kepuasan pengguna. Pengukuran melalui studi kegunaan aplikasi ini bertujuan untuk mengetahui tingkat kegunaan aplikasi berdasarkan penggunaan aplikasi oleh pengguna.

4.4 Pemodelan Kebutuhan

Berdasarkan hasil analisis dari kebutuhan-kebutuhan perangkat lunak yang sudah dilakukan sebelumnya, pada bagian pemodelan kebutuhan ini memiliki tujuan untuk memberikan gambaran atau membuat model dari kebutuhan-kebutuhan perangkat lunak yang akan dikembangkan menjadi aplikasi Reado. Pemodelan kebutuhan ini menggunakan pemodelan kebutuhan dari diagram *use case* dan skenario *use case*. Pemodelan kebutuhan dari aplikasi Reado terdiri atas 10 kebutuhan-kebutuhan fungsional perangkat lunak yang sudah ditentukan sebelumnya.

4.4.1 Diagram *Use Case*

Diagram *use case* pada pengembangan aplikasi Reado ini merupakan salah satu dari pemodelan untuk kebutuhan-kebutuhan perangkat lunak dalam pengembangan aplikasi Reado. Diagram *use case* ini memvisualisasikan atau merepresentasikan interaksi aktor dengan sistem. Diagram *use case* ini terdiri dari 10 kebutuhan-kebutuhan fungsional dalam pengembangan aplikasi Reado yang merupakan interaksi antara aktor yaitu anak-anak dan sistem yaitu aplikasi Reado. Diagram digambarkan dengan beberapa simbol yang menunjukkan aktor, *use case* dan sistem dari aplikasi. Aktor digambarkan dengan simbol manusia atau orang, *use case* digambarkan dengan bentuk elips dan sistem digambarkan dengan bentuk kotak.

Adapun diagram *use case* dari kebutuhan-kebutuhan aplikasi Reado dapat dilihat pada Gambar 4.2. Berdasarkan Gambar 4.2, diagram *use case* ini menunjukkan bahwa aktor yaitu anak-anak dapat berinteraksi secara langsung terhadap sistem yang terdiri atas 11 *use case* serta terdapat 1 *use case* yang memiliki relasi *extend* dan 1 *use case* yang memiliki relasi *include* terhadap *use case* yang lainnya. Adapun 11 *use case* yang secara langsung aktor dapat berinteraksi dengan sistem dan 2 *use case* yang memiliki relasi terhadap *use case* lainnya, antara lain sebagai berikut:

1. *Use case* SRS-RD-01 Mendeteksi Kata yaitu memiliki relasi *extend* terhadap *use case* Melihat Kata-Kata dari Hasil Deteksi Kata dan *use case* Melihat Alfabet dari Hasil Deteksi Kata. Relasi ***extend*** ini menunjukkan bahwa saat mendeteksi kata **bisa** melibatkan proses untuk melihat kata-kata dari hasil deteksi kata atau untuk melihat alfabet dari hasil deteksi kata.



2. *Use case* SRS-RD-02 Melihat Kata-Kata dari Hasil Deteksi Kata ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat kata-kata berdasarkan hasil deteksi kata pada aplikasi Reado.
3. *Use case* SRS-RD-03 Melihat Alfabet dari Hasil Deteksi Kata ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat alfabet-alfabet berdasarkan hasil deteksi kata pada aplikasi Reado.
4. *Use case* SRS-RD-04 Memilih Tahapan Permainan ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk memilih tahapan-tahapan permainan yang terdiri dari tahapan latihan dan tahapan permainan tebak kata berdasarkan gambar, suara dan video.
5. *Use case* SRS-RD-05 Memilih *Level* Permainan ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk memilih *level* dari tahapan-tahapan permainan yang terdiri atas tahapan tebak kata berdasarkan gambar, suara dan video.
6. *Use case* SRS-RD-06 Melihat Permainan Tahapan Latihan ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat halaman permainan untuk tahapan latihan mengulang kata. Permainan tahapan latihan ini ditampilkan dengan permainan tebak kata berdasarkan suara untuk mengulang kata-kata yang sudah dideteksi sebelumnya.
7. *Use case* SRS-RD-07 Melihat Permainan Tebak Kata Berdasarkan Gambar ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat halaman permainan tahapan tebak kata berdasarkan gambar yang memvisualisasikan kata tersebut.
8. *Use case* SRS-RD-08 Melihat Permainan Tebak Kata Berdasarkan Suara ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat halaman permainan tahapan tebak kata berdasarkan suara dari cara pembacaan atau pelafalan kata tersebut.
9. *Use case* SRS-RD-09 Melihat Permainan Tebak Kata Berdasarkan Video ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk melihat halaman permainan tahapan tebak kata berdasarkan video yang memvisualisasikan kata tersebut.
10. *Use case* SRS-RD-10 Menjawab Jawaban dari Permainan ini menunjukkan bahwa aktor dapat berinteraksi dengan sistem untuk menjawab jawaban dari permainan yang terdiri atas tahapan latihan dan tahapan tebak kata berdasarkan gambar, suara dan video. Hasil dari *use case* menjawab jawaban dari permainan ini yaitu dapat memilih jawaban dan dapat menentukan jawaban benar atau salah dari jawaban yang dijawab oleh aktor. Selain itu, *use case* ini juga menampilkan hasil jawaban dari permainan berdasarkan tahapan-tahapan permainan. Tahapan-tahapan permainan terdiri atas permainan tahapan latihan, tahapan tebak kata berdasarkan gambar, tahapan tebak kata berdasarkan suara dan tahapan tebak kata berdasarkan video.



Gambar 4.2 Diagram Use Case dari Aplikasi Reado

4.4.2 Skenario Use Case

Skenario *use case* pada pengembangan aplikasi Reado ini merupakan pemodelan yang merepresentasikan interaksi aktor terhadap sistem secara detail yang dijelaskan secara deskriptif. Skenario *use case* ini didefinisikan berdasarkan kebutuhan-kebutuhan fungsional dari pengembangan aplikasi Reado yang terdiri



dari 10 skenario *use case*. Adapun skenario *use case* dari pengembangan aplikasi Reado ini, antara lain sebagai berikut:

1. Skenario *use case* mendeteksi kata dapat dilihat pada Tabel 4.9.

Tabel 4.9 Skenario Use Case Mendeteksi Kata

Kode	SRS-RD-01
Nama Use Case	Mendeteksi kata
Deskripsi	Sistem harus mampu menyediakan mekanisme untuk mendeteksi kata-kata yang dapat ditangkap oleh kamera secara <i>realtime</i> atau waktu sebenarnya.
Aktor	Anak-anak
Prakondisi	Halaman deteksi kata sudah terbuka atau tersedia dan kamera sudah terbuka.
Alur Utama	1. Sistem menampilkan halaman yang terdiri dari tempat untuk menampilkan kata terdeteksi dan tombol untuk menyetujui kata serta kamera sudah menyala. 2. Aktor mengarahkan kamera terhadap kata yang ingin dideteksi dengan benar. 3. Sistem mendeteksi kata dari gambar yang ditangkap. 4. Sistem menampilkan kata-kata yang dideteksi secara <i>realtime</i> pada halaman. 5. Aktor melihat kata-kata yang dapat dideteksi secara <i>realtime</i>. 6. Aktor menekan tombol untuk menyetujui kata yang sudah dideteksi dan memerintahkan sistem untuk memproses. 7. Sistem mengirimkan kata yang terdeteksi untuk ke halaman selanjutnya.
Alur Alternatif	-
Kondisi Akhir	Kata yang dapat dideteksi ditampilkan pada halaman deteksi kata secara <i>realtime</i> atau waktu sebenarnya dan kata yang terdeteksi tersebut sudah terkirim untuk halaman selanjutnya.

2. Skenario *use case* menampilkan kata-kata dari hasil deteksi kata dapat dilihat pada Tabel 4.10.

Tabel 4.10 Skenario *Use Case* Menampilkan Kata-Kata dari Hasil Deteksi Kata

Kode	SRS-RD-02
Nama Use Case	Menampilkan kata-kata dari hasil deteksi kata
Deskripsi	Sistem harus mampu menampilkan kata-kata dari hasil deteksi kata yang sudah dilakukan sebelumnya dan ditampilkan dalam bentuk tulisan serta cara pembacaan kata-kata tersebut dalam bentuk suara.
Aktor	Anak-anak
Prakondisi	Halaman untuk menampilkan kata-kata dari hasil deteksi kata sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan kata-kata dari hasil deteksi kata pada halaman dan menyediakan cara pembacaan kata yaitu berupa tombol suara. 2. Aktor melihat kata-kata dari hasil deteksi kata. 3. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 4. Sistem menampilkan cara pembacaan kata dalam bentuk suara. 5. Aktor mendengarkan cara pembacaan kata.
Alur Alternatif	-
Kondisi Akhir	Kata-kata ditampilkan pada halaman untuk menampilkan kata-kata dari hasil deteksi kata dan menampilkan cara pembacaan kata dalam bentuk suara.

3. Skenario *use case* menampilkan alfabet dari kata berdasarkan hasil deteksi kata dapat dilihat pada Tabel 4.11.

Tabel 4.11 Skenario *Use Case* Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata

Kode	SRS-RD-03
Nama Use Case	Menampilkan alfabet dari kata berdasarkan hasil deteksi kata
Deskripsi	Sistem harus mampu menampilkan alfabet dari kata berdasarkan hasil deteksi kata yang dipilih dan ditampilkan dalam bentuk tulisan serta cara pembacaan alfabet tersebut dalam bentuk suara dan tulisan.
Aktor	Anak-anak



Tabel 4.11 Skenario *Use Case* Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata (lanjutan)

Prakondisi	Halaman untuk menampilkan alfabet dari kata berdasarkan hasil deteksi kata sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan alfabet dari kata pada halaman dan menyediakan cara pembacaan kata dalam bentuk tulisan dan tombol suara. 2. Aktor melihat alfabet dari kata berdasarkan hasil deteksi kata. 3. Aktor melihat cara pembacaan alfabet dalam bentuk tulisan. 4. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 5. Sistem menampilkan cara pembacaan alfabet dalam bentuk suara. 6. Aktor mendengarkan cara pembacaan alfabet.
Alur Alternatif	—
Kondisi Akhir	Alfabet ditampilkan pada halaman untuk menampilkan alfabet dari kata berdasarkan hasil deteksi kata dan cara pembacaan alfabet dalam bentuk tulisan dan suara.

4. Skenario *use case* menentukan tahapan permainan dapat dilihat pada Tabel 4.12.

Tabel 4.12 Skenario *Use Case* Menentukan Tahapan Permainan

Kode	SRS-RD-04
Nama <i>Use Case</i>	Menentukan tahapan permainan
Deskripsi	Sistem harus mampu menyediakan mekanisme untuk menentukan tahapan permainan yang terdiri atas latihan, tebak kata berdasarkan gambar, tebak kata berdasarkan suara dan tebak kata berdasarkan video.
Aktor	Anak-anak
Prakondisi	Halaman untuk menampilkan tahapan-tahapan permainan sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan tahapan-tahapan permainan pada halaman yang terdiri atas latihan, tebak kata berdasarkan gambar, suara dan video.

Tabel 4.12 Skenario *Use Case* Menentukan Tahapan Permainan (lanjutan)

	2. Aktor melihat tahapan-tahapan permainan. 3. Aktor menekan salah satu tahapan permainan dan memerintahkan sistem untuk memproses. 4. Sistem menentukan tahapan permainan yang dipilih aktor.
Alur Alternatif	-
Kondisi Akhir	Tahapan permainan sudah terpilih dan ditentukan.

5. Skenario *use case* menentukan *level* permainan dapat dilihat pada Tabel 4.13.

Tabel 4.13 Skenario *Use Case* Menentukan *Level* Permainan

Kode	SRS-RD-05
Nama Use Case	Menentukan <i>level</i> permainan
Deskripsi	Sistem harus mampu menyediakan mekanisme untuk menentukan <i>level</i> permainan berdasarkan pilihan aktor yang terdiri atas <i>level</i> pada tahapan permainan tebak kata berdasarkan gambar, suara dan video.
Aktor	Anak-anak
Prakondisi	Halaman untuk menampilkan <i>level</i> tahapan permainan sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan <i>level</i> permainan pada halaman yang terdiri atas <i>level</i> tahapan tebak kata berdasarkan gambar, suara dan video. 2. Aktor melihat <i>level</i> permainan. 3. Aktor menekan salah satu <i>level</i> permainan dan memerintahkan sistem untuk memproses. 4. Sistem menentukan <i>level</i> permainan yang dipilih aktor.
Alur Alternatif	-
Kondisi Akhir	<i>Level</i> permainan sudah terpilih dan ditentukan.

6. Skenario *use case* menampilkan permainan tahapan latihan dapat dilihat pada Tabel 4.14.

Tabel 4.14 Skenario *Use Case* Menampilkan Permainan Tahapan Latihan

Kode	SRS-RD-06
Nama Use Case	Menampilkan permainan tahapan latihan



Tabel 4.14 Skenario Use Case Menampilkan Permainan Tahapan Latihan (lanjutan)

Deskripsi	Sistem harus mampu menampilkan permainan tahapan latihan. Tahapan latihan untuk mengulang kata-kata yang sudah dideteksi sebelumnya dalam bentuk permainan tebak kata berdasarkan suara dari pelafalan kata yang harus ditebak.
Aktor	Anak-anak
Prakondisi	Halaman permainan untuk menampilkan tahapan latihan sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan halaman permainan untuk tahapan latihan yang terdiri atas tombol suara. 2. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 3. Sistem menampilkan cara pembacaan kata yang harus ditebak dalam bentuk suara. 4. Aktor mendengarkan cara pembacaan kata.
Alur Alternatif	-
Kondisi Akhir	Suara untuk cara pembacaan kata yang harus ditebak ditampilkan kepada aktor.

7. Skenario *use case* menampilkan permainan tahapan tebak kata berdasarkan gambar dapat dilihat pada Tabel 4.15.

Tabel 4.15 Skenario Use Case Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar

Kode	SRS-RD-07
Nama Use Case	Menampilkan permainan tahapan tebak kata berdasarkan gambar
Deskripsi	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan gambar. Tahapan tebak kata berdasarkan gambar untuk menebak kata dari gambar yang memvisualisasikan kata yang harus ditebak.
Aktor	Anak-anak
Prakondisi	Halaman permainan untuk menampilkan tahapan tebak kata berdasarkan gambar sudah terbuka atau tersedia.



Tabel 4.15 Skenario *Use Case* Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar (lanjutan)

Alur Utama	1. Sistem menampilkan halaman permainan yang terdiri atas gambar yang memvisualisasikan kata yang harus ditebak. 2. Aktor melihat gambar dari kata yang harus ditebak.
Alur Alternatif	-
Kondisi Akhir	Gambar yang memvisualisasikan kata ditampilkan kepada aktor.

8. Skenario *use case* menampilkan permainan tahapan tebak kata berdasarkan suara dapat dilihat pada Tabel 4.16.

Tabel 4.16 Skenario *Use Case* Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara

Kode	SRS-RD-08
Nama <i>Use Case</i>	Menampilkan permainan tahapan tebak kata berdasarkan suara
Deskripsi	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan suara. Tahapan tebak kata berdasarkan suara untuk menebak kata dari suara pelafalan kata yang harus ditebak.
Aktor	Anak-anak
Prakondisi	Halaman permainan untuk menampilkan tahapan tebak kata berdasarkan suara sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan halaman permainan yang terdiri atas tombol suara. 2. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 3. Sistem menampilkan cara pembacaan kata yang harus ditebak dalam bentuk suara. 4. Aktor mendengarkan cara pembacaan kata.
Alur Alternatif	-
Kondisi Akhir	Suara untuk cara pembacaan kata yang harus ditebak ditampilkan kepada aktor.

9. Skenario *use case* menampilkan permainan tahapan tebak kata berdasarkan video dapat dilihat pada Tabel 4.17.

Tabel 4.17 Skenario Use Case Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video

Kode	SRS-RD-09
Nama Use Case	Menampilkan permainan tahapan tebak kata berdasarkan video
Deskripsi	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan video. Tahapan tebak kata berdasarkan video untuk menebak kata dari video yang memvisualisasikan kata yang harus ditebak.
Aktor	Anak-anak
Prakondisi	Halaman permainan untuk menampilkan tahapan tebak kata berdasarkan video sudah terbuka atau tersedia.
Alur Utama	1. Sistem menampilkan halaman permainan yang terdiri atas video yang memvisualisasikan kata yang harus ditebak. 2. Aktor melihat video dari kata yang harus ditebak.
Alur Alternatif	1.
Kondisi Akhir	Video yang memvisualisasikan kata ditampilkan kepada aktor.

10. Skenario use case menyediakan mekanisme untuk jawaban dari permainan dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario Use Case Menyediakan Mekanisme untuk Jawaban dari Permainan

Kode	SRS-RD-10
Nama Use Case	Menyediakan mekanisme untuk jawaban dari permainan
Deskripsi	Sistem harus mampu menyediakan mekanisme untuk jawaban dan menentukan jawaban yang benar atau salah dari permainan disetiap tahapan. Jawaban yang harus dipilih terdiri atas alfabet-alfabet dari kata yang harus ditebak.
Aktor	Anak-anak
Prakondisi	Halaman permainan untuk menampilkan tahapan latihan atau tebak kata berdasarkan gambar, suara dan video sudah terbuka atau tersedia.



Tabel 4.18 Skenario Use Case Menyediakan Mekanisme untuk Jawaban dari Permainan (lanjutan)

Alur Utama	1. Sistem menampilkan tombol-tombol pilihan alfabet untuk menjawab, tombol untuk mengatur ulang dan tempat untuk menampilkan jawaban yang sudah dipilih oleh aktor. 2. Aktor menekan tombol pilihan jawaban alfabet dan memerintahkan sistem untuk memproses. 3. Sistem menampilkan jawaban yang sudah dipilih oleh aktor. 4. Aktor menekan tombol mengatur ulang dan memerintahkan sistem untuk memproses. 5. Sistem menampilkan jawaban aktor setelah diatur ulang. 6. Aktor melihat jawaban yang sudah dipilih. 7. Sistem melakukan validasi terhadap jawaban dari permainan yang sudah dipilih. 8. Sistem menampilkan hasil jawaban dari permainan dan tombol untuk menuju level selanjutnya. 9. Aktor melihat hasil jawaban dari permainan yaitu "Kamu Benar!" dan tombol "Berikutnya". 10. Aktor menekan tombol dan memerintahkan sistem untuk memproses. 11. Sistem menampilkan halaman tahapan permainan kembali.
Alur Alternatif	Kegagalan validasi jawaban: Jika jawaban tidak benar, maka sistem akan menampilkan hasil jawaban dari permainan yaitu "Oops, Kamu Salah!" dan tombol untuk mengulang permainan yaitu tombol "Coba Kembali".
Kondisi Akhir	Tahapan permainan kembali terbuka atau tersedia.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

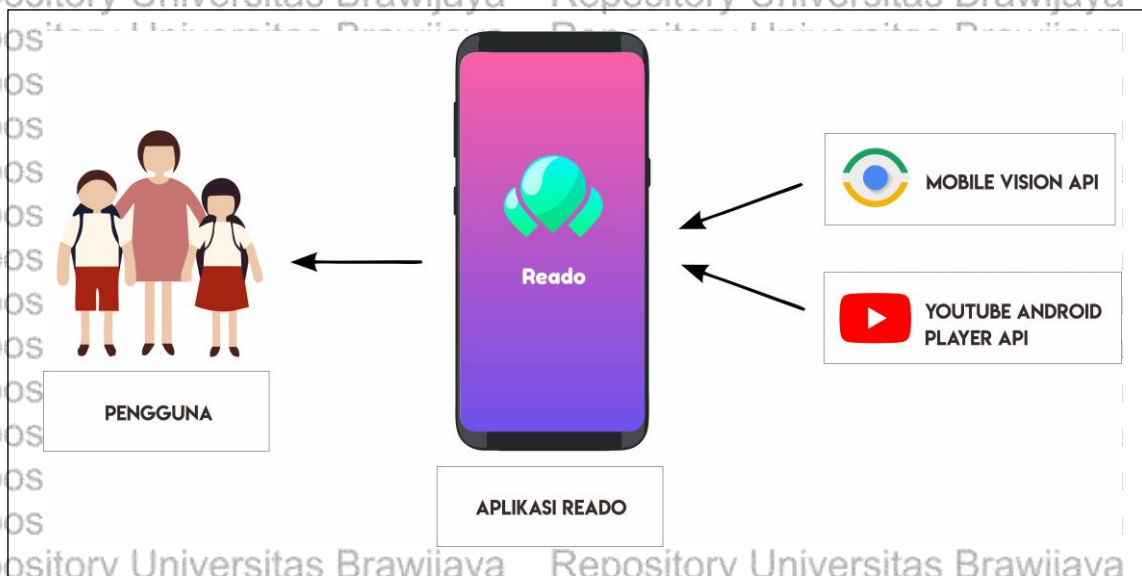
Pada bagian perancangan dan implementasi di dalam penelitian skripsi ini akan dijelaskan terkait dengan tahapan-tahapan dari perancangan dan implementasi. Tahapan perancangan dan implementasi ini menggunakan metode *Design Sprint* yang dilakukan pada hari kedua sampai hari kelima yaitu tahapan *ideate*, *decide*, *prototype* dan *test*. Tahapan-tahapan ini dilakukan untuk mendapatkan prototipe yang tervalidasi dan selanjutnya akan diimplementasikan menjadi aplikasi yang dapat digunakan oleh pengguna.

5.1 Perancangan Perangkat Lunak

Dari perancangan perangkat lunak di dalam pengembangan aplikasi Reado dijelaskan beberapa perancangan yang dilakukan. Perancangan-perancangan terdiri atas perancangan arsitektur aplikasi, perancangan diagram, perancangan basis data, perancangan algoritma dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur Aplikasi

Perancangan arsitektur aplikasi dibuat dengan tujuan untuk memberikan gambaran terkait dengan teknologi yang digunakan dalam pengembangan aplikasi Reado. Aplikasi Reado dikembangkan dengan menggunakan dua teknologi yaitu fitur pada Mobile Vision API dan YouTube Android Player API. Fitur yang digunakan dari Mobile Vision API yaitu *recognize text* untuk mendeteksi kata dari gambar yang ditangkap oleh kamera. Selain itu, pengembangan aplikasi Reado juga menggunakan YouTube Android Player API sebagai penyedia fungsi-fungsi untuk pemutaran video di dalam aplikasi. Adapun skema dari perancangan arsitektur aplikasi Reado dengan menggunakan Mobile Vision API dan Youtube Android Player API dapat dilihat pada Gambar 5.1.



Gambar 5.1 Skema Perancangan Arsitektur Aplikasi Reado



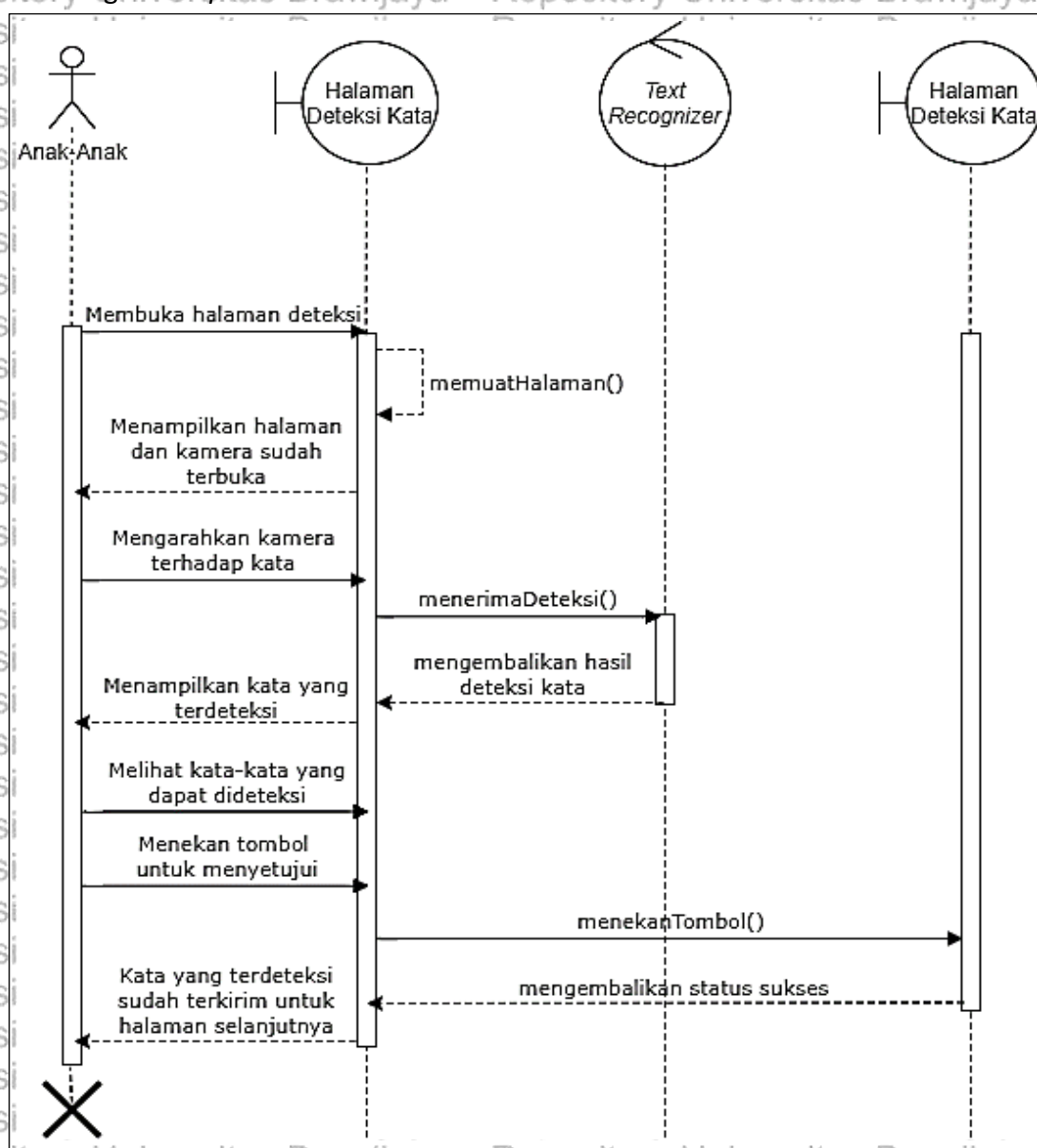
5.1.2 Perancangan Diagram

Perancangan diagram dari pengembangan aplikasi Reado terdiri dari beberapa diagram. Adapun diagram yang digunakan yaitu terdiri atas diagram *sequence* dan diagram *class*.

5.1.2.1 Diagram Sequence

Diagram *sequence* merupakan pemodelan untuk menggambarkan interaksi antar pengguna dan sistem. Dari setiap kebutuhan fungsional yang sudah didefinisikan sebelumnya yang terdiri dari 10 kebutuhan fungsional memiliki masing-masing diagram *sequence*. Adapun diagram *sequence* dari setiap kebutuhan fungsional dari pengembangan aplikasi Reado, antara lain yaitu:

1. Diagram *sequence* mendeteksi kata



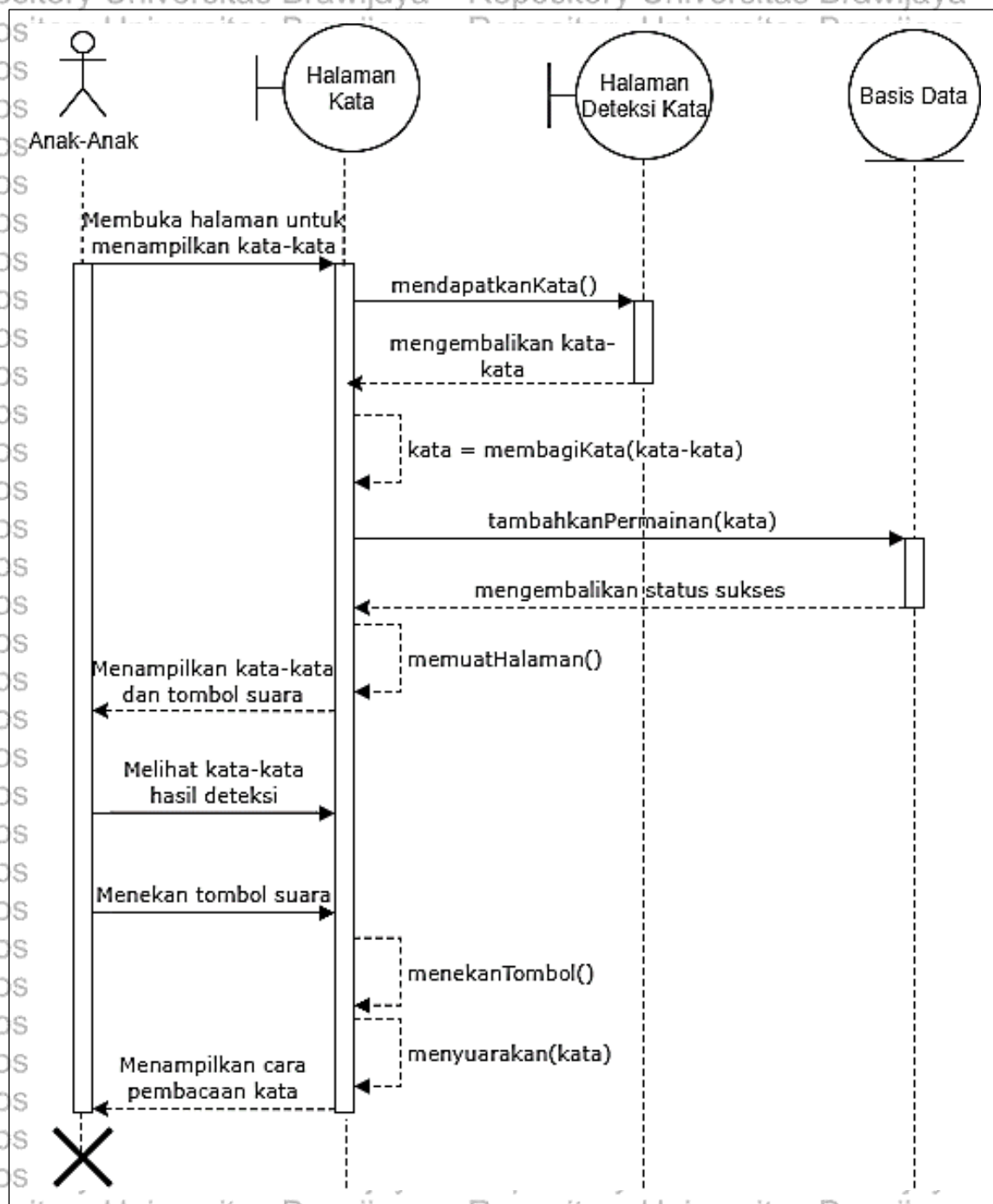
Gambar 5.2 Diagram Sequence Mendeteksi Kata



Berdasarkan Gambar 5.2, diagram *sequence* mendeteksi kata ini menunjukkan interaksi yang diawali dengan aktor membuka halaman deteksi kata. Setelah aktor membuka halaman deteksi kata, lalu halaman deteksi kata akan memuat halaman dengan menampilkan halaman dan menyalakan kamera untuk dapat menangkap gambar dari kata-kata yang ada di sekitar aktor. Interaksi yang kedua yaitu aktor mengarahkan kamera terhadap kata-kata yang ingin dideteksi dan kata-kata tersebut dapat ditemukan di sekitar aktor, buku, poster ataupun majalah. Setelah aktor mengarahkan kamera terhadap kata-kata yang ingin dideteksi, halaman deteksi kata akan memanggil fungsi untuk menerima deteksi kata kepada *Text Recognizer* dari *Mobile Vision API*. Selanjutnya fungsi tersebut akan mengembalikan hasil kata yang sudah dideteksi kepada halaman deteksi kata dan kata tersebut akan ditampilkan pada halaman deteksi kata. Interaksi selanjutnya yaitu aktor melihat kata-kata yang terdeteksi oleh kamera, lalu aktor menekan tombol untuk menyetujui bahwa kata yang terdeteksi sudah benar. Selanjutnya sistem akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan yaitu dengan mengirimkan kata yang terdeteksi tersebut untuk halaman selanjutnya yaitu halaman kata.

2. Diagram *sequence* menampilkan kata-kata dari hasil deteksi kata

Diagram *sequence* menampilkan kata-kata dari hasil deteksi kata ditunjukkan pada Gambar 5.3. Diagram *sequence* ini menunjukkan Interaksi yang diawali dengan aktor membuka halaman pada aplikasi. Setelah aktor membuka halaman untuk menampilkan kata-kata, lalu halaman kata akan memanggil fungsi untuk mendapatkan kata dari halaman deteksi kata dan fungsi ini akan mengembalikan kata-kata yang didapatkan pada halaman deteksi kata. Tahapan selanjutnya yaitu membagi kata-kata yang didapatkan pada halaman sebelumnya dengan menginisialisasikan suatu variabel yaitu variabel kata untuk menyimpan nilai kata-kata yang sudah dibagi atau dipisahkan menjadi kata. Lalu, pada halaman kata juga akan memanggil fungsi untuk menambahkan kata-kata yang didapatkan dari hasil deteksi kata ke dalam basis data dengan menggunakan variabel kata sebagai parameter dari fungsi tersebut dan fungsi ini akan mengembalikan nilai yaitu berupa status sukses. Selanjutnya halaman kata memanggil fungsi untuk menampilkan halaman kata dan tahapan terakhir yaitu akan menampilkan kata-kata secara terpisah dan tombol suara. Selanjutnya, untuk interaksi yang kedua yaitu aktor melihat kata-kata hasil deteksi kata secara terpisah-pisah atau terbagi-bagi dalam sebuah *list* atau daftar. Lalu, interaksi yang ketiga yaitu aktor menekan tombol suara untuk mendengarkan cara pembacaan atau cara pelafalan dari setiap kata hasil deteksi yang sudah ditampilkan sebelumnya. Setelah menekan tombol suara, halaman kata akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan dan juga memanggil fungsi untuk menyuarkan kata yang dipilih oleh aktor. Selanjutnya fungsi tersebut akan membuat halaman kata dapat menampilkan cara pembacaan atau cara pelafalan dari kata hasil deteksi yang dipilih oleh aktor dalam bentuk suara. Interaksi terakhir yang dilakukan oleh aktor yaitu aktor akan mendengarkan cara pembacaan atau cara pelafalan dari kata yang dipilih oleh aktor berdasarkan hasil deteksi kata yang dilakukan pada aplikasi *Reado*.



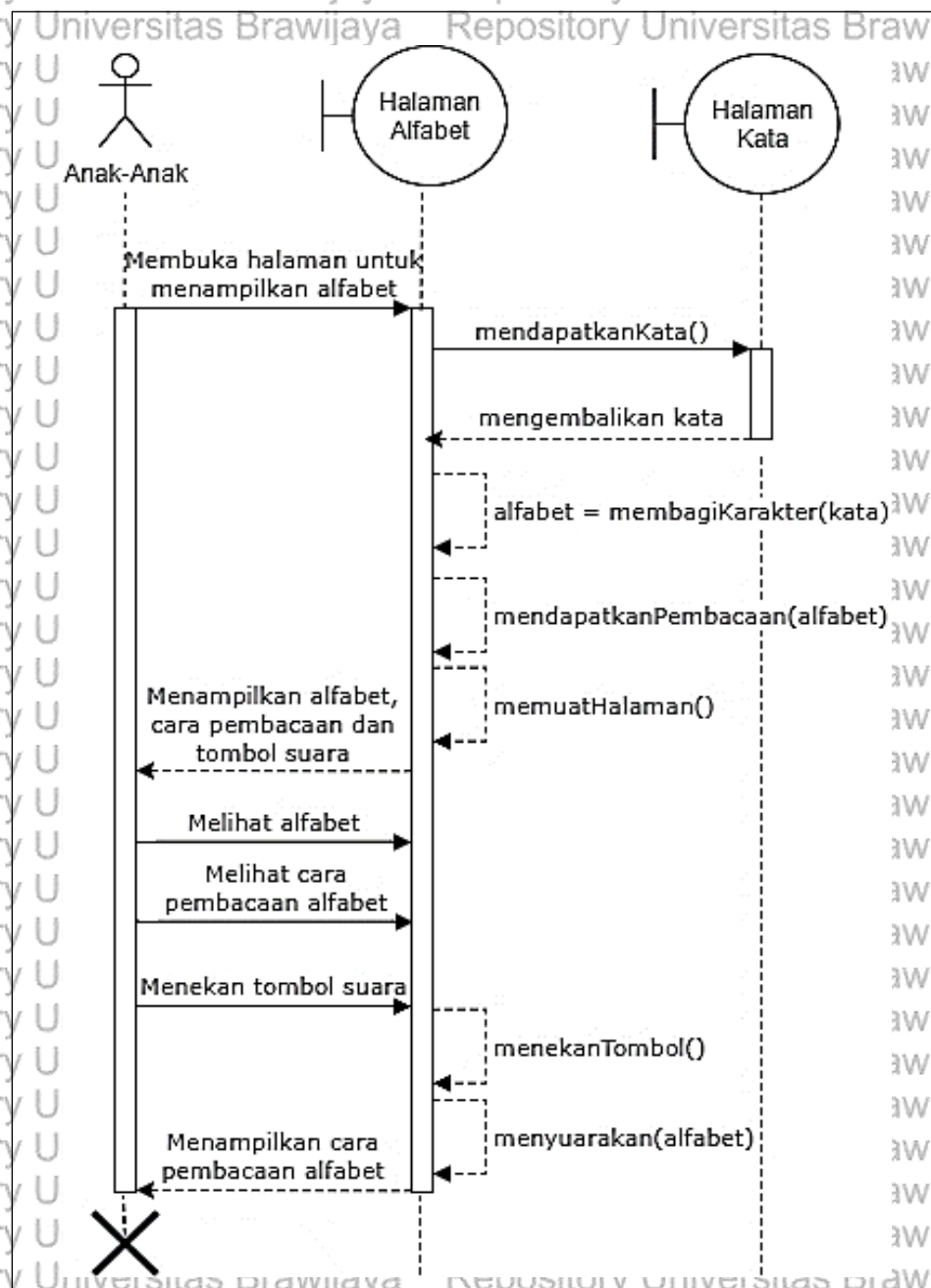
Gambar 5.3 Diagram Sequence Menampilkan Kata-Kata dari Hasil Deteksi Kata

3. Diagram *sequence* menampilkan alfabet dari kata berdasarkan hasil deteksi kata

Diagram *sequence* menampilkan alfabet dari kata berdasarkan hasil deteksi kata ditunjukkan pada Gambar 5.4. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan alfabet. Setelah membuka halaman, lalu halaman alfabet akan memanggil fungsi untuk mendapatkan kata pada halaman kata. Lalu halaman deteksi akan menyimpan nilai dari karakter-karakter kata yang dipilih oleh aktor pada variabel alfabet dan



selanjutnya akan memanggil fungsi untuk mendapatkan cara pembacaan dari setiap alfabet dengan menggunakan variabel alfabet sebagai parameternya. Tahapan selanjutnya yaitu halaman akan memanggil fungsi untuk menampilkan halaman yang terdiri atas tulisan alfabet, tulisan cara pembacaan dan tombol suara. Interaksi yang kedua dan ketiga yaitu melihat alfabet dan melihat cara pembacaan alfabet tersebut. Interaksi terakhir yaitu aktor akan menekan tombol suara, lalu halaman alfabet akan memanggil fungsi untuk menyuarkan cara pembacaan atau pelafalan dari alfabet dan halaman akan menampilkan cara pembacaan alfabet tersebut dalam bentuk suara.

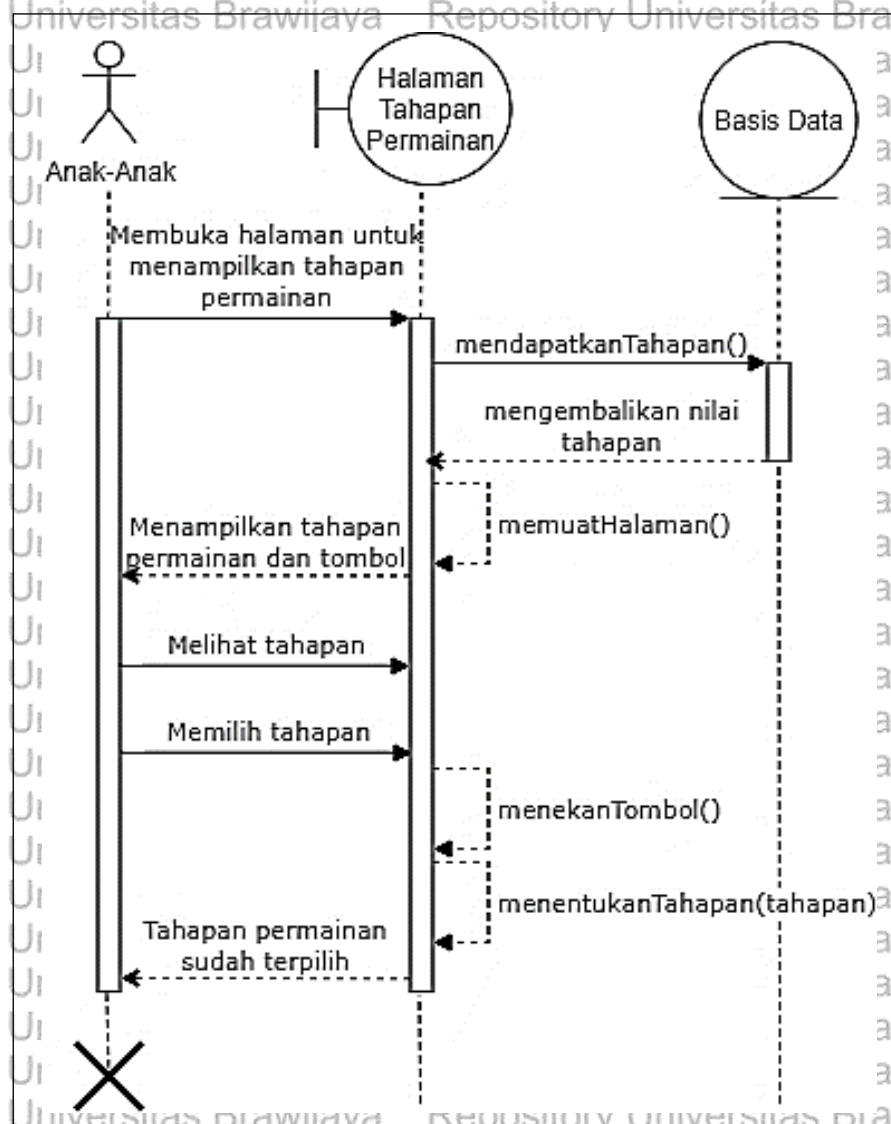


Gambar 5.4 Diagram Sequence Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata



4. Diagram *sequence* menentukan tahapan permainan

Diagram *sequence* menentukan tahapan permainan ditunjukkan pada Gambar 5.5. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan tahapan permainan. Setelah membuka halaman, lalu halaman tahapan permainan akan memanggil fungsi untuk mendapatkan nilai keterangan dari tahapan permainan pada basis data. Selanjutnya halaman akan memanggil fungsi untuk memuat halaman sehingga halaman dapat menampilkan tahapan-tahapan permainan dan tombol untuk menuju tahapan permainan tersebut. Interaksi yang kedua yaitu aktor melihat tahapan-tahapan dari permainan yang ditampilkan. Interaksi terakhir yaitu aktor memilih salah satu tahapan, lalu halaman tahapan permainan akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan dan fungsi untuk menentukan tahapan yang dipilih oleh aktor dan selanjutnya tahapan permainan akan terpilih atau sudah ditentukan.

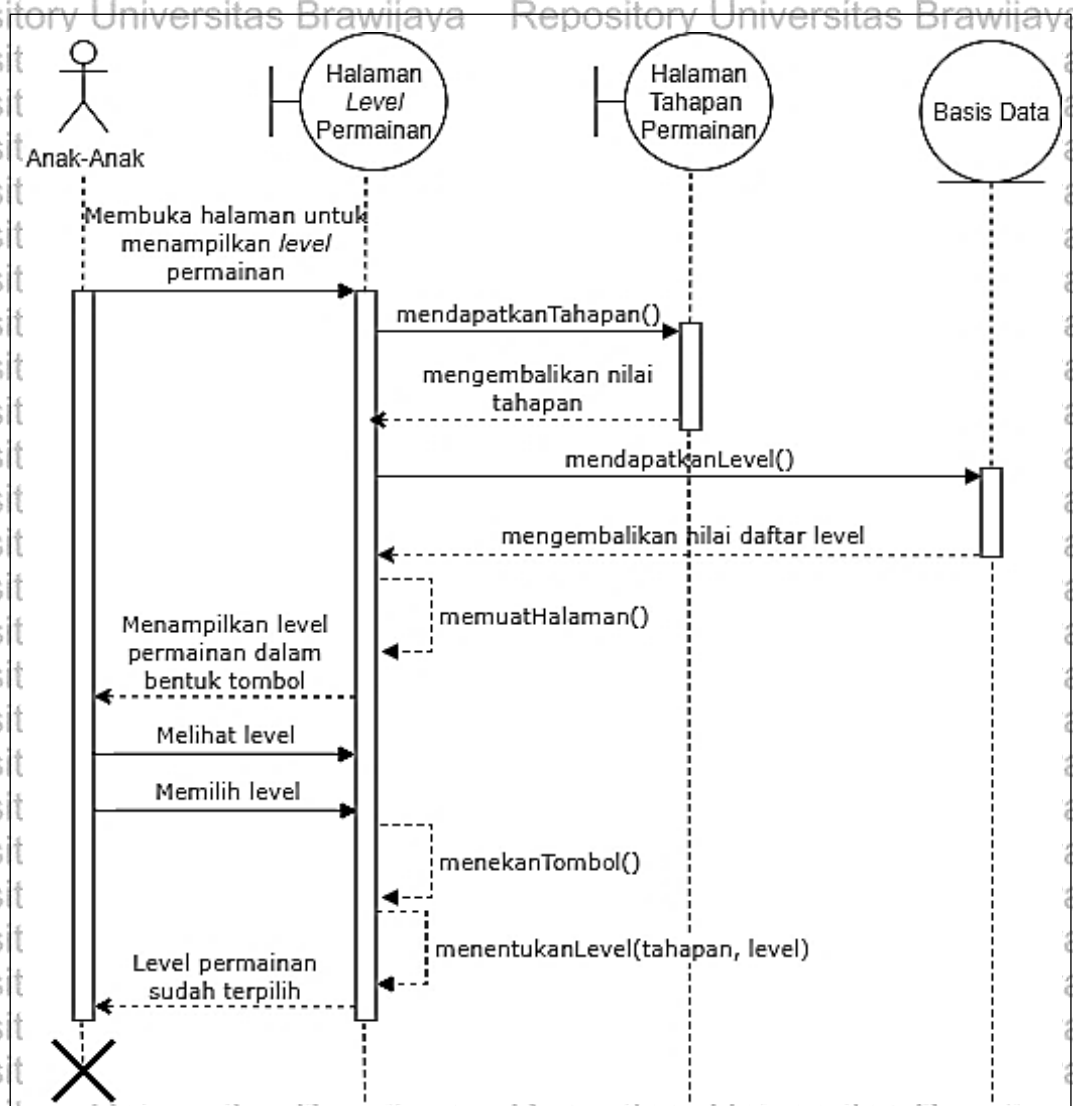


Gambar 5.5 Diagram *Sequence* Menentukan Tahapan Permainan



5. Diagram *sequence* menentukan *level* permainan

Diagram *sequence* menentukan *level* permainan ditunjukkan pada Gambar 5.6. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan *level* permainan. Setelah membuka halaman, lalu halaman tahapan permainan akan memanggil fungsi untuk mendapatkan nilai tahapan dari Halaman Tahapan Permainan dan akan memanggil fungsi untuk mendapatkan nilai *level* yang didapatkan dari basis data. Selanjutnya Halaman Level Permainan akan memanggil fungsi untuk memuat halaman sehingga halaman dapat menampilkan *level* permainan dalam bentuk tombol-tombol. Interaksi yang kedua yaitu aktor melihat *level* dari permainan yang ditampilkan. Interaksi terakhir yaitu aktor memilih salah satu *level*, lalu halaman *level* permainan akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan dan fungsi untuk menentukan *level* yang dipilih oleh aktor sehingga *level* permainan akan terpilih atau sudah ditentukan.

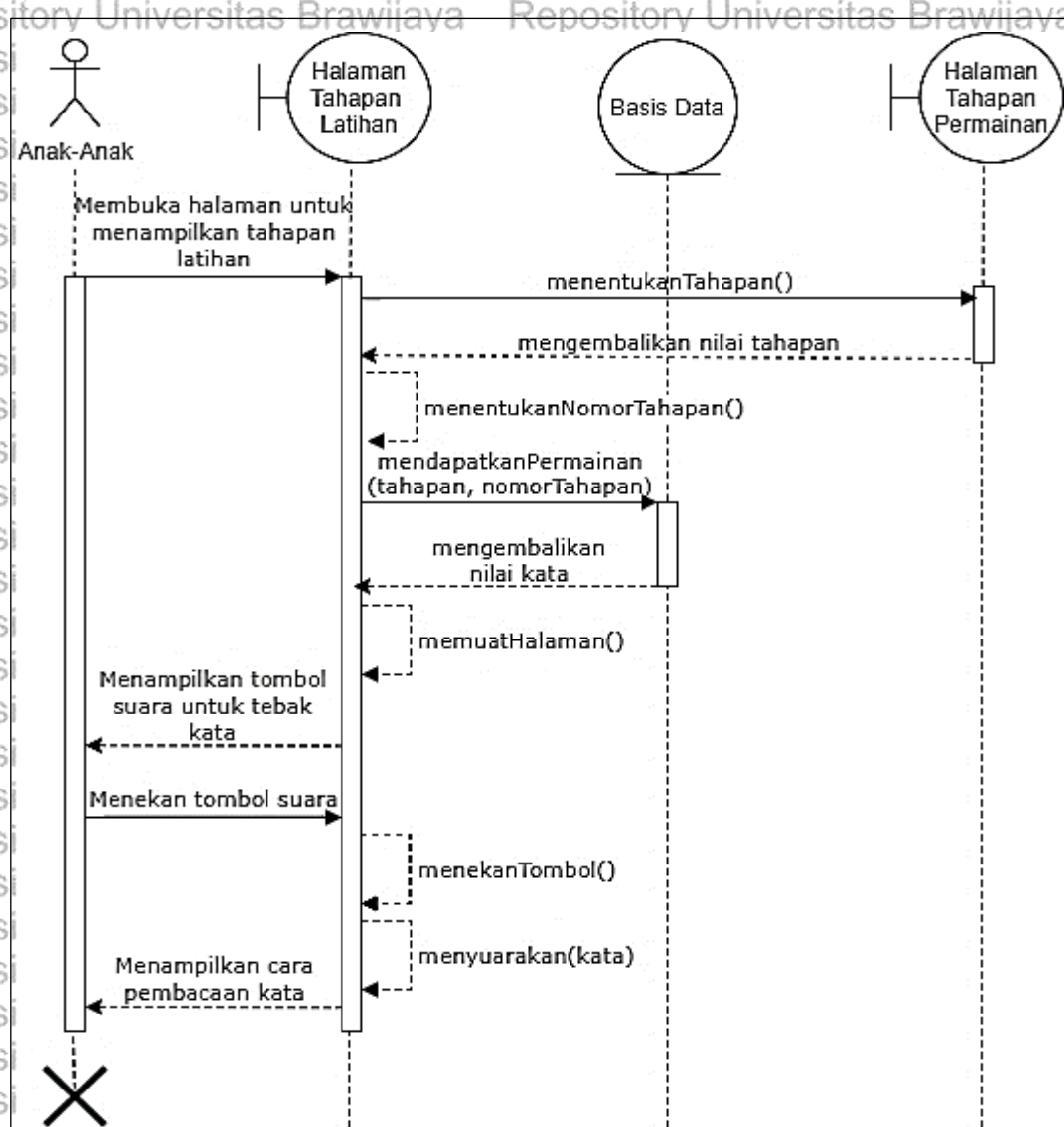


Gambar 5.6 Diagram *Sequence* Menentukan *Level* Permainan



6. Diagram *sequence* menampilkan permainan tahapan latihan

Diagram *sequence* menampilkan permainan tahapan latihan dapat dilihat pada Gambar 5.7. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan tahapan latihan. Setelah membuka halaman, lalu halaman akan memanggil fungsi untuk menentukan tahapan dari halaman sebelumnya dan fungsi untuk menentukan nomor tahapan. Lalu halaman juga akan memanggil fungsi untuk mendapatkan kata dari basis data. Selanjutnya halaman akan memanggil fungsi untuk memuat halaman sehingga dapat menampilkan tombol suara untuk tahapan latihan. Interaksi yang kedua yaitu aktor menekan tombol suara untuk mendengarkan kata yang harus ditebak. Lalu, halaman akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan dan untuk menyuarakan kata yang harus ditebak. Tahapan terakhir yaitu halaman akan menampilkan cara pembacaan atau cara pelafalan dari kata yang harus ditebak melalui suara.

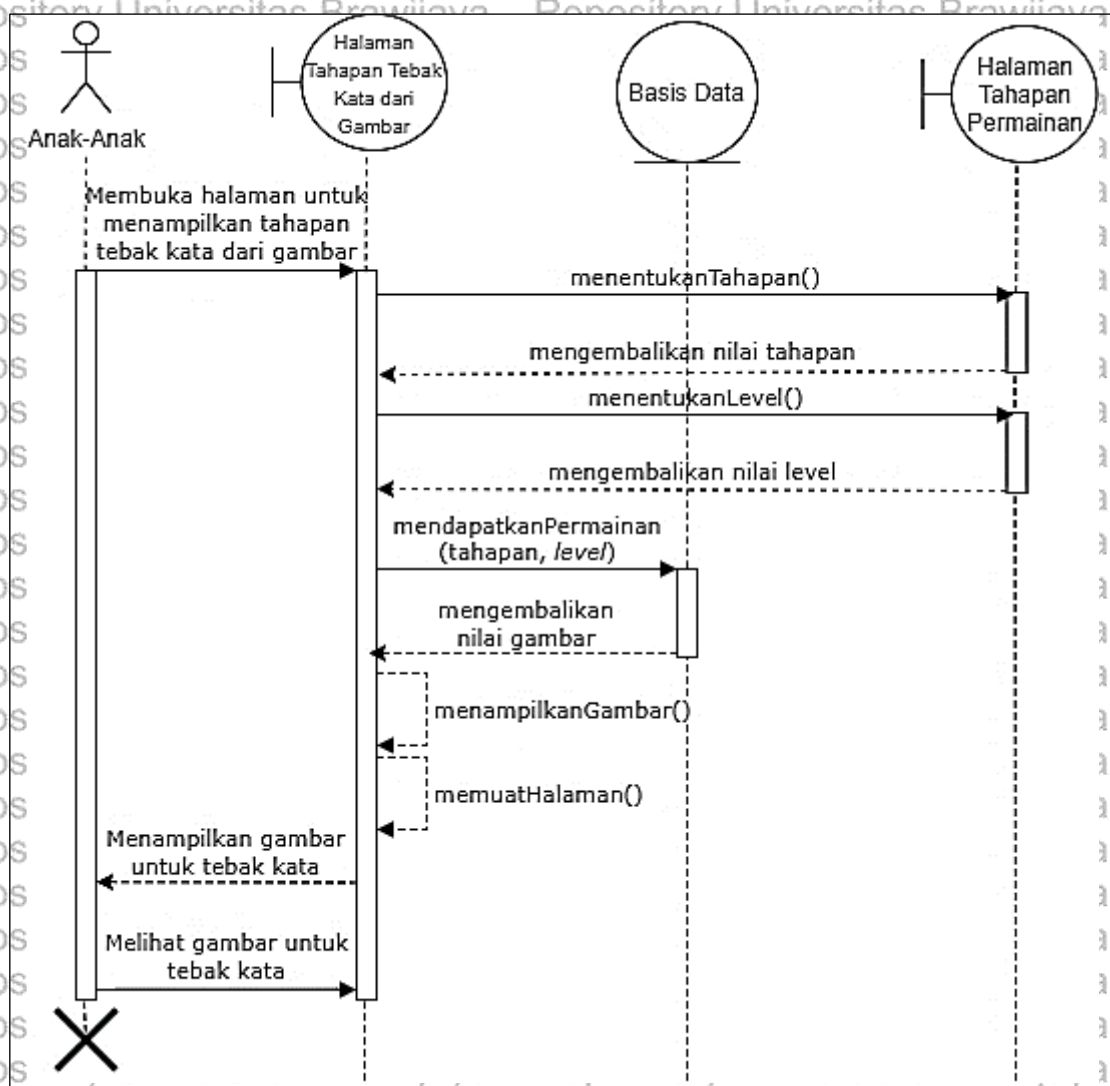


Gambar 5.7 Diagram *Sequence* Menampilkan Permainan Tahapan Latihan



7. Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan gambar

Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan gambar dapat dilihat pada Gambar 5.8. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan tahapan tebak kata berdasarkan gambar. Setelah membuka halaman, lalu halaman akan memanggil fungsi untuk menentukan tahapan dan menentukan level pada halaman sebelumnya. Selanjutnya halaman akan memanggil fungsi untuk mendapatkan gambar yang harus ditebak yang didapatkan dari basis data. Lalu halaman tahapan tebak kata berdasarkan gambar akan menampilkan gambar dari kata yang harus ditebak dan memanggil fungsi untuk memuat halaman. Interaksi yang kedua yaitu aktor akan melihat gambar yang ditampilkan pada aplikasi ReadO untuk dapat ditebak kata dalam Bahasa Inggris dari gambar tersebut pada permainan tahapan tebak kata berdasarkan gambar.

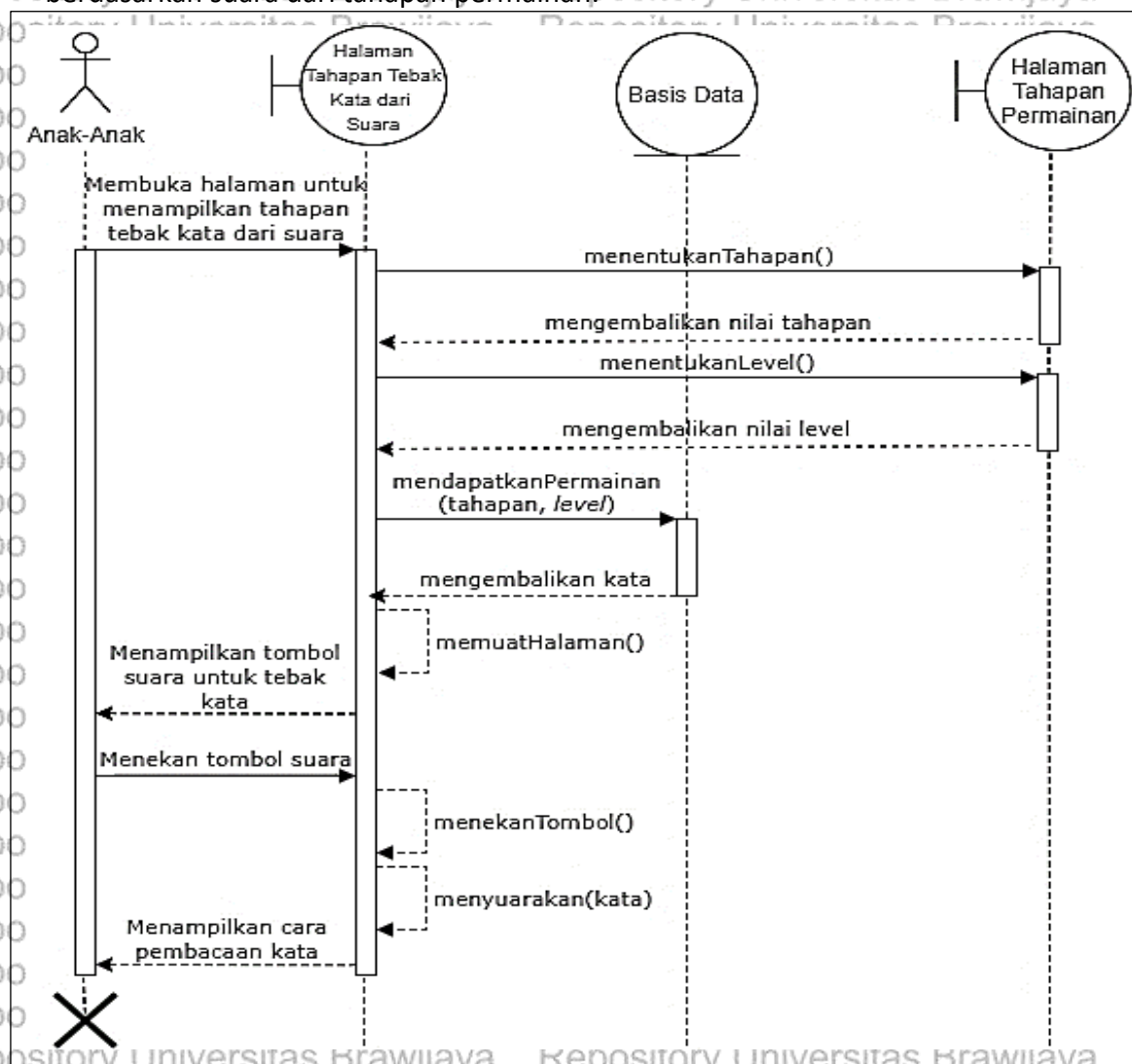


Gambar 5.8 Diagram Sequence Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar



8. Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan suara

Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan suara dapat dilihat pada Gambar 5.9. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan tahapan tebak kata dari suara. Setelah membuka halaman, lalu halaman akan memanggil fungsi untuk menentukan tahapan dan menentukan *level* yang didapatkan dari halaman sebelumnya. Selanjutnya halaman akan memanggil fungsi untuk mendapatkan kata dari permainan tebak kata berdasarkan suara. Lalu halaman memuat halaman sehingga dapat menampilkan tombol suara untuk mendengarkan kata yang harus ditebak. Interaksi yang kedua yaitu aktor menekan tombol suara untuk mendengarkan kata. Lalu, halaman akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan dan fungsi untuk menyuarkan kata. Selanjutnya halaman akan menampilkan cara pembacaan dari kata yang harus ditebak berdasarkan suara dari tahapan permainan.



Gambar 5.9 Diagram Sequence Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara

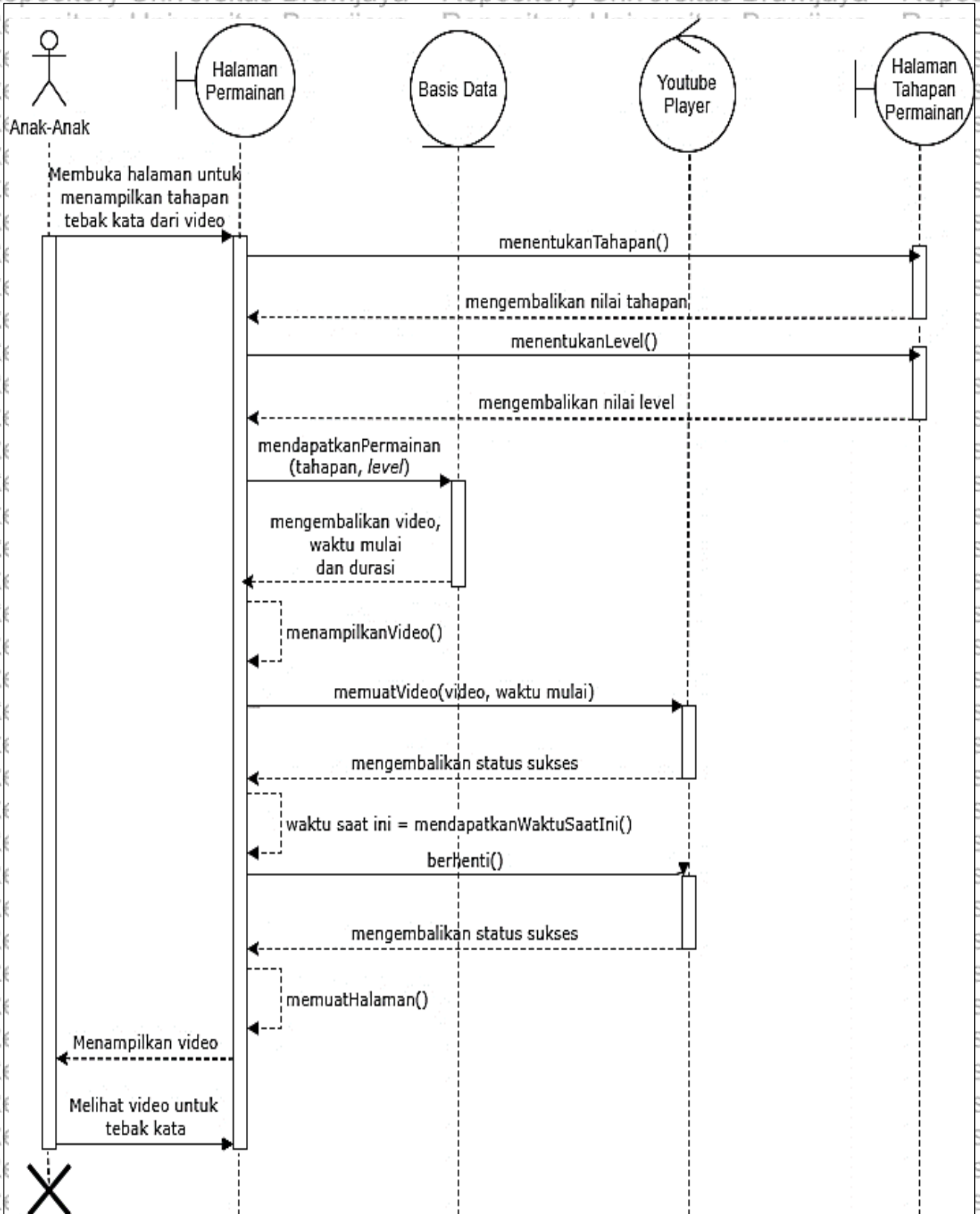


9. Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan video

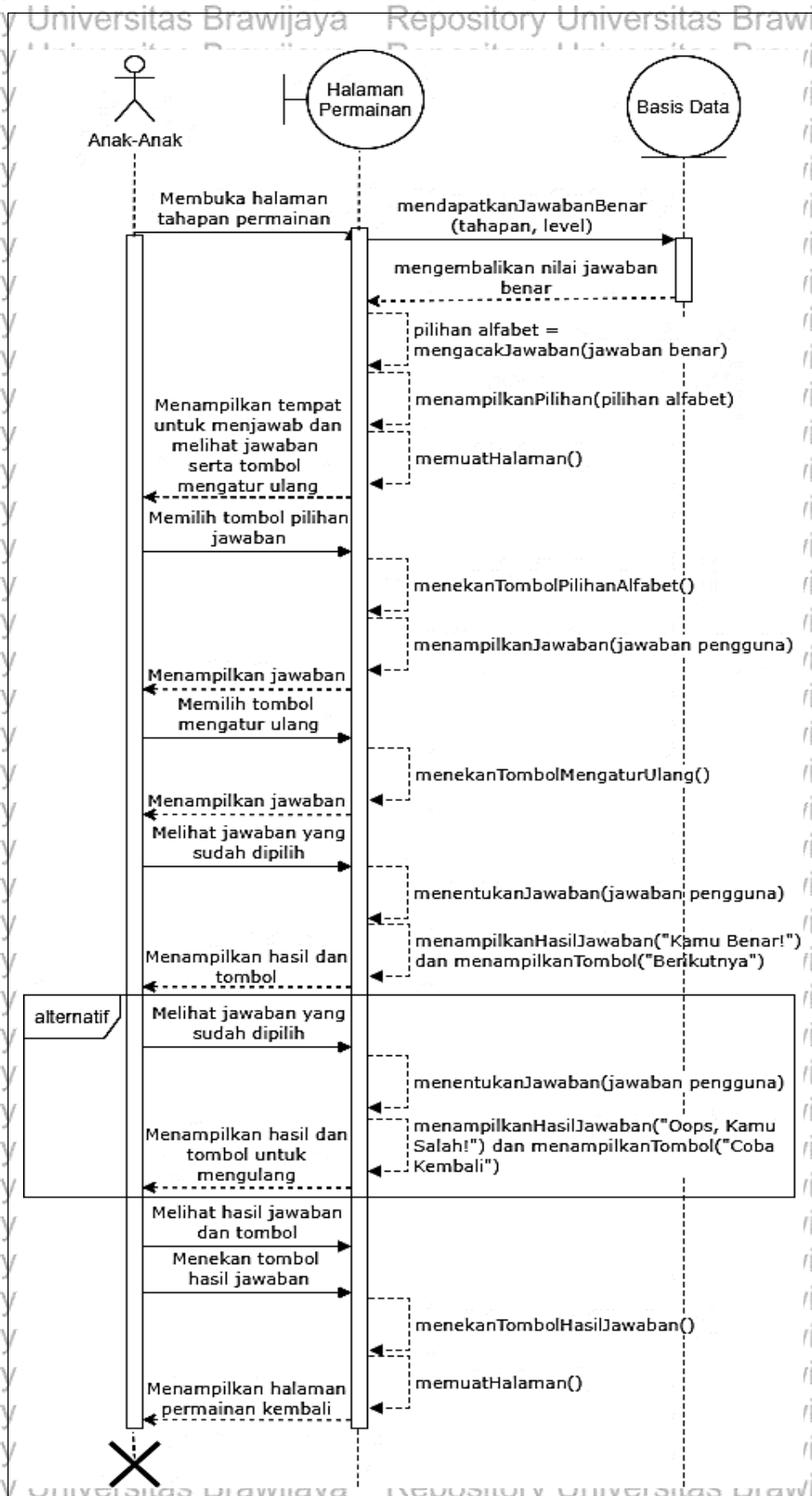
Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan video dapat dilihat pada Gambar 5.10. Diagram *sequence* menampilkan permainan tahapan tebak kata berdasarkan video ini menunjukkan interaksi yang diawali dengan aktor membuka halaman untuk menampilkan tahapan tebak kata berdasarkan video. Setelah membuka halaman, lalu halaman akan memanggil fungsi untuk menentukan tahapan yang didapatkan dari halaman sebelumnya dan fungsi untuk menentukan level yang didapatkan dari halaman sebelumnya. Lalu halaman akan memanggil fungsi untuk mendapatkan permainan yaitu berupa nilai dari video, waktu mulai dan durasi yang didapatkan dari basis data. Selanjutnya halaman akan memanggil fungsi untuk menampilkan video. Tahapan selanjutnya yaitu memanggil fungsi untuk memuat video pada *Youtube Player*. *Youtube Player* merupakan fitur dari YouTube Android Player API untuk menampilkan video Youtube ke dalam aplikasi Reado, sehingga dengan menggunakan fitur ini maka halaman tahapan tebak kata dari video bisa untuk menampilkan video. Selanjutnya halaman akan memanggil fungsi untuk mendapatkan nilai waktu saat ini. Lalu, halamannya akan memanggil fungsi untuk memberhentikan video pada *Youtube Player* dan fungsi untuk memuat halaman. Interaksi selanjutnya yaitu aktor akan melihat video yang memvisualisasikan gambar untuk menebak kata dalam tahapan permainan tebak kata berdasarkan video melalui aplikasi Reado.

10. Diagram *sequence* menyediakan mekanisme untuk jawaban dari permainan

Diagram *sequence* menyediakan mekanisme untuk jawaban dari permainan dapat dilihat pada Gambar 5.11. Diagram *sequence* ini menunjukkan interaksi yang diawali dengan aktor membuka halaman, lalu halaman akan memanggil fungsi untuk mendapatkan jawaban yang benar dari basis data. Selanjutnya halaman akan memanggil fungsi untuk mengacak pilihan jawaban menjadi alfabet berdasarkan jawaban yang benar, lalu memanggil fungsi untuk menampilkan pilihan jawaban dan fungsi untuk memuat halaman permainan sehingga dapat menampilkan tempat untuk menjawab dan melihat jawaban serta tombol untuk mengatur ulang pada halaman. Interaksi selanjutnya yaitu halaman akan memanggil fungsi untuk memberikan perintah ketika tombol pilihan alfabet ditekan dan fungsi untuk menampilkan jawaban yang sudah dijawab oleh pengguna. Selanjutnya, aktor akan memilih tombol untuk mengatur ulang jawaban, sehingga halaman akan memanggil fungsi untuk memberikan perintah ketika tombol untuk mengatur ulang ditekan. Interaksi selanjutnya yaitu aktor melihat jawaban yang sudah dipilih. Secara bersamaan halaman akan memanggil fungsi untuk menentukan jawaban yang sudah dijawab oleh pengguna. Jika jawaban benar, maka akan ditampilkan hasil jawaban yaitu "Kamu Benar!" dan menampilkan tombol "Berikutnya". Namun jika jawaban salah, maka akan ditampilkan hasil jawaban yaitu "Oops, Kamu Salah!" dan ditampilkan tombol yaitu "Coba Kembali" untuk mengulang kembali tahapan permainan. Interaksi selanjutnya yaitu aktor akan melihat hasil jawaban dan tombol yang ditampilkan dan aktor akan menekan tombol dari hasil jawaban tersebut dan halaman akan menampilkan halaman permainan kembali.



Gambar 5.10 Diagram Sequence Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video



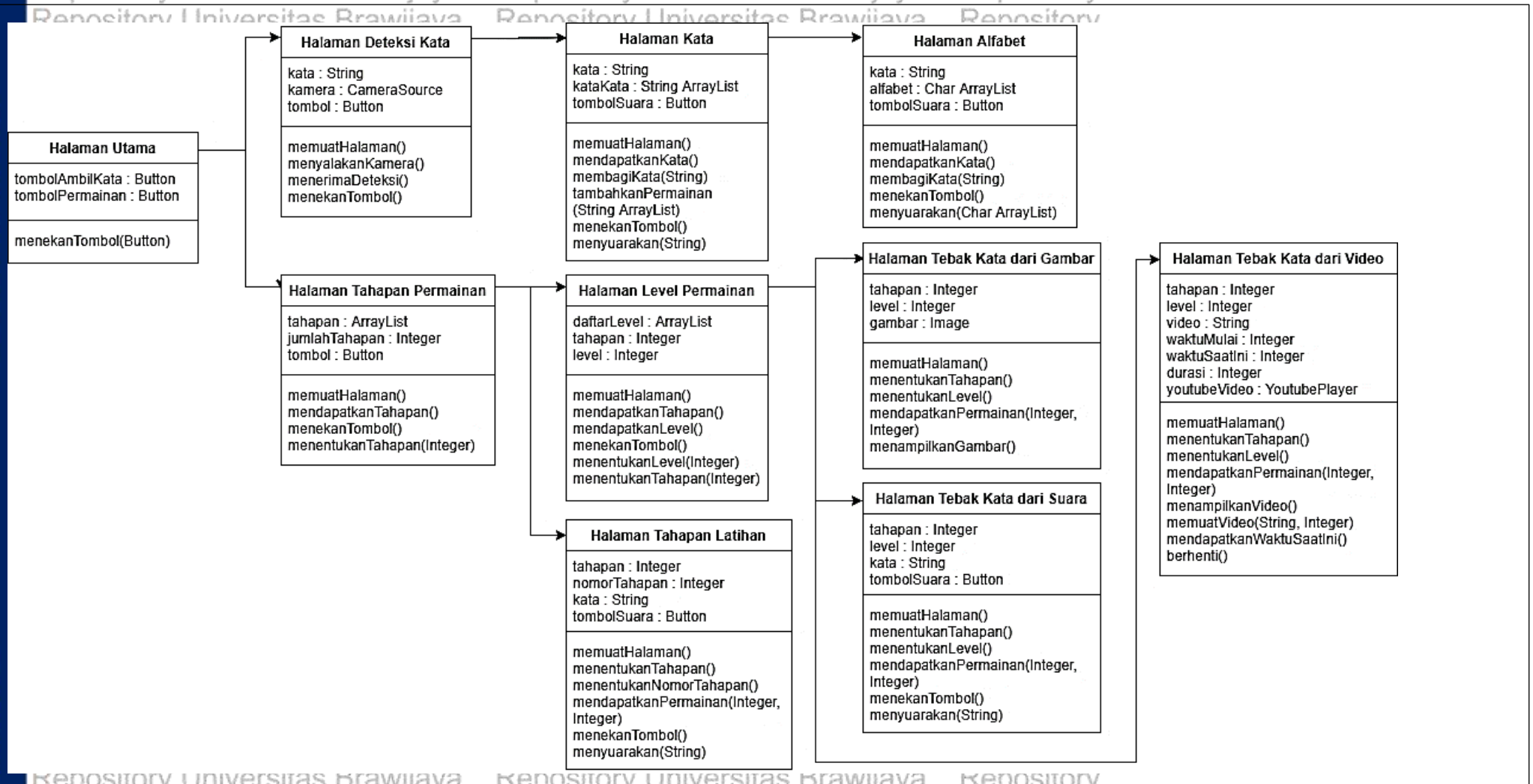
Gambar 5.11 Diagram Sequence Menyediakan Mekanisme untuk Jawaban dari Permainan



5.1.2.2 Diagram *Class*

Diagram *class* merupakan sebuah diagram yang merepresentasikan model dari hubungan antar kelas-kelas, mengatur alur logika ataupun gambaran dari fungsi-fungsi yang dibutuhkan untuk implementasi dari aplikasi ReadO. Adapun perancangan diagram *class* pada pengembangan aplikasi ReadO dapat dilihat pada Gambar 5.12.

Berdasarkan Gambar 5.12, terdapat 9 kelas yang saling memiliki hubungan atau relasi. Kelas pertama yaitu Halaman Utama yang memiliki relasi terhadap 2 kelas yaitu Halaman Deteksi Kata dan Halaman Tahapan Permainan. Halaman utama memiliki atribut yaitu tombol ambil kata dan tombol permainan serta memiliki fungsi yaitu menekan tombol untuk menuju kepada halaman yang diinginkan. Kelas kedua yaitu Halaman Deteksi Kata yang memiliki relasi dengan Halaman Kata. Halaman Deteksi Kata memiliki atribut yaitu variabel kata, kamera dan tombol serta fungsi untuk memuat halaman, menyalakan kamera, menerima deteksi kata dan fungsi untuk memberikan perintah ketika menekan tombol. Kelas Ketiga yaitu Halaman Kata yang memiliki relasi dengan Halaman Alfabet. Halaman Kata memiliki atribut yaitu variabel kata, kata-kata dan tombol suara serta fungsi untuk memuat halaman, mendapatkan kata dari halaman sebelumnya, membagi kata secara terpisah-pisah, menambahkan permainan ke dalam basis data, memberikan perintah ketika menekan tombol dan menyuarakan kata. Kelas Keempat yaitu Halaman Alfabet yang memiliki atribut yaitu variabel kata, alfabet dan tombol suara serta fungsi untuk memuat halaman, mendapatkan kata dari halaman sebelumnya, membagi kata menjadi karakter, memberikan perintah ketika tombol ditekan dan menyuarakan suara dari karakter tersebut. Kelas kelima yaitu Halaman Tahapan Permainan yang memiliki relasi dengan Halaman *Level/ Permainan* dan Halaman Tahapan Latihan. Halaman Tahapan Permainan memiliki atribut yaitu variabel tahapan, jumlah tahapan dan tombol serta memiliki fungsi yaitu untuk memuat halaman, mendapatkan tahapan, menekan tombol dan menentukan tahapan. Kelas keenam yaitu Halaman *Level/ Permainan* yang memiliki relasi dengan Halaman Tebak Kata dari Gambar, Suara dan Video. Halaman *Level/ Permainan* memiliki atribut yaitu variabel daftar *level*, tahapan dan *level* serta memiliki fungsi yaitu memuat halaman, mendapatkan tahapan, mendapatkan *level*, menekan tombol, menentukan *level* dan menentukan tahapan. Selanjutnya kelas ketujuh yaitu Halaman Tebak Kata dari Gambar dengan memiliki atribut yaitu variabel tahapan, *level* dan gambar serta fungsi untuk memuat halaman, menentukan tahapan, menentukan *level*, mendapatkan permainan dan menampilkan gambar. Kelas kedelapan yaitu Halaman Tebak Kata dari Suara dengan memiliki atribut yaitu variabel tahapan, *level*, kata dan tombol suara serta fungsi untuk memuat halaman, menentukan tahapan, menentukan *level*, mendapatkan permainan, menekan tombol dan menyuarakan kata. Kelas kesembilan yaitu Halaman Tebak Kata dari Video dengan memiliki atribut yaitu variabel tahapan, *level*, video, waktu mulai, waktu saat ini, durasi dan video youtube serta fungsi untuk memuat halaman, menentukan tahapan, menentukan *level*, mendapatkan permainan, menampilkan video, memuat video dan mendapatkan waktu saat ini.



Gambar 5.12 Perancangan Diagram Class Aplikasi Reado



5.1.3 Perancangan Basis Data

Perancangan basis data ditunjukkan dengan menggunakan pemodelan dari ERD (*Entity Relationship Diagram*) yang dapat dilihat pada Gambar 5.13. Pemodelan ini terdiri atas dua entitas yaitu entitas *Quizzes* untuk menyimpan isi dari permainan di dalam aplikasi Reado. Entitas dari *Quizzes* memiliki atribut yang terdiri atas id sebagai atribut unik, *stage*, *level*, *question*, *answere*, *media*, *time_start* dan *duration*. Selain itu, entitas dari *Stages* digunakan untuk menyimpan tahapan-tahapan permainan dari aplikasi Reado. Entitas dari *Stages* memiliki atribut yang terdiri atas id sebagai atribut unik, *name*, *media* dan *description*. Adapun detail dari entitas dan atribut yang terdapat pada perancangan basis data aplikasi Reado dapat dilihat pada Tabel 5.1 dan 5.2.

Tabel 5.1 Perancangan Basis Data dari Entitas *Quizzes*

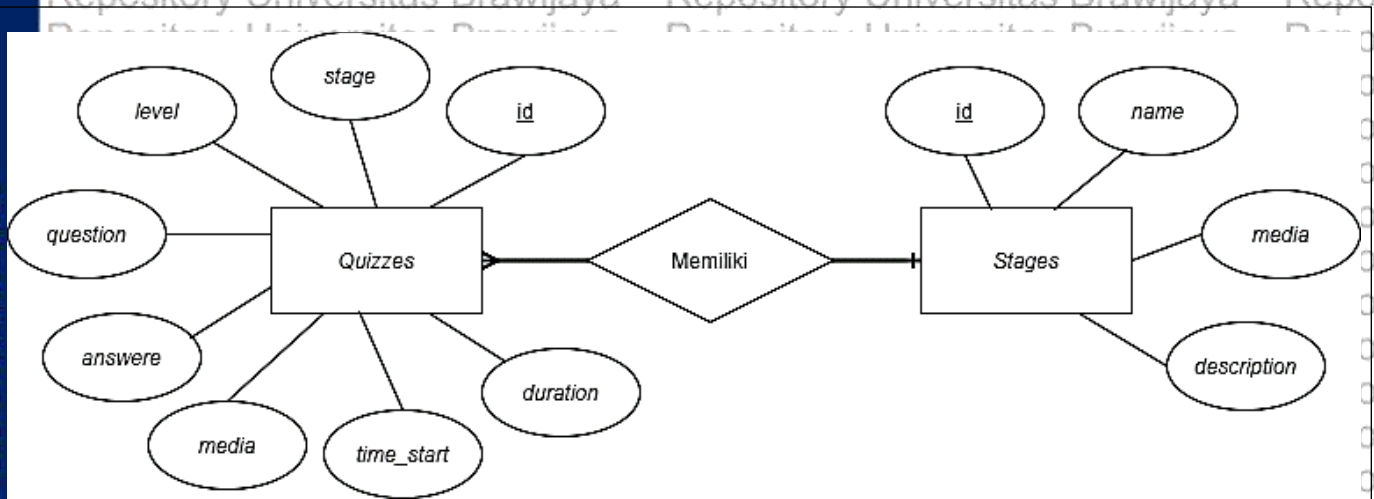
Entitas <i>Quizzes</i>			
No.	Atribut	Tipe Data	Keterangan
1.	id	<i>Integer</i>	Merupakan atribut yang digunakan sebagai atribut unik dari entitas <i>quizzes</i> .
2.	<i>stage</i>	<i>Integer</i>	Merupakan atribut yang menunjukkan nilai dari tahapan pada permainan.
3.	<i>level</i>	<i>Integer</i>	Merupakan atribut yang menunjukkan nilai dari <i>level</i> pada permainan.
4.	<i>question</i>	<i>Text</i>	Merupakan atribut yang menyimpan nilai dari huruf-huruf acak untuk pilihan jawaban dari permainan.
5.	<i>answere</i>	<i>Text</i>	Merupakan atribut yang menyimpan nilai dari jawaban dari permainan yang benar.
6.	<i>media</i>	<i>Text</i>	Merupakan atribut yang menyimpan nilai berupa gambar.
7.	<i>time_start</i>	<i>Integer</i>	Merupakan atribut yang menyimpan nilai berupa waktu mulai dari video.
8.	<i>duration</i>	<i>Integer</i>	Merupakan atribut yang menyimpan nilai berupa lamanya waktu untuk pemutaran video.

Tabel 5.2 Perancangan Basis Data dari Entitas *Stages*

Entitas <i>Stages</i>			
No.	Atribut	Tipe Data	Keterangan
1.	id	<i>Integer</i>	Merupakan atribut yang digunakan sebagai atribut unik dari entitas <i>stages</i> .

Tabel 5.2 Perancangan Basis Data dari Entitas *Stages* (lanjutan)

Entitas <i>Stages</i>			
No.	Atribut	Tipe Data	Keterangan
2.	<i>name</i>	Text	Merupakan atribut yang menyimpan nilai yaitu nama dari tahapan pada tahapan permainan.
3.	<i>description</i>	Text	Merupakan atribut yang menyimpan nilai yaitu deskripsi dari tahapan pada tahapan permainan.
4.	<i>media</i>	Text	Merupakan atribut yang menyimpan nilai berupa gambar dari logo setiap tahapan pada tahapan permainan.



Gambar 5.13 Perancangan ERD (*Entity Relationship Diagram*) Aplikasi Reado

5.1.4 Perancangan Algoritma

Perancangan algoritma merupakan perancangan dari alur logika yang nantinya akan dikembangkan menjadi kode program. Perancangan ini akan diimplementasikan menjadi kode program dengan menggunakan bahasa pemrograman Java. Perancangan algoritma menerapkan bentuk dari *pseudocode* yang mana menggunakan bentuk bahasa alami dari manusia. Penggunaan *pseudocode* ini bertujuan untuk dapat mendefinisikan kode program dari pengembangan aplikasi Reado sehingga memudahkan untuk memahami algoritma yang akan dikembangkan. Perancangan algoritma ini terdiri atas 10 perancangan algoritma dari setiap masing-masing kebutuhan-kebutuhan fungsional dari pengembangan aplikasi Reado. Adapun 10 perancangan algoritma dengan menggunakan bentuk *pseudocode* dari pengembangan aplikasi Reado, antara lain sebagai berikut:



1. Perancangan algoritma mendeteksi kata

Tabel 5.3 Perancangan Algoritma Mendeteksi Kata

Pseudocode Algoritma: Mendeteksi Kata	
Baris (ke-)	Pseudocode
1.	Start
2.	String kata = ""
3.	CameraSource kamera
4.	Button tombol
5.	
6.	Function memuatHalaman()
7.	menyalakanKamera()
8.	menerimaDeteksi()
9.	menekanTombol()
10.	End Function
11.	
12.	Function menyalakanKamera()
13.	Kamera->mulai()
14.	End Function
15.	
16.	Function menerimaDeteksi()
17.	menjalankan()
18.	For(int i = 0, i < kata->size())
19.	Kata->append("")
20.	End For
21.	End Function
22.	
23.	Function menekanTombol()
24.	If(kata->length() > 0
25.	startActivity()
26.	End If
27.	End Function
28.	
29.	End

Penjelasan dari *pseudocode* pada Tabel 5.3 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas kata dengan tipe data *string*, kamera dengan tipe data *camera source* dan tombol dengan tipe data *button*.
- Baris 6 sampai 10 merupakan fungsi untuk memuat halaman dari Halaman Deteksi Kata yang memanggil fungsi-fungsi yaitu fungsi untuk menyalakan kamera pada aplikasi, fungsi untuk menerima deteksi dari kata yang ditangkap oleh kamera, dan fungsi untuk menekan tombol.
- Baris 12 sampai 14 merupakan fungsi untuk menyalakan kamera dengan variabel yaitu kamera sebagai objek memanggil fungsi untuk menyalakan kamera dari kelas *Camera Source*.
- Baris 16 sampai 21 merupakan fungsi untuk menerima deteksi dari kata yang ditangkap oleh kamera. Terdapat algoritma sederhana untuk menerima deteksi dari kata yaitu memanggil fungsi untuk menjalankan deteksi kata dari fitur *text recognition* pada Mobile Vision API dan menggunakan perulangan *for* dengan batasan jika variabel *i* lebih kecil dari ukuran kata untuk



memberikan jarak kepada kata yang didapatkan sehingga ketika ditampilkan setiap kata dipisahkan oleh jarak.

- Baris 23 sampai 27 merupakan fungsi untuk menekan tombol dari Halaman Deteksi Kata. Fungsi ini terdiri atas algoritma untuk menuju halaman selanjutnya yaitu Halaman Kata jika sudah terdapat kata yang terdeteksi.

2. Perancangan algoritma menampilkan kata-kata dari hasil deteksi kata

Tabel 5.4 Perancangan Algoritma Menampilkan Kata-Kata dari Hasil Deteksi Kata

Pseudocode Algoritma: Menampilkan Kata-Kata dari Hasil Deteksi Kata	
Baris ke-	Pseudocode
1.	Start
2.	String kata
3.	String ArrayList kataKata
4.	Button tombolSuara
5.	
6.	Function memuatHalaman()
7.	kata = mendapatkanKata()
8.	kataKata = membagiKata(kata)
9.	For(kataKata -> size())
10.	tambahkanPermainan(kataKata)
11.	End For
12.	menekanTombol()
13.	End Function
14.	
15.	Function menekanTombol()
16.	menyuarakan(kata)
17.	End Function
18.	
19.	End

Penjelasan dari *pseudocode* pada Tabel 5.4 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas kata dengan tipe data *string*, kataKata dengan tipe data *string arraylist* dan tombol suara dengan tipe data *button*.
- Baris 6 sampai 13 merupakan fungsi untuk memuat halaman dari Halaman Kata. Tahapan pertama menginisialisasikan variabel kata dengan nilai kembalian dari fungsi untuk mendapatkan kata dari halaman sebelumnya. Selanjutnya yaitu memanggil fungsi untuk membagi kata-kata yang terdapat pada variabel kata dan hasil dari fungsi ini akan disimpan pada variabel kataKata. Lalu terdapat perulangan *for* untuk menambahkan kata-kata ke dalam basis data dengan mengulang pertambahan tersebut sebanyak kata-kata yang terdeteksi. Selanjutnya yaitu memanggil fungsi untuk memberikan perintah ketika tombol ditekan.
- Baris 15 sampai 17 merupakan fungsi untuk memberikan perintah ketika tombol ditekan. Fungsi ini dapat untuk menyuarakan suara dari kata yang dipilih oleh pengguna dengan menggunakan *Google Text To Speech* sehingga dapat menyuarakan kata yang dipilih oleh pengguna dari pengembangan aplikasi Readoo.



3. Perancangan algoritma menampilkan alfabet dari kata berdasarkan hasil deteksi kata

Tabel 5.5 Perancangan Algoritma Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata

Pseudocode Algoritma: Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	
Baris ke-	Pseudocode
1.	Start
2.	String kata
3.	Char ArrayList alfabet
4.	Button tombolSuara
5.	
6.	Function memuatHalaman()
7.	kata = mendapatkanKata()
8.	alfabet = membagiKata(kata)
9.	menekanTombol()
10.	End Function
11.	
12.	Function menekanTombol()
13.	menyuarakan(alfabet)
14.	End Function
15.	
16.	End

Penjelasan dari *pseudocode* pada Tabel 5.5 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas kata dengan tipe data *string*, alfabet dengan tipe data *char arraylist* dan tombol suara dengan tipe data *button*.
- Baris 6 sampai 10 merupakan fungsi untuk memuat halaman dari Halaman Alfabet. Tahapan pertama menginisialisasikan variabel kata dengan nilai kembalian dari fungsi untuk mendapatkan kata dari halaman sebelumnya. Selanjutnya yaitu memanggil fungsi untuk membagi kata-kata yang terdapat pada variabel kata dan hasil dari fungsi ini akan disimpan pada variabel alfabet. Lalu tahapan selanjutnya yaitu memanggil fungsi untuk memberikan perintah ketika tombol ditekan.
- Baris 12 sampai 14 merupakan fungsi untuk memberikan perintah ketika tombol ditekan. Fungsi ini dapat untuk menyuarakan suara dari alfabet yang dipilih oleh pengguna dengan menggunakan *Google Text To Speech* sehingga dapat menyuarakan alfabet yang dipilih oleh pengguna dari pengembangan aplikasi Readoo.

4. Perancangan algoritma menentukan tahapan permainan

Tabel 5.6 Perancangan Algoritma Menentukan Tahapan Permainan

Pseudocode Algoritma: Menentukan Tahapan Permainan	
Baris ke-	Pseudocode
1.	Start
2.	ArrayList tahapan
3.	Integer jumlahTahapan



Tabel 5.6 Perancangan Algoritma Menentukan Tahapan Permainan (lanjutan)

Pseudocode Algoritma: Menentukan Tahapan Permainan	
Baris ke-	Pseudocode
4.	Button tombol
5.	
6.	Function memuatHalaman()
7.	tahap = mendapatkanTahapan()
8.	jumlahTahapan = tahap - > size() - 1
9.	menekanTombol()
10.	End Function
11.	
12.	Function menekanTombol()
13.	menentukanTahapan(tahap)
14.	End Function
15.	
16.	Function menentukanTahapan(Integer)
17.	Switch(tahap -> getCurrentItem())
18.	Case 0:
19.	startActivity(Halaman Permainan Tahapan Latihan)
20.	Case 1:
21.	startActivity(Tahapan 1, Halaman Level Permainan)
22.	Case 2:
23.	startActivity(Tahapan 2, Halaman Level Permainan)
24.	Case 3:
25.	startActivity(Tahapan 3, Halaman Level Permainan)
26.	End Switch
27.	End Function
28.	
29.	End

Penjelasan dari *pseudocode* pada Tabel 5.6 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas tahapan dengan tipe data *arraylist*, jumlah tahapan dengan tipe data *integer* dan tombol dengan tipe data *button*.
- Baris 6 sampai 10 merupakan fungsi untuk memuat halaman dari Halaman Tahapan Permainan. Tahapan pertama menginisialisasikan tahapan dengan nilai kembalian dari fungsi untuk mendapatkan tahapan yang didapatkan dari basis data. Selanjutnya yaitu menginisialisasikan variabel jumlah tahapan dengan nilai dari ukuran variabel tahapan dikurang 1 untuk menjadi nilai indeks dari *arraylist* yang dimulai dengan indeks ke-0. Lalu tahapan selanjutnya yaitu memanggil fungsi untuk memberikan perintah ketika tombol ditekan.
- Baris 12 sampai 14 merupakan fungsi untuk memberikan perintah ketika tombol ditekan. Fungsi ini akan memanggil fungsi untuk menentukan tahapan dari Halaman Tahapan Permainan.
- Baris 16 sampai 27 merupakan fungsi untuk menentukan tahapan pada permainan dari aplikasi ReadO. Fungsi untuk menentukan tahapan ini menggunakan *switch* untuk menentukan atau menyeleksi tahapan berdasarkan nilai dari tahapan yang dipilih. Jika nilai tahapan sama dengan 0, maka akan menuju Halaman Permainan Tahapan Latihan. Selain itu, jika nilai tahapan sama dengan 1, 2 atau 3, maka akan menuju Halaman *Level*



Permainan berdasarkan *level* dari tahapan permainan yang dipilih oleh pengguna.

5. Perancangan algoritma menentukan *level* permainan

Tabel 5.7 Perancangan Algoritma Menentukan *Level* Permainan

Pseudocode Algoritma: Menentukan <i>Level</i> Permainan	
Baris ke-	Pseudocode
1.	Start
2.	ArrayList daftarLevel
3.	Integer tahapan
4.	Integer level
5.	
6.	Function memuatHalaman()
7.	tahapan = mendapatkanTahapan()
8.	level = mendapatkanLevel()
9.	menekanTombol()
10.	End Function
11.	
12.	Function menekanTombol()
13.	menentukanLevel(level)
14.	menentukanTahapan(tahapan)
15.	End Function
16.	
17.	Function menentukanTahapan(Integer)
18.	Switch(tahapan -> getCurrentItem())
19.	Case 1:
20.	startActivity(Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar)
21.	Case 2:
22.	startActivity(Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara)
23.	Case 3:
24.	startActivity(Halaman Permainan Tahapan Tebak Kata Berdasarkan Video)
25.	End Switch
26.	End Function
27.	
28.	End

Penjelasan dari *pseudocode* pada Tabel 5.7 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel daftar *level* dengan tipe data *arraylist*, tahapan dengan tipe data *integer* dan *level* dengan tipe data *button*.
- Baris 6 sampai 10 merupakan fungsi untuk memuat halaman dari Halaman *Level* Permainan. Tahapan pertama menginisialisasikan tahapan dengan nilai kembalian dari fungsi untuk mendapatkan tahapan yang didapatkan dari Halaman Tahapan Permainan. Selanjutnya yaitu menginisialisasikan variabel *level* dengan nilai kembalian dari fungsi untuk mendapatkan *level* yang didapatkan dari basis data. Tahapan terakhir yaitu halaman akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan.
- Baris 12 sampai 15 merupakan fungsi untuk memberikan perintah ketika tombol ditekan. Fungsi ini akan memanggil fungsi untuk menentukan *level* dari *level* yang dipilih oleh pengguna dan akan memanggil fungsi untuk



menentukan tahapan dengan nilai tahapan yang sudah didapatkan dari Halaman Tahapan Permainan.

- Baris 17 sampai 26 merupakan fungsi untuk menentukan permainan berdasarkan *level* dan tahapan yang dipilih pada permainan dari aplikasi Reado. Fungsi untuk menentukan tahapan ini menggunakan *switch* untuk menentukan atau menyeleksi tahapan berdasarkan nilai dari tahapan yang dipilih. Jika nilai tahapan sama dengan 1, maka akan menuju Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar. Lalu, jika nilai tahapan sama dengan 2, maka akan menuju Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara. Selanjutnya, jika nilai tahapan sama dengan 3, maka akan menuju Halaman Permainan Tahapan Tebak Kata Berdasarkan Video.

6. Perancangan algoritma menampilkan permainan tahapan latihan

Tabel 5.8 Perancangan Algoritma Menampilkan Permainan Tahapan Latihan

Pseudocode Algoritma: Menampilkan Permainan Tahapan Latihan	
Baris ke-	Pseudocode
1.	Start
2.	Integer tahapan
3.	Integer nomorTahapan
4.	String kata
5.	Button tombolSuara
6.	
7.	Function memuatHalaman()
8.	tahapan = menentukanTahapan()
9.	nomorTahapan = menentukanNomorTahapan()
10.	kata = mendapatkanPermainan(tahapan, nomorTahapan)
11.	menekanTombol()
12.	End Function
13.	
14.	Function menekanTombol()
15.	menyuarakan(kata)
16.	End Function
17.	
18.	End

Penjelasan dari *pseudocode* pada Tabel 5.8 adalah sebagai berikut:

Baris 2 sampai 5 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel daftar tahapan dengan tipe data Integer, nomorTahapan dengan tipe data *integer* dan kata dengan tipe data *string*.

Baris 7 sampai 12 merupakan fungsi untuk memuat halaman dari Halaman Permainan Tahapan Latihan. Tahapan pertama menginisialisasikan variabel tahapan dengan nilai kembalian dari fungsi untuk mendapatkan tahapan yang didapatkan dari halaman sebelumnya. Selanjutnya yaitu menginisialisasikan variabel nomor tahapan dengan nilai kembalian dari fungsi untuk mengatur nomor tahapan pada permainan tahapan latihan. Lalu, halaman akan menginisialisasi variabel kata dengan memanggil fungsi untuk mendapatkan permainan dari basis data. Tahapan terakhir yaitu halaman akan memanggil fungsi untuk memberikan perintah ketika tombol ditekan.



- Baris 14 sampai 16 merupakan fungsi untuk memberikan perintah ketika tombol ditekan. Fungsi ini akan memanggil fungsi untuk menyuarakan kata yang harus ditebak dari permainan tahapan latihan.

- 7. Perancangan algoritma menampilkan permainan tahapan tebak kata berdasarkan gambar

Tabel 5.9 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar

Pseudocode Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	
Baris ke-	Pseudocode
1.	Start
2.	Integer tahapan
3.	Integer level
4.	Image gambar
5.	
6.	Function memuatHalaman()
7.	tahapan = menentukanTahapan()
8.	level = menentukanLevel()
9.	gambar = mendapatkanPermainan(tahapan, level)
10.	menampilkanGambar()
11.	End Function
12.	
13.	Function menampilkanGambar()
14.	setImage(gambar)
15.	End Function
16.	
17.	End

Penjelasan dari *pseudocode* pada Tabel 5.9 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel daftar tahapan dengan tipe data Integer, *level* dengan tipe data *integer* dan gambar dengan tipe data *image*.
- Baris 6 sampai 11 merupakan fungsi untuk memuat halaman dari Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar. Tahapan pertama menginisialisasikan variabel tahapan dengan nilai kembalian dari fungsi untuk mendapatkan nilai tahapan yang didapatkan dari halaman sebelumnya dan menginisialisasikan variabel *level* dengan nilai kembalian dari fungsi untuk mendapatkan nilai *level* dari halaman sebelumnya. Selanjutnya yaitu menginisialisasikan variabel gambar dengan nilai kembalian dari fungsi untuk mendapatkan permainan dari basis data. Tahapan terakhir yaitu halaman akan memanggil fungsi untuk menampilkan gambar dari kata yang harus ditebak dalam permainan tebak kata.
- Baris 13 sampai 15 merupakan fungsi untuk menampilkan gambar yang harus ditebak dari permainan tebak kata. Fungsi ini akan memanggil fungsi untuk menentukan gambar yang akan ditampilkan pada aplikasi.
- 8. Perancangan algoritma menampilkan permainan tahapan tebak kata berdasarkan suara



Tabel 5.10 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara

Pseudocode Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara		
Baris ke-	Pseudocode	
1.	Start	
2.	Integer tahapan	
3.	Integer level	
4.	String kata	
5.	Button tombolSuara	
6.		
7.	Function memuatHalaman()	
8.	tahapan = menentukanTahapan()	
9.	level = menentukanLevel()	
10.	kata = mendapatkanPermainan(tahapan, level)	
11.	menekanTombol()	
12.	End Function	
13.		
14.	Function menekanTombol()	
15.	menyuarakan(kata)	
16.	End Function	
17.		
18.	End	

Penjelasan dari *pseudocode* pada Tabel 5.10 adalah sebagai berikut:

- Baris 2 sampai 4 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel daftar tahapan dengan tipe data *integer*, *level* dengan tipe data *integer*, kata dengan tipe data *string* dan tombol suara dengan tipe data *button*.
- Baris 7 sampai 12 merupakan fungsi untuk memuat halaman dari Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara. Tahapan pertama akan menginisialisasikan variabel tahapan dengan nilai kembalian dari fungsi untuk mendapatkan nilai tahapan yang didapatkan dari halaman sebelumnya dan menginisialisasikan variabel *level* dengan nilai kembalian dari fungsi untuk mendapatkan nilai *level* dari halaman sebelumnya. Tahapan selanjutnya yaitu menginisialisasikan variabel kata dengan nilai kembalian dari fungsi untuk mendapatkan permainan yaitu berupa kata yang harus ditebak yang mana kata tersebut didapatkan dari basis data. Tahapan terakhir yaitu halaman akan memanggil fungsi untuk memberikan perintah ketika tombol suara ditekan.
- Baris 14 sampai 16 merupakan fungsi untuk memberikan perintah ketika tombol suara ditekan atau dipilih oleh pengguna. Fungsi ini akan memanggil fungsi yaitu untuk menyuarakan dari variabel kata yang harus ditebak dalam Bahasa Inggris dari permainan tahapan tebak kata berdasarkan suara pada aplikasi Readod.
- 9. Perancangan algoritma menampilkan permainan tahapan tebak kata berdasarkan video



Tabel 5.11 Perancangan Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video

Pseudocode Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	
Baris ke-	Pseudocode
1.	Start
2.	Integer tahapan
3.	Integer level
4.	String video
5.	Integer waktuMulai
6.	Integer waktuSaatIni
7.	Integer durasi
8.	YoutubePlayer youtubeVideo
9.	
10.	Function memuatHalaman()
11.	tahapan = menentukanTahapan()
12.	level = menentukanLevel()
13.	video = mendapatkanPermainan(tahapan, level)
14.	waktuMulai = mendapatkanPermainan(tahapan, level)
15.	durasi = mendapatkanPermainan(tahapan, level)
16.	menampilkanVideo()
17.	End Function
18.	
19.	Function menampilkanVideo()
20.	youtubeVideo -> memuatVideo(video, waktuMulai)
21.	waktuSaatIni = mendapatkanWaktuSaatIni()
22.	If (waktuSaatIni > durasi)
23.	youtubeVideo -> berhenti()
24.	End If
25.	End Function
26.	
27.	End

Penjelasan dari *pseudocode* pada Tabel 5.11 adalah sebagai berikut:

- Baris 2 sampai 8 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel tahapan dengan tipe data *integer*, *level* dengan tipe data *integer*, video dengan tipe data *string*, waktu mulai dengan tipe data *integer*, waktu saat ini dengan tipe data *integer*, durasi dengan tipe data *integer* dan youtubeVideo dengan tipe data *Youtube Player*.
- Baris 10 sampai 17 merupakan fungsi untuk memuat halaman dari Halaman Permainan Tahapan Tebak Kata Berdasarkan Video. Tahapan pertama akan menginisialisasikan variabel tahapan dengan nilai kembalian dari fungsi untuk mendapatkan nilai tahapan yang didapatkan dari halaman sebelumnya dan menginisialisasikan variabel *level* dengan nilai kembalian dari fungsi untuk mendapatkan nilai *level* dari halaman sebelumnya. Tahapan selanjutnya yaitu menginisialisasikan variabel video, variabel waktu mulai dan variabel durasi dengan nilai kembalian dari fungsi untuk mendapatkan permainan yang didapatkan dari basis data. Tahapan terakhir yaitu halaman akan memanggil fungsi untuk menampilkan video.
- Baris 19 sampai 25 merupakan fungsi untuk menampilkan video pada permainan tebak kata berdasarkan video. Tahapan pertama dari fungsi ini yaitu objek *youtube video* akan memanggil fungsi untuk memuat video



dengan menggunakan Youtube Android Player API. Selanjutnya yaitu menginisialisasi variabel waktu saat ini dengan memanggil fungsi untuk mendapatkan waktu saat ini. Lalu, halaman melakukan penyeleksian *if* dengan kondisi jika nilai variabel waktu saat ini lebih dari nilai variabel durasi, maka objek *youtube video* akan memanggil fungsi untuk memberhentikan video atau menjeda video.

10. Perancangan algoritma menyediakan mekanisme untuk jawaban dari permainan

Tabel 5.12 Perancangan Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan

Pseudocode Algoritma: Menyediakan Mekanisme untuk Jawaban dari Permainan	
Baris ke	Pseudocode
1.	Start
2.	Integer tahapan
3.	Integer level
4.	Integer indeks
5.	Char alfabet
6.	String jawabanPengguna
7.	String jawabanBenar
8.	Char Array pilihanAlfabet
9.	Button tombolAlfabet
10.	Button tombol
11.	
12.	Function memuatHalaman()
13.	jawabanBenar = mendapatkanJawabanBenar(tahapan, level)
14.	pilihanAlfabet = mengacakJawaban(jawabanBenar)
15.	menampilkanPilihan(pilihanAlfabet)
16.	menekanTombolPilihanAlfabet()
17.	menekanTombolMengaturUlang()
18.	menentukanJawaban(jawabanPengguna)
19.	menekanTombolHasilJawaban()
20.	End Function
21.	
22.	Function mengacakJawaban(String)
23.	For(i = pilihanAlfabet -> length() -1, i > 0, i--)
24.	indeks = i + 1
25.	alfabet = pilihanAlfabet[indeks]
26.	pilihanAlfabet[indeks] = pilihanAlfabet[i]
27.	pilihanAlfabet[i] = alfabet
28.	End For
29.	End Function
30.	
31.	Function menekanTombolPilihanAlfabet()
32.	jawabanPengguna += mendapatkanJawaban()
33.	menampilkanJawaban(jawabanPengguna)
34.	End Function
35.	
36.	Function menekanTombolMengaturUlang()
37.	jawabanPengguna = ""
38.	pilihanAlfabet = mengacakJawaban(jawabanBenar)
39.	menampilkanPilihan(pilihanAlfabet)
40.	End Function
41.	
42.	Function menekanTombolHasilJawaban()
43.	startActivity(tahapan, level)
44.	End Function



Tabel 5.12 Perancangan Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan (lanjutan)

Pseudocode Algoritma: Menyediakan Mekanisme untuk Jawaban dari Permainan	
Baris ke-	Pseudocode
45.	Function menentukanJawaban()
46.	If (jawabanPengguna == jawabanBenar)
47.	MenampilkanHasilJawaban("Kamu Benar!")
48.	MenampilkanTombol("Berikutnya")
49.	level = level + 1
50.	Else
51.	MenampilkanHasilJawaban("Oops, Kamu Salah!")
52.	MenampilkanTombol("Coba Kembali")
53.	End Function
54.	
55.	End

Penjelasan dari *pseudocode* pada Tabel 5.11 adalah sebagai berikut:

- Baris 2 sampai 10 yaitu mendeklarasikan variabel berdasarkan tipe data tertentu yang terdiri atas variabel tahapan dengan tipe data *integer*, *level* dengan tipe data *integer*, indeks dengan tipe data *integer*, alfabet dengan tipe data *char*, jawaban pengguna dengan tipe data *string*, jawaban benar dengan tipe data *string*, pilihan alfabet dengan tipe data *char array*, tombol alfabet dengan tipe data *button* dan tombol dengan tipe data *button*.
- Baris 12 sampai 20 merupakan fungsi untuk memuat halaman dengan tahapan pertama yaitu akan menginisialisasikan variabel jawaban benar dengan nilai kembalian dari fungsi untuk mendapatkan jawaban benar yang didapatkan dari basis data dan menginisialisasikan variabel pilihan alfabet dengan nilai kembalian dari fungsi untuk mengacak pilihan jawaban dari variabel jawaban benar. Tahapan selanjutnya yaitu menampilkan pilihan jawaban berdasarkan variabel pilihan alfabet. Lalu memanggil fungsi untuk memberikan perintah ketika tombol pilihan alfabet ditekan dan memanggil fungsi untuk mengatur ulang jawaban yang sudah dijawab oleh pengguna. Selanjutnya yaitu menentukan jawaban berdasarkan jawaban pengguna yang sudah dijawab dan memanggil fungsi untuk memberikan perintah ketika tombol dari hasil jawaban ditekan oleh pengguna.
- Baris 22 sampai 29 merupakan fungsi untuk mengacak pilihan jawaban berdasarkan jawaban yang benar dari basis data. Fungsi ini menggunakan pengulangan *for* dengan penginisialisasian yaitu nilai variabel *i* sama dengan nilai panjang dari variabel pilihan alfabet dikurang 1 dan kondisi yaitu nilai variabel *i* lebih dari 0 serta pengulangan dilakukan dengan kondisi nilai dari variabel *i* dikurangi 1. Pengulangan yang dilakukan terdiri dari beberapa algoritma yaitu nilai variabel indeks sama dengan nilai variabel *i* ditambah *a*, nilai variabel alfabet sama dengan *array* dari pilihan alfabet dengan indeks dari nilai variabel indeks, *array* pilihan alfabet dengan indeks yaitu nilai dari variabel indeks sama dengan *array* pilihan alfabet dengan indeks ke-*i*, dan



array pilihan alfabet dengan indeks ke-*i* sama dengan nilai dari variabel alfabet.

- Baris 31 sampai 34 merupakan fungsi untuk memberikan perintah ketika pengguna menekan tombol dari pilihan alfabet. Fungsi ini terdiri dari beberapa algoritma yaitu nilai dari variabel jawaban pengguna sama dengan nilai dari variabel jawaban pengguna tersebut ditambahkan dengan nilai kembalian dari fungsi untuk mendapatkan jawaban dari pengguna dan algoritma yaitu memanggil fungsi untuk menampilkan jawaban dari pengguna.
- Baris 36 sampai 40 merupakan fungsi untuk memberikan perintah ketika pengguna menekan tombol untuk mengatur ulang jawaban dari pengguna. Fungsi ini terdiri dari beberapa algoritma yaitu nilai dari variabel jawaban pengguna sama dengan nilai *string* kosong, nilai dari variabel *array* pilihan alfabet sama dengan nilai kembalian dari fungsi untuk mengacak pilihan jawaban dari jawaban yang benar dan algoritma yaitu memanggil fungsi untuk menampilkan pilihan alfabet.
- Baris 42 sampai 44 merupakan fungsi untuk memberikan perintah ketika pengguna menekan tombol dari hasil jawaban yang ditampilkan. Fungsi ini akan memanggil fungsi untuk menuju kepada halaman permainan berdasarkan nilai dari variabel tahapan dan nilai variabel *level*.
- Baris 45 sampai 53 merupakan fungsi untuk menentukan jawaban yang sudah dijawab oleh pengguna. Fungsi ini menggunakan penyeleksian *if* dengan kondisi yaitu jika nilai dari variabel jawaban pengguna sama dengan nilai dari variabel jawaban benar maka akan dijalankan perintah yaitu memanggil fungsi untuk menampilkan hasil jawaban yaitu "Kamu Benar!", fungsi untuk menampilkan tombol yaitu "Berikutnya" dan nilai dari variabel *level* akan ditambahkan 1. Namun jika nilai dari variabel jawaban salah maka akan dijalankan perintah yaitu memanggil fungsi untuk menampilkan hasil jawaban yaitu "Oops, Kamu Salah!" dan fungsi untuk menampilkan tombol yaitu "Coba Kembali".

5.1.5 Perancangan Antarmuka

Di dalam melakukan perancangan antarmuka pada pengembangan aplikasi Reado, penelitian ini menggunakan metode *Design Sprint* yang dilakukan oleh tim pengembang pada hari kedua sampai hari kelima pengembangan aplikasi Reado yaitu tahapan *ideate*, *decide*, *prototype* dan *test*. Tahapan ini dilakukan untuk membuat prototipe yang sudah tervalidasi oleh subjek penelitian yaitu pengajar yang mana akan digunakan untuk diimplementasikan atau dikembangkan menjadi aplikasi Reado dalam bentuk kode program sehingga dapat digunakan oleh pengguna yaitu anak-anak. Adapun penjelasan detail terkait dengan langkah-langkah pada tahapan *ideate*, *decide*, *prototype*, dan tahapan *test* yang dilakukan di dalam penelitian ini yaitu untuk membuat perancangan antarmuka pada pengembangan aplikasi Reado, antara lain sebagai berikut:



5.1.5.1 Melakukan *Lightning Demos*

Pada tahapan awal untuk perancangan antarmuka dari aplikasi Reado, dilakukan *lightning demos* yaitu dengan merepresentasikan beberapa solusi dari tampilan-tampilan aplikasi dari beberapa kompetitor sehingga dapat memberikan gambaran untuk tampilan aplikasi yang akan dikembangkan selanjutnya. Tampilan-tampilan aplikasi dari beberapa kompetitor diambil berdasarkan kelebihan-kelebihan dari tampilan aplikasi yang dapat menarik perhatian anak untuk menggunakan aplikasi tersebut. Adapun penjelasan dari aplikasi kompetitor yang menjadi referensi dalam pengembangan aplikasi Reado dijelaskan pada Tabel 5.13.

Tabel 5.13 Aplikasi Kompetitor dari Pengembangan Aplikasi Reado

No.	Nama Aplikasi	Deskripsi Aplikasi
1.	Aplikasi Kuis Gambar Saku	Aplikasi Kuis Gambar Saku merupakan aplikasi permainan untuk menebak gambar-gambar yang disukai oleh anak-anak.
2.	Aplikasi Cerdas Cermat SMA	Aplikasi Cerdas Cermat SMA merupakan aplikasi edukasi interaktif yang terdiri dari soal-soal pelajaran untuk Sekolah Menengah Atas (SMA) sehingga siswa SMA dapat melatih pemahaman mereka dengan aplikasi yang menerapkan permainan interaktif ini.
3.	Aplikasi <i>Baby First Words Flashcards – Kids Learning Games</i>	Aplikasi <i>Baby First Words Flashcards</i> merupakan aplikasi dalam bentuk kartu <i>flash</i> yang membantu bayi, balita dan anak-anak pra sekolah untuk mempelajari dan memperluas kosakata baik dalam bentuk kata-kata maupun suara.
4.	Aplikasi <i>First Words for Baby</i>	Aplikasi <i>First Words for Baby</i> merupakan aplikasi yang membantu bayi untuk mempelajari lebih dari 120 kosakata-kosakata umum yang dipilih dalam bentuk kartu <i>flash</i> .

5.1.5.2 Membuat Sketsa

Tahapan selanjutnya dalam perancangan antarmuka dari pengembangan aplikasi Reado yaitu tim pengembang melakukan pembuatan sketsa. Pembuatan sketsa ini dilakukan untuk menggambarkan beberapa solusi berdasarkan kebutuhan-kebutuhan perangkat lunak untuk tampilan dari aplikasi Reado. Pembuatan sketsa ini dilakukan bersama dengan tim pengembang yang mana masing-masing anggota tim pengembang mendefinisikan gambaran untuk sketsa tampilan dari aplikasi Reado. Berdasarkan hasil pendefinisian tersebut, tim pengembang menghasilkan 25 tampilan sketsa untuk solusi dari kebutuhan-kebutuhan aplikasi Reado yang akan dikembangkan. Tampilan sketsa yang sudah didefinisikan oleh masing-masing anggota dari tim pengembang akan dijadikan sebagai referensi untuk menentukan tampilan yang akan dikembangkan untuk aplikasi Reado.



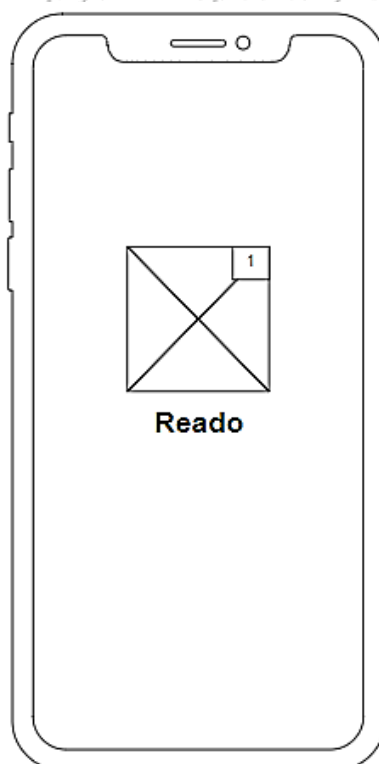
5.1.5.3 Melakukan *Sticky Decision*

Tahapan selanjutnya yaitu tim pengembang akan melakukan *sticky decision*. Tahapan *sticky decision* merupakan tahapan yang dilakukan untuk menyeleksi dan menentukan tampilan yang akan dikembangkan untuk aplikasi Reado. Penentuan tampilan ini menggunakan referensi tampilan berdasarkan pembuatan sketsa yang sudah dilakukan pada tahapan sebelumnya yang terdiri atas 25 tampilan sketsa. Tahapan ini dilakukan dengan cara berdiskusi bersama dengan tim pengembang untuk menentukan sketsa yang akan dijadikan tampilan dari pengembangan aplikasi Reado. Hasil dari tahapan *sticky decision* ini yaitu tampilan-tampilan yang akan dikembangkan untuk tampilan aplikasi Reado yang terdiri dari 11 tampilan aplikasi.

5.1.5.4 Membuat *Storyboard*

Tahapan ini merupakan tahapan yang dilakukan oleh tim pengembang untuk membuat *storyboard* dari tampilan dalam pengembangan aplikasi Reado. *Storyboard* merupakan gambaran dari cara kerja atau alur pada tampilan-tampilan aplikasi secara keseluruhan dan sistematis. Pembuatan *storyboard* ini dilakukan berdasarkan hasil dari tahapan sebelumnya yaitu tahapan untuk melakukan *sticky decision*. Hasil tahapan *sticky decision* ini terdiri atas 11 tampilan aplikasi yang akan dikembangkan untuk aplikasi Reado. Adapun tampilan-tampilan dari aplikasi Reado yang akan dikembangkan dan penjelasan detail terkait dengan tampilan aplikasi tersebut, antara lain sebagai berikut:

1. Tampilan aplikasi untuk logo aplikasi Reado



Gambar 5.14 Tampilan Aplikasi untuk Logo Aplikasi Reado

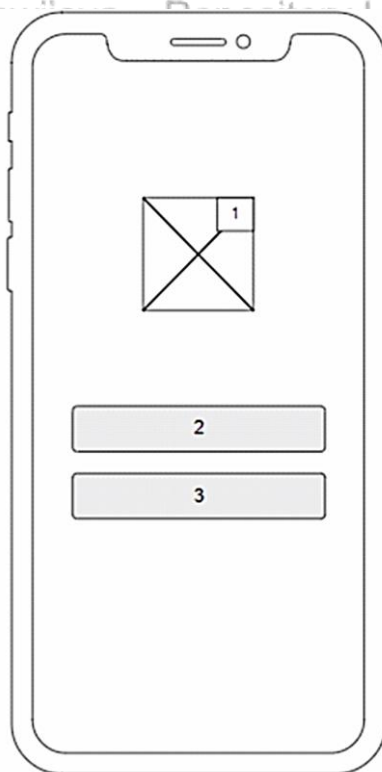


Adapun penjelasan dari tampilan aplikasi untuk logo aplikasi Reado pada Gambar 5.14 dapat dilihat pada Tabel 5.14.

Tabel 5.14 Penjelasan Tampilan Aplikasi untuk Logo Aplikasi Reado

No.	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Gambar logo dari aplikasi Reado

2. Tampilan aplikasi untuk halaman utama aplikasi Reado



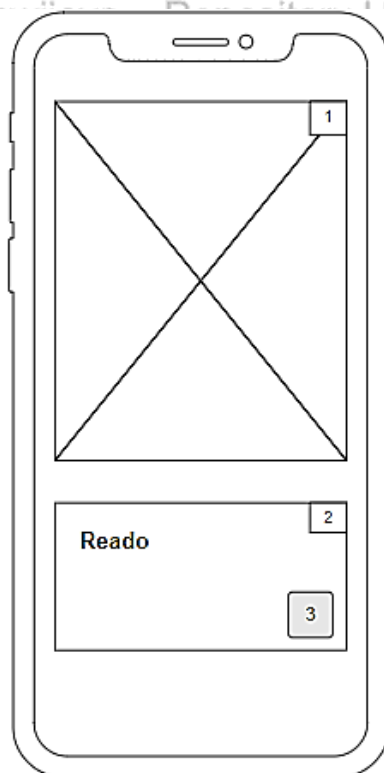
Gambar 5.15 Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado

Adapun penjelasan dari tampilan aplikasi untuk halaman utama aplikasi Reado pada Gambar 5.15 dapat dilihat pada Tabel 5.15.

Tabel 5.15 Penjelasan Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado

No.	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Gambar logo dari aplikasi Reado
2	Tombol Ambil Kata	Tombol	Tombol untuk mendeteksi kata
3	Tombol Game Kuis	Tombol	Tombol untuk memainkan permainan

3. Tampilan aplikasi untuk halaman deteksi kata



Gambar 5.16 Tampilan Aplikasi untuk Halaman Deteksi Kata

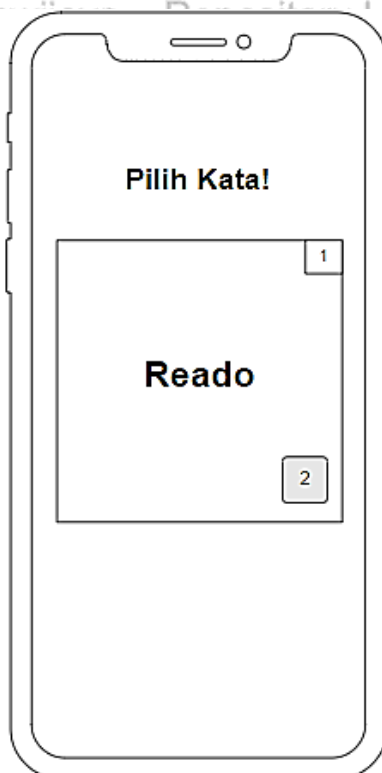
Adapun penjelasan dari tampilan aplikasi untuk halaman deteksi kata pada Gambar 5.16 dapat dilihat pada Tabel 5.16.

Tabel 5.16 Penjelasan Tampilan Aplikasi untuk Halaman Deteksi Kata

No.	Nama Objek	Tipe	Keterangan
1	Tampilan Kamera	SurfaceView	Tampilan kamera untuk mengambil atau menangkap kata
2	Teks Kata yang Diambil atau Ditangkap	Teks	Teks kata yang menunjukkan kata yang diambil atau ditangkap oleh kamera
3	Tombol Selanjutnya	Tombol	Tombol untuk menyetujui kata yang sudah diambil atau ditangkap dan untuk menuju ke halaman selanjutnya

4. Tampilan aplikasi untuk halaman kata dari hasil deteksi kata

Tampilan aplikasi untuk halaman kata dari hasil deteksi kata memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman kata dari hasil deteksi kata dapat dilihat pada Gambar 5.17 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.17.



Gambar 5.17 Tampilan Aplikasi untuk Halaman Kata dari Hasil Deteksi Kata

Tabel 5.17 Penjelasan Tampilan Aplikasi untuk Halaman Kata

No.	Nama Objek	Tipe	Keterangan
1	Daftar Hasil Kata yang Dideteksi	<i>RecyclerView</i>	Daftar hasil kata yang dideteksi oleh kamera yang terdiri atas kata-kata secara terpisah.
2	Tombol Suara	Tombol	Tombol untuk mendengarkan cara pembacaan kata dalam Bahasa Inggris

5. Tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata

Tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata dapat dilihat pada Gambar 5.18 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.18.

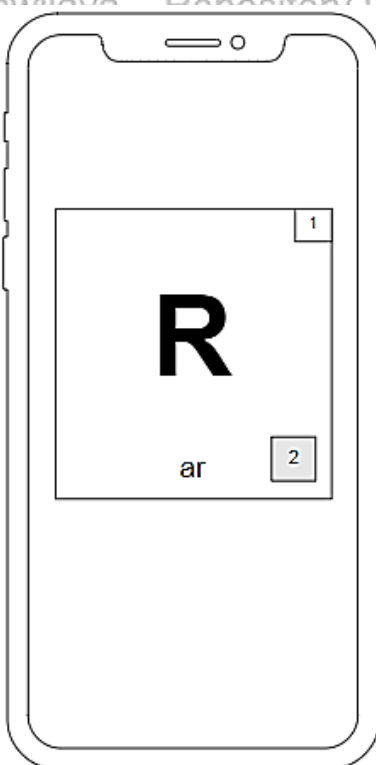
Tabel 5.18 Penjelasan Tampilan Aplikasi untuk Halaman Alfabet

No.	Nama Objek	Tipe	Keterangan
1	Daftar Alfabet	<i>RecyclerView</i>	Daftar alfabet dari kata yang dipilih ditampilkan secara terpisah.



Tabel 5.18 Penjelasan Tampilan Aplikasi untuk Halaman Alfabet (lanjutan)

No.	Nama Objek	Tipe	Keterangan
2	Tombol Suara	Tombol	Tombol untuk mendengarkan cara pembacaan alfabet dalam Bahasa Inggris



Gambar 5.18 Tampilan Aplikasi untuk Halaman Alfabet dari Hasil Deteksi Kata

6. Tampilan aplikasi untuk halaman tahapan permainan

Tampilan aplikasi untuk halaman tahapan permainan memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman tahapan permainan dapat dilihat pada Gambar 5.19 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.19.

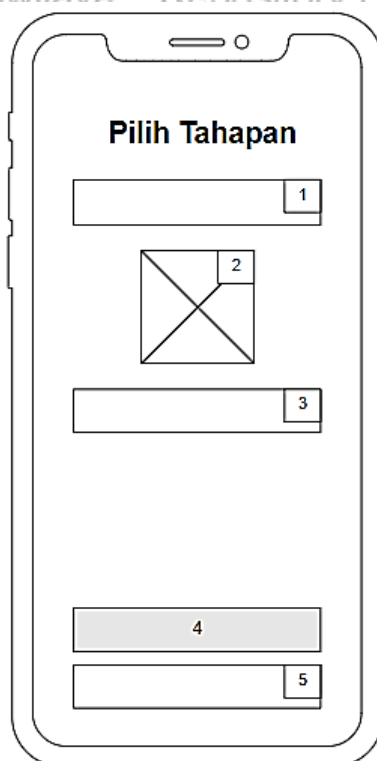
Tabel 5.19 Penjelasan Tampilan Aplikasi untuk Halaman Tahapan Permainan

No.	Nama Objek	Tipe	Keterangan
1	Teks Nama Tahapan	Teks	Teks yang menunjukkan nama dari tahapan permainan
2	Logo Tahapan	Gambar	Gambar logo dari setiap tahapan permainan
3	Teks Deskripsi Tahapan	Teks	Teks yang menunjukkan deskripsi dari setiap tahapan permainan



Tabel 5.19 Penjelasan Tampilan Aplikasi untuk Halaman Tahapan Permainan (lanjutan)

No.	Nama Objek	Tipe	Keterangan
4	Tombol Selanjutnya	Tombol	Tombol untuk memainkan tahapan yang dipilih
5	Tombol Kembali ke Beranda	Teks	Teks yang memiliki <i>link</i> dan sebagai tombol untuk menuju ke halaman beranda



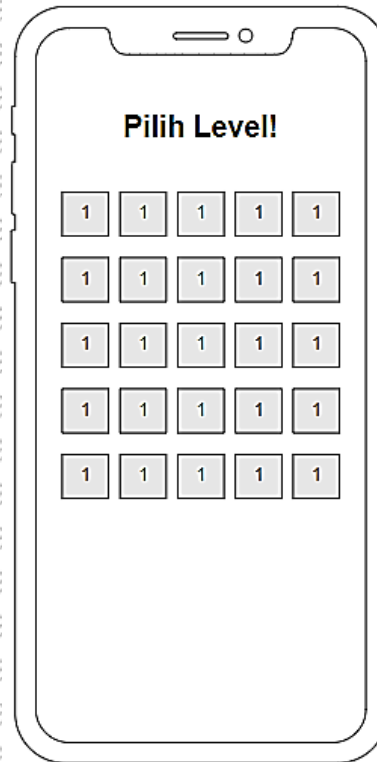
Gambar 5.19 Tampilan Aplikasi untuk Halaman Tahapan Permainan

7. Tampilan aplikasi untuk halaman *level* permainan

Tampilan aplikasi untuk halaman *level* permainan memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman *level* permainan dapat dilihat pada Gambar 5.20 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.20.

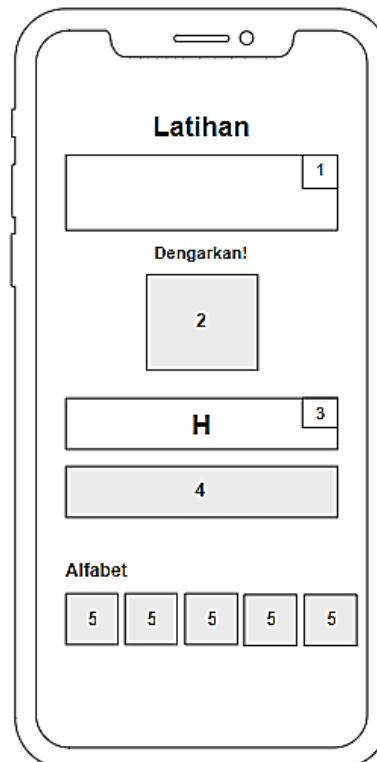
Tabel 5.20 Penjelasan Tampilan Aplikasi untuk Halaman *Level* Permainan

No.	Nama Objek	Tipe	Keterangan
1	Tombol <i>Level</i>	Tombol	Tombol untuk memilih <i>level</i> pada tahapan permainan



Gambar 5.20 Tampilan Aplikasi untuk Halaman Level Permainan

8. Tampilan aplikasi untuk halaman permainan tahapan latihan



Gambar 5.21 Tampilan Aplikasi untuk Halaman Permainan Tahapan Latihan



Adapun penjelasan dari tampilan aplikasi untuk halaman permainan tahapan latihan pada Gambar 5.21 dapat dilihat pada Tabel 5.21.

Tabel 5.21 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Latihan

No.	Nama Objek	Tipe	Keterangan
1	Teks Deskripsi Tahapan Latihan	Teks	Teks yang menampilkan deskripsi dari tahapan latihan
2	Tombol Suara untuk Tebak Kata	Tombol	Tombol untuk mendengarkan kata yang harus ditebak
3	Teks Jawaban	Teks	Teks yang menampilkan jawaban yang sudah dijawab oleh pengguna
4	Tombol Mengatur Ulang	Tombol	Tombol untuk mengatur ulang jawaban tebak kata
5	Daftar Pilihan Jawaban Tebak Kata	<i>RecyclerView</i>	Daftar pilihan dari jawaban tebak kata yang dapat dipilih oleh pengguna

9. Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar

Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar dapat dilihat pada Gambar 5.22 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.22.

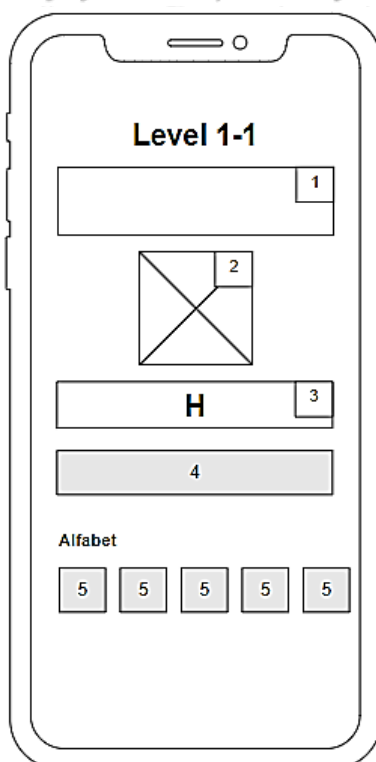
Tabel 5.22 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar

No.	Nama Objek	Tipe	Keterangan
1	Teks Deskripsi Tahapan Tebak Kata	Teks	Teks yang menampilkan deskripsi dari tahapan tebak kata berdasarkan gambar
2	Gambar Tebak Kata	Gambar	Gambar yang harus ditebak pada tahapan tebak kata
3	Teks Jawaban	Teks	Teks yang menampilkan jawaban yang sudah dijawab oleh pengguna
4	Tombol Mangatur Ulang	Tombol	Tombol untuk mengatur ulang jawaban tebak kata



Tabel 5.22 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar (lanjutan)

No.	Nama Objek	Tipe	Keterangan
5	Daftar Pilihan Jawaban Tebak Kata	RecyclerView	Daftar pilihan jawaban tebak kata yang dapat dipilih oleh pengguna



Gambar 5.22 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Gambar

10. Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara

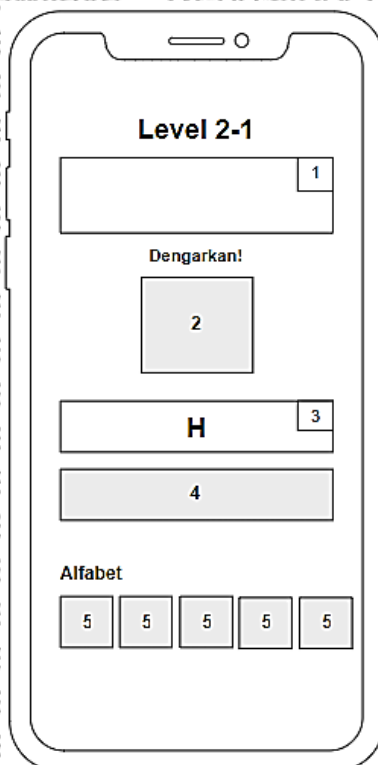
Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara dapat dilihat pada Gambar 5.23 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.23.

Tabel 5.23 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara

No.	Nama Objek	Tipe	Keterangan
1	Teks Deskripsi Tahapan Tebak Kata	Teks	Teks yang menampilkan deskripsi dari tahapan tebak kata berdasarkan suara

Tabel 5.23 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara (lanjutan)

No.	Nama Objek	Tipe	Keterangan
2	Tombol Suara Untuk Tebak Kata	Tombol	Tombol untuk mendengarkan kata yang harus ditebak
3	Teks Jawaban	Teks	Teks yang menampilkan jawaban yang sudah dijawab oleh pengguna
4	Tombol Mangatur Ulang	Tombol	Tombol untuk mengatur ulang jawaban tebak kata
5	Daftar Pilihan Jawaban Tebak Kata	Recycler View	Daftar pilihan jawaban tebak kata yang dapat dipilih oleh pengguna



Gambar 5.23 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Suara

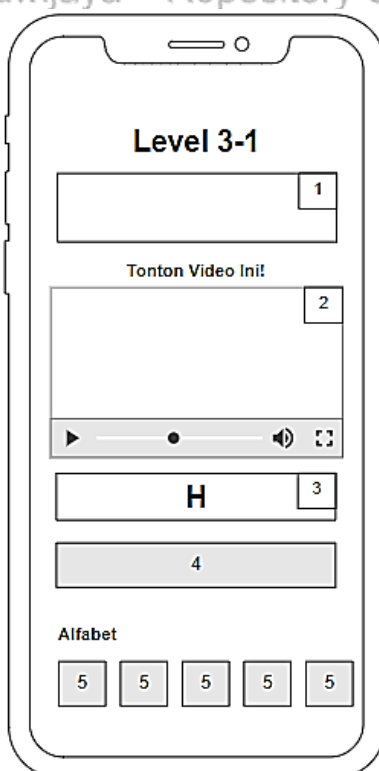
11. Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video

Tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video memiliki penjelasan terkait dengan objek-objek yang terdapat pada tampilan aplikasi tersebut. Adapun tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video dapat dilihat pada Gambar 5.24 dan penjelasan dari tampilan aplikasi untuk halaman ini juga dapat dilihat dalam Tabel 5.24.



Tabel 5.24 Penjelasan Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Video

No.	Nama Objek	Tipe	Keterangan
1	Teks Deskripsi Tahapan Tebak Kata	Teks	Teks yang menampilkan deskripsi tahapan tebak kata berdasarkan video
2	Video Tebak Kata	<i>YoutubePlayerView</i>	Video yang menampilkan tampilan untuk ditebak kata
3	Teks Jawaban	Teks	Teks yang menampilkan jawaban yang sudah dijawab oleh pengguna
4	Tombol Mengatur Ulang	Tombol	Tombol untuk mengatur ulang jawaban tebak kata
5	Daftar Pilihan Jawaban Tebak Kata	<i>RecyclerView</i>	Daftar pilihan jawaban tebak kata yang dapat dipilih oleh pengguna

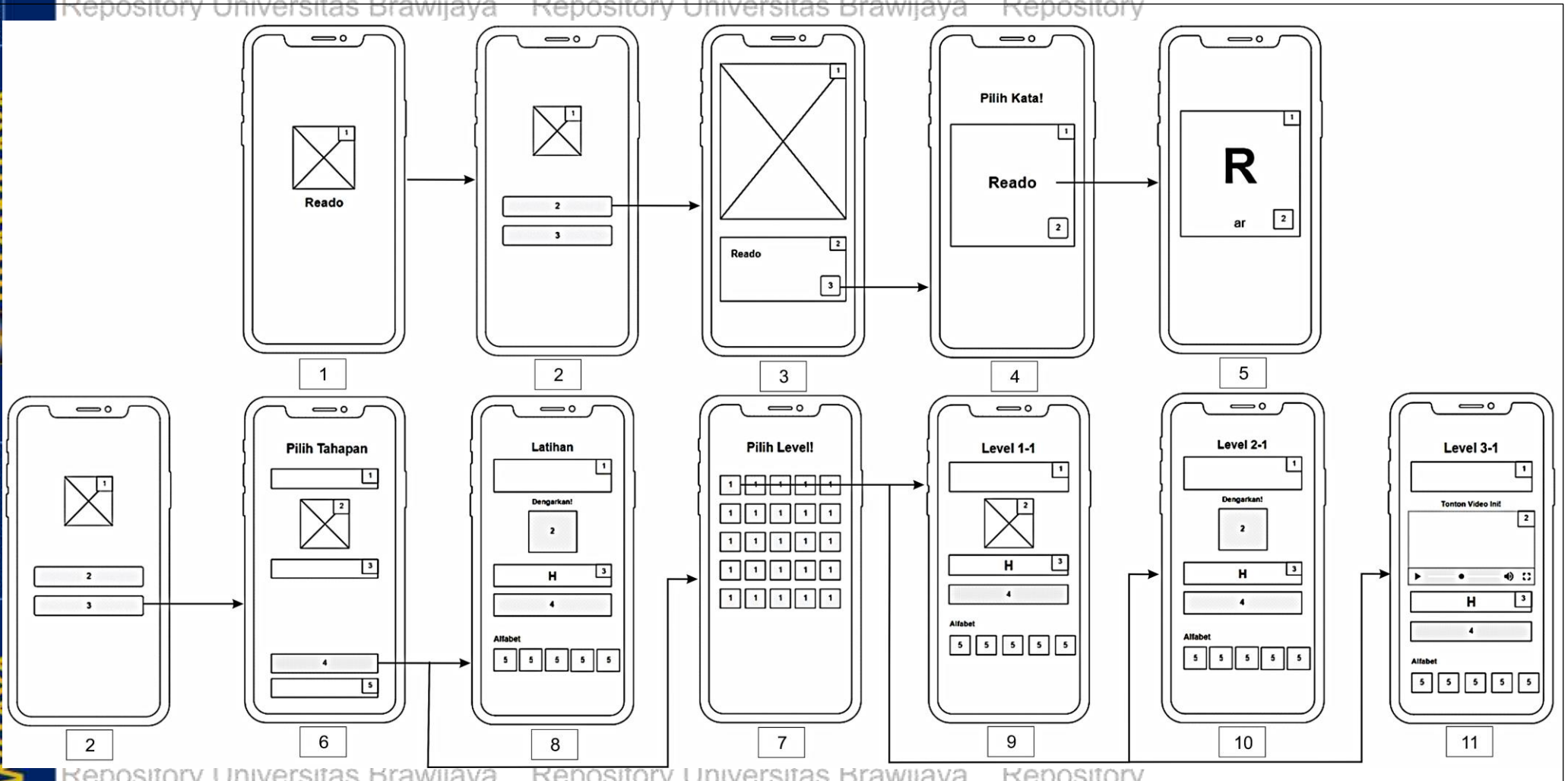


Gambar 5.24 Tampilan Aplikasi untuk Halaman Permainan Tahapan Tebak Kata Berdasarkan Video

Berdasarkan tampilan aplikasi sebelumnya yang terdiri atas 11 tampilan aplikasi yaitu tampilan aplikasi untuk logo aplikasi Readoo, tampilan aplikasi untuk



halaman utama aplikasi Reado, tampilan aplikasi untuk halaman deteksi kata, tampilan aplikasi untuk halaman kata dari hasil deteksi kata, tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata, tampilan aplikasi untuk halaman tahapan permainan, tampilan aplikasi untuk halaman *level* permainan, tampilan aplikasi untuk halaman permainan tahapan latihan, tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar, tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara dan tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video. Tampilan-tampilan dari aplikasi tersebut membuat pembuatan *storyboard* pada pengembangan aplikasi Reado ini dapat untuk ditentukan. Adapun *storyboard* dari tampilan aplikasi Reado dapat dilihat pada Gambar 5.25. Berdasarkan Gambar 5.25 yang menggambarkan alur atau *storyboard* dari tampilan aplikasi Reado memiliki alur yaitu ketika pengguna membuka aplikasi Reado, maka pengguna akan ditampilkan halaman untuk menampilkan logo dari aplikasi Reado. Selanjutnya, secara otomatis pengguna akan ditampilkan pada tampilan aplikasi untuk halaman utama yang menampilkan 2 tombol utama yaitu tombol “Ambil Kata” dan tombol “Permainan”. Ketika pengguna memilih untuk menekan tombol “Ambil Kata”, maka pengguna akan ditampilkan pada tampilan aplikasi untuk halaman deteksi kata. Setelah pengguna mendeteksi kata dan menekan tombol bahwa pengguna menyetujui hasil dari kata yang terdeteksi, maka pengguna akan ditampilkan tampilan aplikasi untuk halaman kata dari hasil deteksi kata. Lalu ketika pengguna memilih salah satu kata, maka pengguna akan ditampilkan pada tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata. Namun jika pada tampilan aplikasi untuk halaman utama yaitu pengguna memilih tombol “Permainan”, maka pengguna akan ditampilkan tampilan aplikasi untuk halaman tahapan permainan. Jika pengguna memilih permainan tahapan latihan, maka pengguna akan ditampilkan tampilan aplikasi untuk halaman permainan tahapan latihan. Namun, jika pengguna memilih permainan tahapan tebak kata berdasarkan gambar, suara ataupun video, maka pengguna akan ditampilkan tampilan aplikasi untuk halaman *level* permainan. Tampilan untuk halaman *level* permainan dapat berbeda antara tahapan tebak kata berdasarkan gambar, suara ataupun video. Selanjutnya, jika pengguna memilih permainan tahapan tebak kata berdasarkan gambar, ketika pengguna memilih salah satu *level* maka akan ditampilkan tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar dari *level* yang dipilih oleh pengguna tersebut. Jika pengguna memilih permainan tahapan tebak kata berdasarkan suara, ketika pengguna memilih salah satu *level* maka akan ditampilkan tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara dari *level* yang dipilih oleh pengguna tersebut. Namun jika pengguna memilih permainan tahapan tebak kata berdasarkan video, ketika pengguna memilih salah satu *level* maka akan ditampilkan tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video dari *level* yang dipilih oleh pengguna tersebut. Selanjutnya ketika pengguna berhasil untuk menyelesaikan permainan berdasarkan tahapan tertentu maka pengguna akan ditampilkan kembali dengan tampilan aplikasi untuk halaman permainan baik dari tahapan latihan ataupun tahapan tebak kata berdasarkan gambar, suara ataupun video.



Gambar 5.25 Storyboard Aplikasi Reado



5.1.5.5 Membuat Prototipe

Tahapan selanjutnya yaitu tim pengembang membuat prototipe berdasarkan tampilan yang sudah ditentukan dari tahapan-tahapan sebelumnya. Tahapan untuk membuat prototipe ini bertujuan untuk membuat prototipe yang tervalidasi oleh subjek penelitian yaitu pengajar dan merupakan prototipe yang *high-fidelity*. Pembuatan prototipe ini dibuat dengan menggunakan aplikasi yaitu Adobe Xd untuk mendesain prototipe yang *high-fidelity* dan interaktif. Prototipe yang *high-fidelity* dan interaktif ini bertujuan untuk memudahkan tim pengembang untuk memberikan gambaran terhadap aplikasi Reado yang akan dikembangkan pada saat melakukan pengujian dengan subjek penelitian yaitu pengajar. Adapun tampilan dari prototipe dari pengembangan aplikasi Reado, antara lain sebagai berikut:

1. Prototipe tampilan aplikasi untuk logo aplikasi Reado

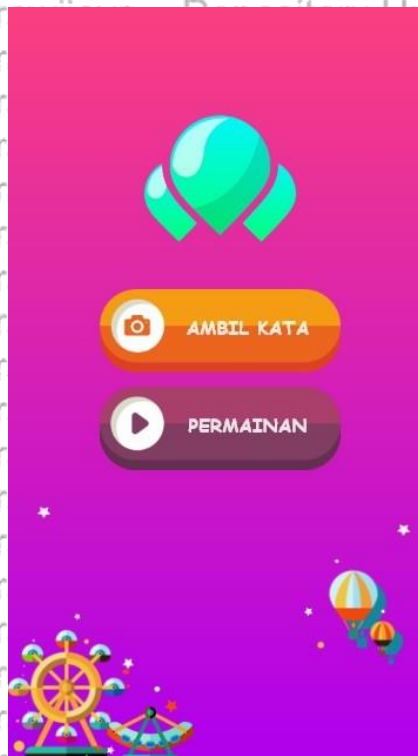
Adapun prototipe tampilan aplikasi untuk menampilkan logo dari aplikasi Reado dapat dilihat pada Gambar 5.26.



Gambar 5.26 Prototipe Tampilan Aplikasi untuk Logo Aplikasi Reado

2. Prototipe tampilan aplikasi untuk halaman utama aplikasi Reado

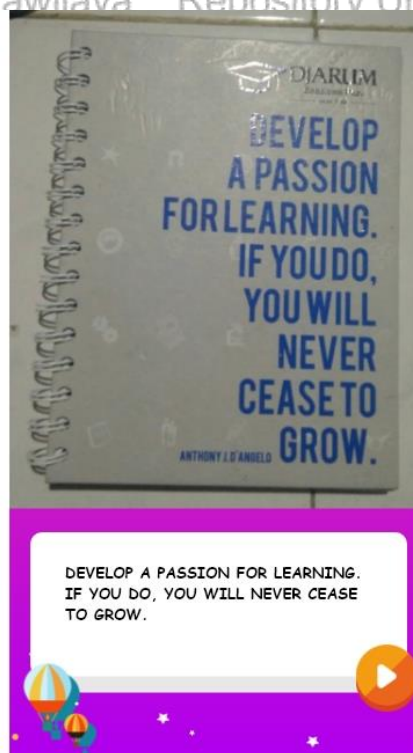
Adapun prototipe dari tampilan aplikasi untuk menampilkan halaman utama dari aplikasi Reado yang dapat dilihat pada Gambar 5.27. Pada halaman ini akan ditampilkan dua tombol dari fitur utama yaitu tombol “Ambil Kata” dan tombol “Permainan” serta menampilkan logo dari aplikasi Reado.



Gambar 5.27 Prototipe Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado

3. Prototipe tampilan aplikasi untuk halaman deteksi kata

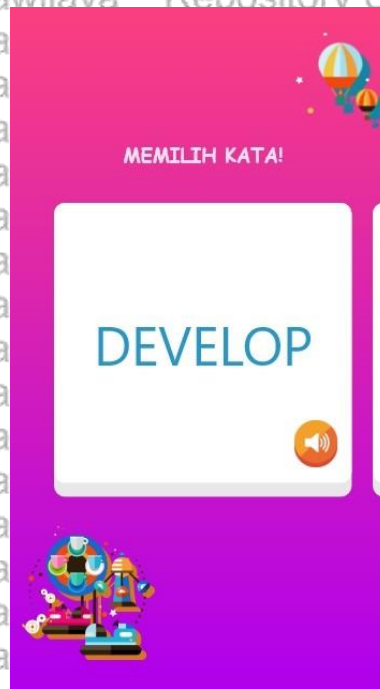
Adapun prototipe tampilan aplikasi untuk menampilkan halaman deteksi kata dari aplikasi Reado dapat dilihat pada Gambar 5.28.



Gambar 5.28 Prototipe Tampilan Aplikasi untuk Halaman Deteksi Kata

4. Prototipe tampilan aplikasi untuk halaman kata dari hasil deteksi kata

Adapun prototipe tampilan aplikasi untuk menampilkan halaman kata dari hasil deteksi kata dapat dilihat pada Gambar 5.29.



Gambar 5.29 Prototipe Tampilan Aplikasi untuk Halaman Kata

5. Prototipe tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata

Adapun prototipe tampilan aplikasi untuk menampilkan halaman alfabet dari hasil deteksi kata dapat dilihat pada Gambar 5.30.

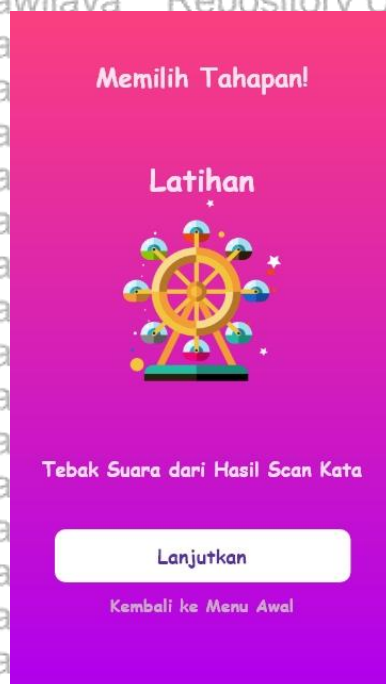


Gambar 5.30 Prototipe Tampilan Aplikasi untuk Halaman Alfabet



6. Prototipe tampilan aplikasi untuk halaman tahapan permainan

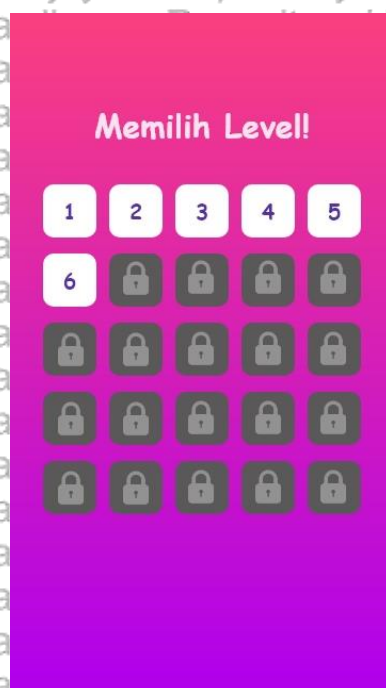
Adapun prototipe tampilan aplikasi untuk menampilkan halaman tahapan permainan dapat dilihat pada Gambar 5.31.



Gambar 5.31 Prototipe Tampilan Aplikasi untuk Halaman Tahapan Permainan

7. Prototipe tampilan aplikasi untuk halaman /level/ permainan

Adapun prototipe tampilan aplikasi untuk menampilkan halaman /level/ permainan dapat dilihat pada Gambar 5.32.



Gambar 5.32 Prototipe Tampilan Aplikasi untuk Halaman Level/ Permainan



8. Prototipe tampilan aplikasi untuk halaman permainan tahapan latihan

Adapun prototipe tampilan aplikasi untuk menampilkan halaman permainan tahapan latihan dapat dilihat pada Gambar 5.33.



Gambar 5.33 Prototipe Tampilan Aplikasi untuk Halaman Tahapan Latihan

9. Prototipe tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar

Adapun prototipe tampilan aplikasi untuk menampilkan halaman permainan tahapan tebak kata berdasarkan gambar dapat dilihat pada Gambar 5.34.



Gambar 5.34 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Gambar

10. Prototipe tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan suara

Adapun prototipe tampilan aplikasi untuk menampilkan halaman permainan tahapan tebak kata berdasarkan suara dapat dilihat pada Gambar 5.35.



Gambar 5.35 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Suara

11. Prototipe tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan video

Adapun prototipe tampilan aplikasi untuk menampilkan halaman permainan tahapan tebak kata berdasarkan video dapat dilihat pada Gambar 5.36.



Gambar 5.36 Prototipe Tampilan Aplikasi untuk Tebak Kata dari Video



5.1.5.6 Melakukan Pengujian Prototipe

Tahapan pengujian prototipe ini merupakan tahapan untuk memvalidasi prototipe yang sudah dibuat pada tahapan sebelumnya. Pengujian prototipe dilakukan dengan menggunakan teknik pengumpulan data yaitu wawancara atau *user interview*. Tahapan ini dilakukan bersama dengan subjek penelitian yaitu pengajar yang mana tahapan ini bertujuan untuk memvalidasi prototipe dari pengembangan aplikasi Reado. Berdasarkan hasil dari pengujian yang dilakukan tidak terdapat perubahan dari prototipe yang menggambarkan aplikasi Reado yang akan dikembangkan. Subjek penelitian yaitu pengajar menyetujui prototipe atau model gambaran dari aplikasi Reado yang akan diimplementasikan dikarenakan seluruh kebutuhan dari pengguna sudah terpenuhi melalui prototipe tersebut. Selanjutnya, prototipe yang sudah dibuat sebelumnya akan dijadikan referensi gambaran dari aplikasi Reado dan untuk diimplementasikan menjadi aplikasi yang dapat digunakan oleh pengguna.

5.2 Implementasi Perangkat Lunak

Dari implementasi perangkat lunak di dalam pengembangan aplikasi Reado dijelaskan beberapa tahapan yang dilakukan. Tahapan yang dilakukan dari implementasi perangkat lunak ini terdiri atas spesifikasi perangkat keras, spesifikasi perangkat lunak, batasan implementasi, implementasi basis data, implementasi algoritma, dan implementasi antarmuka.

5.2.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras dalam tahapan implementasi perangkat lunak bertujuan untuk menentukan rincian dari perangkat keras yang digunakan. Perangkat keras yang digunakan yaitu laptop dan *smartphone*. Adapun spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 5.25 untuk spesifikasi perangkat bergerak yaitu laptop dan spesifikasi perangkat bergerak yaitu *smartphone* dapat dilihat pada Tabel 5.26.

Tabel 5.25 Spesifikasi Perangkat Keras Laptop

Komponen	Spesifikasi
Model	ASUS A455LA
Prosesor	Intel i3 5005U
RAM	12GB DDR3L
GPU	Intel HD Graphics 5500
Drive	SSD 128GB

Tabel 5.26 Spesifikasi Perangkat Keras Smartphone

Komponen	Spesifikasi
Model	XIAOMI Redmi 4X



Tabel 5.26 Spesifikasi Perangkat Keras *Smartphone* (lanjutan)

Komponen	Spesifikasi
Prosesor	Snapdragon 435
RAM	4GB
Penyimpanan	64GB

5.2.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak dalam tahapan implementasi perangkat lunak bertujuan untuk menentukan rincian dari perangkat lunak yang digunakan. Perangkat lunak yang digunakan dalam pengembangan aplikasi *Reado* yaitu terbagi atas sistem operasi komputer dan *smartphone*, editor dokumen, IDE (*Integrated Development Environment*) dan bahasa pemrograman. Spesifikasi perangkat lunak yang digunakan pada tahapan implementasi perangkat lunak dapat dilihat pada Tabel 5.27.

Tabel 5.27 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi Komputer	Windows 10
Sistem Operasi <i>Smartphone</i>	Android 7 <i>Nougat</i>
Editor dokumen	Microsoft Word 2019
IDE (<i>Integrated Development Environment</i>)	Android Studio 3.4.1
Bahasa Pemrograman	Java dan XML

5.2.3 Batasan Implementasi

Batasan-batasan implementasi dari pengembangan aplikasi *Reado* pada tahapan implementasi perangkat lunak ini terdiri atas 8 batasan. Batasan implementasi bertujuan untuk memberikan gambaran dalam pengembangan aplikasi *Reado* sehingga pengembangan aplikasi ini dapat lebih terfokus kepada tujuan dari pengembangan yang sudah ditentukan. Adapun batasan-batasan implementasi ini dari pengembangan aplikasi *Reado* dalam tahapan implementasi perangkat lunak, antara lain sebagai berikut:

1. Aplikasi merupakan aplikasi perangkat bergerak berbasis *mobile*.
2. Aplikasi ini digunakan dengan menggunakan koneksi internet.
3. Aplikasi ini dikembangkan dengan penerapan konsep *object oriented*.
4. Penggunaan bahasa pemrograman dalam penelitian ini menggunakan bahasa pemrograman Java dan XML serta menggunakan IDE (*Integrated Development Environment*) yaitu Android Studio 3.4.1.



5. Aplikasi ini dapat digunakan hanya pada *platform* Android dengan versi Android minimal Jelly Bean 4.3.

5.2.4 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan perancangan basis data yang sudah dilakukan sebelumnya pada perancangan perangkat lunak. Implementasi basis data ini menggunakan sistem manajemen basis data yaitu *SQLite*. Implementasi basis data dari pengembangan aplikasi Reado terdiri atas 2 entitas yang diimplementasikan dalam bentuk tabel yaitu entitas *Quizzes* dan entitas *Stages*. Adapun algoritma dari implementasi basis data dalam pengembangan aplikasi Reado dapat dilihat pada Tabel 5.28.

Tabel 5.28 Implementasi Basis Data Aplikasi Reado

Algoritma: Membuat Tabel quizzes en	
Baris ke-	Kode Program
1.	CREATE TABLE quizzes en(
2.	" id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
3.	"stage" INTEGER,
4.	"level" INTEGER,
5.	"question" TEXT,
6.	"answere" TEXT,
7.	"media" TEXT,
8.	"time_start" INTEGER,
9.	"duartion" INTEGER
10.	);
11.	
12.	CREATE TABLE stages_en(
13.	" id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
14.	"name" TEXT,
15.	"description" TEXT,
16.	"media" TEXT
17.	);

Penjelasan dari algoritma pada Tabel 5.28 adalah sebagai berikut:

- Baris 1 sampai 10 merupakan algoritma untuk membuat tabel dari entitas *Quizzes* yang memiliki atribut-atribut yaitu *id* dengan tipe data *integer* yang merupakan *primary key* dan dengan kondisi yaitu *not null* dan *auto increment* yang mana tidak dapat memiliki nilai kosong dan akan selalu bertambah nilainya secara otomatis, *stage* dengan tipe data *integer* yaitu nilai dari tahapan permainan, *level* dengan tipe data *integer* yang merupakan nilai dari level permainan, *question* dengan tipe data *text* yaitu nilai yang menyimpan pertanyaan berupa kata yang harus ditebak, *answere* dengan tipe data *text*, *media* dengan tipe data *text*, *time_start* dengan tipe data *integer*, dan *duration* dengan tipe data *integer*.
- Baris 12 sampai 17 merupakan algoritma untuk membuat tabel dari entitas *Stages* yang memiliki atribut-atribut yaitu *id* dengan tipe data *integer* yang merupakan *primary key* dari tabel dan dengan kondisi yaitu *not null* dan *auto increment* yang mana tidak dapat memiliki nilai kosong dan akan selalu bertambah nilainya secara otomatis, *name* dengan tipe data *text*, *description* dengan tipe data *text* yang merupakan deskripsi dari setiap tahapannya dan

media dengan tipe data *text* yang merupakan atribut yang menyimpan nama dari media gambar ataupun video.

5.2.5 Implementasi Algoritma

Implementasi algoritma merupakan implementasi dalam bentuk kode program untuk mengembangkan aplikasi Reado berdasarkan perancangan algoritma yang sudah dibuat sebelumnya. Implementasi ini dilakukan dengan menggunakan bahasa pemrograman Java untuk alur logika dari aplikasi. Implementasi algoritma ini terdiri atas 10 implementasi algoritma dari setiap masing-masing kebutuhan-kebutuhan fungsional dari pengembangan aplikasi Reado. Adapun 10 implementasi algoritma dalam bahasa pemrograman Java dari pengembangan aplikasi Reado, antara lain sebagai berikut:

1. Implementasi algoritma mendeteksi kata

Tabel 5.29 Implementasi Algoritma Mendeteksi Kata

Algoritma: Mendeteksi Kata	
Baris ke-	Kode Program
1.	String word;
2.	TextView textView;
3.	SurfaceView cameraView;
4.	CameraSource cameraSource;
5.	final int RequestCameraPermissionID = 1001;
6.	Button btnStart;
7.	
8.	@Override
9.	protected void onCreate(Bundle savedInstanceState) {
10.	super.onCreate(savedInstanceState);
11.	setContentView(R.layout.activity_text_scanner);
12.	
13.	cameraView = (SurfaceView) findViewById(R.id.svScanner);
14.	textView = (TextView) findViewById(R.id.txtScanned);
15.	btnStart = (Button) findViewById(R.id.btnStart);
16.	word = "";
17.	
18.	TextRecognizer textRecognizer = new
19.	TextRecognizer.Builder(getApplicationContext()).build();
20.	if (!textRecognizer.isOperational()) {
21.	Log.w("MainActivity", "Detector dependencies belum
22.	tersedia");
23.	} else {
24.	cameraSource = new
25.	CameraSource.Builder(getApplicationContext(), textRecognizer)
26.	.setFacing(CameraSource.CAMERA_FACING_BACK)
27.	.setRequestedPreviewSize(1280, 1024)
28.	.setRequestedFps(2.0f)
29.	.setAutoFocusEnabled(true)
30.	.build();
31.	cameraView.getHolder().addCallback(new
32.	SurfaceHolder.Callback() {
33.	@Override
	public void surfaceCreated(SurfaceHolder surfaceHolder) {
	try {
	if
	(ActivityCompat.checkSelfPermission(getApplicationContext(),
	Manifest.permission.CAMERA) !=
	PackageManager.PERMISSION_GRANTED) {
	ActivityCompat.requestPermissions(TextScanner



Tabel 5.29 Implementasi Algoritma Mendeteksi Kata (lanjutan)

Baris	Kode Program
33.	Activity.this,
34.	new String[] {Manifest.permission.CAMERA},
35.	RequestCameraPermissionID);
36.	return;
37.	}
38.	cameraSource.start(cameraView.getHolder());
39.	} catch (IOException e) {
40.	e.printStackTrace();
41.	}
42.	}
43.	}
44.	@Override
45.	public void surfaceChanged(SurfaceHolder surfaceHolder, int
46.	i, int il, int i2) {
47.	}
48.	}
49.	@Override
50.	public void surfaceDestroyed(SurfaceHolder surfaceHolder) {
51.	cameraSource.stop();
52.	}
53.	};
54.	
55.	textRecognizer.setProcessor(new
56.	Detector.Processor<TextBlock>()) {
57.	@Override
58.	public void release() {
59.	}
60.	
61.	@Override
62.	public void
63.	receiveDetections(Detector.Detections<TextBlock> detections) {
64.	final SparseArray<TextBlock> items =
65.	detections.getDetectedItems();
66.	if (items.size() != 0) {
67.	textView.post(new Runnable() {
68.	@Override
69.	public void run() {
70.	StringBuilder stringBuilder = new StringBuilder();
71.	for (int i = 0; i < items.size(); ++i) {
72.	TextBlock item = items.valueAt(i);
73.	stringBuilder.append(item.getValue());
74.	stringBuilder.append(" ");
75.	word = stringBuilder.toString();
76.	textView.setText(stringBuilder.toString());
77.	}
78.	}
79.	}
80.	});
81.	}
82.	
83.	btnStart.setOnClickListener(new View.OnClickListener() {
84.	@Override
85.	public void onClick(View v) {
86.	if (word.length() > 0) {
87.	Intent toWordActivity = new
	Intent(TextScannerActivity.this, WordActivity.class);



Tabel 5.29 Implementasi Algoritma Mendeteksi Kata (lanjutan)

Baris	Kode Program
88.	toWordActivity.putExtra("word", word);
89.	startActivity(toWordActivity);
90.	}else{
91.	Toast.makeText(TextScannerActivity.this,
92.	R.string.no_text, Toast.LENGTH_SHORT).show();
93.	}
94.	});
95.	}
96.	
97.	@Override
98.	public void onRequestPermissionsResult(int requestCode, @NonNull
99.	String[] permissions, @NonNull int[] grantResults) {
100.	switch (requestCode) {
101.	case RequestCameraPermissionID: {
102.	if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
103.	if (ActivityCompat.checkSelfPermission(this,
104.	Manifest.permission.CAMERA) !=
105.	PackageManager.PERMISSION_GRANTED) {
106.	return;
107.	} try {
108.	cameraSource.start(cameraView.getHolder()
109.	) catch (IOException e) {
110.	e.printStackTrace();
111.	}
112.	break;
113.	}
114.	}
115.	}

Penjelasan dari algoritma pada Tabel 5.29 adalah sebagai berikut:

Baris 1 sampai 6 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *word*, *textView*, *cameraView*, *cameraSource*, *RequestCameraPermissionID* dan *btnStart*.

Baris 8 sampai 95 merupakan fungsi untuk memuat halaman dari halaman deteksi kata. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk menyalakan kamera, untuk mendeteksi kata, untuk menerima kata hasil deteksi, untuk memberikan perintah ketika tombol ditekan dan untuk menampilkan kata yang terdeteksi pada halaman.

Baris 97 sampai 115 merupakan fungsi untuk meminta permintaan untuk izin penggunaan kamera pada aplikasi ReadO dan memberikan izin untuk penggunaan kamera. Fungsi ini menggunakan nilai dari variabel *RequestCameraPermissionID* yaitu 1001 yang mana jika kode permintaan sama dengan nilai variabel *RequestCameraPermissionID* yang sudah didefinisikan sebelumnya, maka kamera akan dinyalakan. Kamera ini akan digunakan untuk mendeteksi kata dari aplikasi ReadO.



2. Implementasi algoritma menampilkan kata-kata dari hasil deteksi kata

Tabel 5.30 Implementasi Algoritma Menampilkan Kata

Algoritma: Menampilkan Kata dari Hasil Deteksi Kata	
Baris ke-	Kode Program
1.	public class WordActivity extends AppCompatActivity {
2.	
3.	String word;
4.	ArrayList<String> words;
5.	RecyclerView rvWords;
6.	QuizzesHandler quizzesHandler;
7.	
8.	@Override
9.	protected void onCreate(Bundle savedInstanceState) {
10.	super.onCreate(savedInstanceState);
11.	setContentView(R.layout.activity_word);
12.	quizzesHandler = new QuizzesHandler(this);
13.	Bundle extras = getIntent().getExtras();
14.	if (extras != null) {
15.	word = extras.getString("word").toUpperCase();
16.	words = new ArrayList<>(Arrays.asList(word.split("\\s+")));
17.	for (String word: words) {
18.	Quiz quiz = new Quiz(0, 0, 0, word, word, null, 0, 0);
19.	quizzesHandler.addQuiz(quiz);
20.	}
21.	
22.	rvWords = (RecyclerView) findViewById(R.id.rv_words);
23.	rvWords.setAdapter(new WordAdapter(words));
24.	rvWords.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.HORIZONTAL, false));
25.	}
26.	
27.	
28.	public class WordAdapter extends
29.	RecyclerView.Adapter<WordAdapter.WordViewHolder> {
30.	Context context;
31.	TextToSpeech tts;
32.	
33.	@NonNull
34.	@Override
35.	public WordViewHolder onCreateViewHolder(@NonNull ViewGroup
36.	parent, int i) {
37.	context = parent.getContext();
38.	View itemView =
39.	LayoutInflater.from(context).inflate(R.layout.item_word_card,
40.	parent, false);
41.	itemView.setLayoutParams().width = (int)
42.	(parent.getMeasuredHeight() * 1);
43.	itemView.requestLayout();
44.	tts = new TextToSpeech(context, new
45.	TextToSpeech.OnInitListener() {
46.	@Override
47.	public void onInit(int status) {
48.	if (status != TextToSpeech.ERROR) {
49.	tts.setLanguage(Locale.getDefault());
50.	}
51.	});
	return new WordViewHolder(itemView);
	}
	@Override



Tabel 5.30 Implementasi Algoritma Menampilkan Kata (lanjutan)

Algoritma: Menampilkan Kata dari Hasil Deteksi Kata	
Baris ke-	Kode Program
52.	<code>public void onBindViewHolder(@NonNull ViewHolder viewHolder,</code>
53.	<code>int i) {</code>
54.	<code>final String word = words.get(i);</code>
55.	<code>viewHolder.word.setText(word);</code>
56.	<code>viewHolder.word.setOnClickListener(new View.OnClickListener() {</code>
57.	<code>@Override</code>
58.	<code>public void onClick(View v) {</code>
59.	<code>Intent toAlphabetActivity = new Intent(context,</code>
60.	<code>AlphabetActivity.class);</code>
61.	<code>toAlphabetActivity.putExtra("word", word.trim());</code>
62.	<code>context.startActivity(toAlphabetActivity);</code>
63.	<code>}}</code>
64.	<code>});</code>
65.	<code>viewHolder.btnSound.setOnClickListener(new</code>
66.	<code>View.OnClickListener() {</code>
67.	<code>@Override</code>
68.	<code>public void onClick(View v) {</code>
69.	<code>tts.speak(word, TextToSpeech.QUEUE_FLUSH, null, "reado-" +</code>
70.	<code>word);</code>
71.	<code>}}</code>
72.	<code>}</code>
73.	<code>class ViewHolder extends RecyclerView.ViewHolder {</code>
74.	<code>public TextView word;</code>
75.	<code>public Button btnSound;</code>
76.	<code>public ViewHolder(@NonNull View itemView) {</code>
77.	<code>super(itemView);</code>
78.	<code>this.word = (TextView) itemView.findViewById(R.id.txtWord);</code>
79.	<code>this.btnSound = (Button)</code>
80.	<code>itemView.findViewById(R.id.btnSound);</code>
81.	<code>}</code>
82.	<code>}</code>
83.	<code>}</code>
84.	<code>}</code>

Penjelasan dari algoritma pada Tabel 5.30 adalah sebagai berikut:

- Baris 3 sampai 6 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *word*, *words*, *rvWords* dan *quizzesHandler*.
- Baris 8 sampai 25 merupakan fungsi untuk memuat halaman dari halaman kata. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk membagi kata dengan menggunakan fungsi *split()*, untuk mendapatkan kata dari halaman sebelumnya dan untuk menambahkan permainan ke dalam basis data.
- Baris 30 dan 31 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *context* dan *tts*.
- Baris 33 sampai 49 merupakan fungsi untuk mendefinisikan nilai dari variabel *context* dan *tts*. Variabel *context* didefinisikan berdasarkan tampilan dari



halaman kata sedangkan variabel *tts* didefinisikan dengan menggunakan fungsi dari *Text To Speech* untuk mengatur bahasa yang digunakan yaitu menggunakan fungsi *setLanguage()*

- Baris 51 sampai 71 merupakan fungsi untuk memberikan perintah ketika kata dipilih oleh pengguna, untuk memberikan perintah ketika tombol suara ditekan yaitu dengan memanggil fungsi *speak()* dan untuk mengirimkan kata ke halaman selanjutnya dengan menggunakan fungsi *putExtra()* dari kelas *Intent*.

- Baris 73 sampai 83 merupakan fungsi untuk mendeklarasi dan mendefinisikan variabel yang terdiri atas variabel *word* dan *btnSound*.

3. Implementasi algoritma menampilkan alfabet dari kata berdasarkan hasil deteksi kata

Tabel 5.31 Implementasi Algoritma Menampilkan Alfabet

Algoritma: Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	
Baris	Kode Program
1.	<code>public class AlphabetActivity extends AppCompatActivity {</code>
2.	
3.	<code>String word;</code>
4.	<code>ArrayList<String> alphabets;</code>
5.	<code>RecyclerView rvAlphabet;</code>
6.	<code>QuizzesHandler quizzesHandler;</code>
7.	
8.	<code>@Override</code>
9.	<code>protected void onCreate(Bundle savedInstanceState) {</code>
10.	<code>super.onCreate(savedInstanceState);</code>
11.	<code>setContentView(R.layout.activity_alphabet);</code>
12.	
13.	<code>alphabets = new ArrayList<>();</code>
14.	<code>Bundle extras = getIntent().getExtras();</code>
15.	<code>if (extras != null) {</code>
16.	<code>word = extras.getString("word");</code>
17.	<code>for(char alphabet : word.toCharArray()) {</code>
18.	<code>alphabets.add(alphabet);</code>
19.	<code>}</code>
20.	<code>}</code>
21.	<code>rvAlphabet = (RecyclerView) findViewById(R.id.rv_alphabets);</code>
22.	<code>rvAlphabet.setAdapter(new AlphabetAdapter(alphabets));</code>
23.	<code>rvAlphabet.setLayoutManager(new LinearLayoutManager(this,</code>
24.	<code>LinearLayoutManager.HORIZONTAL, false));</code>
25.	<code>}</code>
26.	
27.	<code>public class AlpgabetAdapter extends</code>
28.	<code>RecyclerView.Adapter<AlphabetAdapter.WordViewHolder> {</code>
29.	<code>Context context;</code>
30.	<code>TextToSpeech tts;</code>
31.	<code>String[] alphabetPronounce;</code>
32.	<code>String[] numericPronounce;</code>
33.	
34.	<code>@NonNull</code>
35.	<code>@Override</code>
36.	<code>public AlphabetViewHolder onCreateViewHolder(@NonNull ViewGroup</code>
37.	<code>parent, int i) {</code>
	<code>context = parent.getContext();</code>



Tabel 5.31 Implementasi Algoritma Menampilkan Alfabet (lanjutan)

Algoritma: Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	
Baris	Kode Program
ke-	
38.	alphabetPronounce =
	context.getResources().getStringArray(R.array.alphabets);
39.	numericPronounce =
	context.getResources().getStringArray(R.array.numerics);
40.	
41.	View itemView =
	LayoutInflater.from(context).inflate(R.layout.item_alphabet_card,
	parent, false);
42.	itemView.getLayoutParams().width = (int)
	(parent.getMeasuredHeight() * 1);
43.	itemView.requestLayout();
44.	
45.	tts = new TextToSpeech(context, new
	TextToSpeech.OnInitListener() {
46.	@Override
47.	public void onInit(int status) {
48.	if (status != TextToSpeech.ERROR) {
49.	tts.setLanguage(Locale.getDefault());
50.	}
51.	});
52.	});
53.	return new AlphabetViewHolder(itemView);
54.	}
55.	
56.	@Override
57.	public void onBindViewHolder(@NonNull AlphabetViewHolder
	alphanbetViewHolder, int i) {
58.	final char alphabet = alphabets.get(i);
59.	String tempPronounce = "" + alphabets.get(i);
60.	if((int)alphabet >= 65 && (int)alphabet <= 90){
61.	tempPronounce = this.alphabetPronounce[(int)alphabet - 65];
62.	}
63.	else if ((int)alphabet >= 48 && (int)alphabet <= 57){
64.	tempPronounce = this.numericPronounce[(int)alphabet - 48];
65.	}
66.	final String pronounce = tempPronounce;
67.	alphanbetViewHolder.txtAlphabet.setText("" + alphabet);
68.	alphanbetViewHolder.txtRead.setText(pronounce);
69.	alphanbetViewHolder.btnSound.setOnClickListener(new
	View.OnClickListener() {
70.	@Override
71.	public void onClick(View v) {
72.	tts.speak("" + alphabet, TextToSpeech.QUEUE_FLUSH, null,
	"read!" + alphabet);
73.	}
74.	});
75.	}
76.	}
77.	class AlphabetViewHolder extends RecyclerView.ViewHolder{
78.	public TextView txtAlphabet, txtRead;
79.	public Button btnSound;
80.	public AlphabetViewHolder(@NonNull View itemView) {
81.	super(itemView);
82.	txtAlphabet = (TextView)
	itemView.findViewById(R.id.txtAlphabet);
83.	txtRead = (TextView) itemView.findViewById(R.id.txtRead);
84.	btnSound = (Button) itemView.findViewById(R.id.btnSound);
85.	}
86.	}
87.	}



Penjelasan dari algoritma pada Tabel 5.31 adalah sebagai berikut:

- Baris 3 sampai 6 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *word*, *alphabets*, *rvAlphabet* dan *quizzesHandler*.
- Baris 8 sampai 24 merupakan fungsi untuk memuat halaman dari halaman alfabet. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk mendapatkan kata, untuk membagi kata menjadi alfabet dengan menggunakan fungsi *toCharArray()* dan untuk mendapatkan kata dari halaman sebelumnya.
- Baris 29 sampai 32 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *context*, *tts*, *alphabetPronounce*, dan *numericPronounce*.
- Baris 34 sampai 54 merupakan fungsi untuk mendefinisikan nilai dari variabel *context*, *tts*, *alphabetPronounce*, dan *numericPronounce*. Variabel *tts* didefinisikan dengan menggunakan fungsi dari *Text To Speech* untuk mengatur bahasa yang digunakan yaitu menggunakan fungsi *setLanguage()*.
- Baris 56 sampai 75 merupakan fungsi untuk mendefinisikan nilai dari variabel *alphabetPronounce* dan *numericPronounce* berdasarkan cara pembacaan dari alfabet yang dipilih, untuk menampilkan cara pembacaan dan alfabet, untuk memberikan perintah ketika alfabet dipilih oleh pengguna dan untuk memberikan perintah ketika tombol suara ditekan yaitu dengan memanggil fungsi *speak()*.
- Baris 77 sampai 86 merupakan fungsi untuk mendeklarasi dan mendefinisikan variabel yang terdiri atas variabel *txtAlphabet*, *txtRead* dan *btnSound*.

4. Implementasi algoritma menentukan tahapan permainan

Tabel 5.32 Implementasi Algoritma Menentukan Tahapan Permainan

Algoritma: Menentukan Tahapan Permainan	
Baris key	Kode Program
1.	<code>public class StageActivity extends AppCompatActivity implements</code>
2.	<code>DiscreteScrollView.ScrollStateChangeListener<StageAdapter.StageViewHolder>,</code>
3.	<code>DiscreteScrollView.OnItemChangeListener<StageAdapter.StageViewHolder> {</code>
4.	<code>ArrayList<Stage> stages;</code>
5.	<code>DiscreteScrollView rvStage;</code>
6.	<code>int stageCount;</code>
7.	<code>int stageZeroQuizzesCount, currentStage;</code>
8.	<code>Button btnContinue;</code>
9.	<code>TextView txtBackHome;</code>
10.	<code>StagesHandler stagesHandler;</code>
11.	<code>QuizzesHandler quizzesHandler;</code>
12.	<code>@Override</code>
13.	<code>protected void onCreate(Bundle savedInstanceState) {</code>
14.	<code>super.onCreate(savedInstanceState);</code>
15.	<code>setContentViews(R.layout.activity_stage);</code>
16.	<code>txtBackHome = (TextView) findViewById(R.id.btnBackHome);</code>
17.	<code>btnContinue = (Button) findViewById(R.id.btnContinue);</code>
18.	<code>}</code>



Tabel 5.32 Implementasi Algoritma Menentukan Tahapan Permainan (lanjutan)

Algoritma: Menentukan Tahapan Permainan	
Baris ke-	Kode Program
19.	rvStage = (DiscreteScrollView) findViewById(R.id.rvStage);
20.	stages = stagesHandler.getAllStages();
21.	stageCount = stages.size() - 1;
22.	currentStage = mPreferences.getInt("currentStage", 1);
23.	stageZeroQuizzesCount = quizzesHandler.getQuizzesCountByStage(0);
24.	stagesHandler = new StagesHandler(this);
25.	quizzesHandler = new QuizzesHandler(this);
26.	
27.	txtBackHome.setOnClickListener(new View.OnClickListener() {
28.	@Override
29.	public void onClick(View v) {
30.	finish();
31.	}
32.	});
33.	btnContinue.setOnClickListener(new View.OnClickListener() {
34.	@Override
35.	public void onClick(View v) {
36.	Intent intent = new Intent(StageActivity.this, LevelActivity.class);
37.	switch (rvStage.getCurrentItem()) {
38.	case 0:
39.	if (stageZeroQuizzesCount == 0) {
40.	Toast.makeText(getBaseContext(), "Tidak ada kata yang diambil tersimpan", Toast.LENGTH_SHORT).show();
41.	return;
42.	} else if (stageZeroQuizzesCount > 10) {
43.	stageZeroQuizzesCount = 10;
44.	}
45.	ArrayList<Quiz> quizzes = new ArrayList<Quiz>();
46.	quizzesHandler.getRandomQuizzes(0, stageZeroQuizzesCount);
47.	};
48.	Intent toQuizStageZeroActivity = new Intent(StageActivity.this, QuizStageZeroActivity.class);
49.	toQuizStageZeroActivity.putExtra("quizzes", quizzes);
50.	startActivity(toQuizStageZeroActivity);
51.	break;
52.	case 1:
53.	intent.putExtra("stage", 1);
54.	break;
55.	case 2:
56.	intent.putExtra("stage", 2);
57.	break;
58.	case 3:
59.	intent.putExtra("stage", 3);
60.	break;
61.	default:
62.	intent.putExtra("stage", 1);
63.	}
64.	if (rvStage.getCurrentItem() <= currentStage) {
65.	startActivity(intent);
66.	}
67.	}
68.	});
69.	rvStage.setAdapter(new StageAdapter(this, stages));
70.	rvStage.setItemTransformer(new ScaleTransformer.Builder()



Tabel 5.32 Implementasi Algoritma Menentukan Tahapan Permainan (lanjutan)

Algoritma: Menentukan Tahapan Permainan	
Baris ke-	Kode Program
71.	.setMaxScale(1.05f)
72.	.setMinScale(0.8f)
73.	.setPivotX(Pivot.X.CENTER) // CENTER is a default one
74.	.setPivotY(Pivot.Y.BOTTOM) // CENTER is a default one
75.	.build());
76.	rvStage.addOnItemChangeListener(this);
77.	rvStage.addScrollStateChangeListener(this);
78.	if(currentStage <= stageCount) {
79.	rvStage.scrollToPosition(currentStage);
80.	}
81.	}
82.	}
83.	
84.	public class StageAdapter extends RecyclerView.Adapter<StageAdapter.WordViewHolder> {
85.	
86.	Context context;
87.	int currentStage;
88.	
89.	public StageAdapter(Context context, ArrayList<Stage> stages) {
90.	super();
91.	currentStage = mPreferences.getInt("currentStage", 1);
92.	}
93.	
94.	@NonNull
95.	@Override
96.	public StageViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
97.	context = parent.getContext();
98.	View itemView = LayoutInflater.from(context).inflate(R.layout.item_alphabet_card, parent, false);
99.	itemView.getLayoutParams().width = (int) (parent.getMeasuredHeight() * 1);
100.	itemView.requestLayout();
101.	return new StageViewHolder(itemView);
102.	}
103.	
104.	@Override
105.	public void onBindViewHolder(@NonNull StageViewHolder stageViewHolder, int i) {
106.	final Stage stage = stages.get(i);
107.	
108.	stageViewHolder.imgStage.getLayoutParams().height = stageViewHolder.imgStage.getLayoutParams().width;
109.	
110.	Drawable image = AssetReader.getDrawableFromAssets(context, stage.getMedia());
111.	
112.	final int stageIndex = stage.getId() - 1;
113.	
114.	if(stageIndex == currentStage){
115.	stageViewHolder.txtDescription.setVisibility(View.VISIBLE);
116.	}
117.	if(stageIndex <= currentStage){
118.	stageViewHolder.vLock.setVisibility(View.GONE);
119.	stageViewHolder.imgLock.setVisibility(View.GONE);



Tabel 5.32 Implementasi Algoritma Menentukan Tahapan Permainan (lanjutan)

Algoritma: Menentukan Tahapan Permainan	
Baris ke-	Kode Program
120.	stageViewHolder.itemStage.setOnClickListener(new
121.	View.OnClickListener() {
122.	@Override
123.	public void onClick(View v) {
124.	if (stageIndex == 0) {
125.	int stageZeroQuizzesCount =
126.	quizzesHandler.getQuizzesCountByStage(0);
127.	Intent toQuizStageZeroActivity = new Intent(context,
128.	QuizStageZeroActivity.class);
129.	if (stageZeroQuizzesCount == 0) {
130.	Toast.makeText(context, "Tidak ada kata yang diambil",
131.	Toast.LENGTH_SHORT).show();
132.	return;
133.	} else if (stageZeroQuizzesCount > 10) {
134.	stageZeroQuizzesCount = 10;
135.	} else {
136.	ArrayList<Quiz> quizzes = new
137.	ArrayList<Quiz>();
138.	quizzesHandler.getRandomQuizzes(0,
139.	stageZeroQuizzesCount);
140.	toQuizStageZeroActivity.putExtra("quizzes", quizzes);
141.	context.startActivity(toQuizStageZeroActivity);
142.	} else {
143.	Intent toLevelActivity = new Intent(context,
144.	LevelActivity.class);
145.	toLevelActivity.putExtra("stage", stageIndex);
146.	context.startActivity(toLevelActivity);
147.	}
148.	} else {
149.	return;
150.	}
151.	}
152.	};
153.	}
154.	
155.	public class StageViewHolder extends RecyclerView.ViewHolder {
156.	TextView txtTitle, txtDescription;
157.	ImageView imgStage, imgLock;
158.	
159.	public StageViewHolder(@NonNull View itemView) {
160.	super(itemView);
161.	txtTitle = (TextView) itemView.findViewById(R.id.txtTitle);
162.	imgStage = (ImageView) itemView.findViewById(R.id.imgStage);
163.	txtDescription = (TextView)
164.	itemView.findViewById(R.id.txtDescription);
165.	imgLock = (ImageView) itemView.findViewById(R.id.imgLock);
166.	}
167.	public void showText() {
168.	TransitionManager.beginDelayedTransition(itemStage, set);
169.	txtDescription.setVisibility(View.VISIBLE);
170.	}
171.	public void hideText() {
172.	TransitionManager.beginDelayedTransition(itemStage, set);
173.	txtDescription.setVisibility(View.INVISIBLE);
174.	}
175.	}



Penjelasan dari algoritma pada Tabel 5.32 adalah sebagai berikut:

- Baris 3 sampai 10 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *stages*, *rvStage*, *stageCount*, *stageZeroQuizzesCount*, *currentStage*, *btnContinue*, *txtBackHome*, *stagesHandler* dan *quizzesHandler*.
- Baris 12 sampai 81 merupakan fungsi untuk memuat halaman dari halaman tahapan permainan. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk memberikan perintah ketika tombol dari *txtBackHome* dan *btnContinue* ditekan dan menampilkan tahapan-tahapan.
- Baris 86 dan 87 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *context* dan *currentStage*.
- Baris 89 sampai 92 merupakan fungsi untuk mendefinisikan nilai dari variabel *currentStage* yang didapatkan dari basis data.
- Baris 94 sampai 102 merupakan fungsi untuk mendefinisikan nilai dari variabel *context*.
- Baris 104 sampai 145 merupakan fungsi untuk mendapatkan tahapan-tahapan dari basis data, untuk menentukan tahapan yang dipilih oleh pengguna, untuk memberikan perintah ketika tahapan dipilih atau ditekan oleh pengguna, untuk memberikan perintah ketika pengguna memilih tahapan latihan maka akan membuka halaman tahapan latihan dan memberikan perintah ketika pengguna memilih tahapan tebak kata, maka akan membuka halaman *level* permainan.
- Baris 148 dan 149 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *txtTitle*, *txtDescription*, *imgStage* dan *imgLock*.
- Baris 151 sampai 157 merupakan fungsi untuk mendefinisikan nilai dari variabel *txtTitle*, *imgStage*, *txtDescription* dan *imgLock*.
- Baris 159 sampai 162 merupakan fungsi untuk menampilkan teks.
- Baris 164 sampai 167 merupakan fungsi untuk menyembunyikan teks.

5. Implementasi algoritma menentukan *level* permainan

Tabel 5.33 Implementasi Algoritma Menentukan *Level* Permainan

Algoritma: Menentukan <i>Level</i> Permainan	
Baris ke-	Kode Program
1.	<code>public class StageActivity extends AppCompatActivity {</code>
2.	
3.	<code> ArrayList<Quiz> quizzes;</code>
4.	<code> RecyclerView rvLevel;</code>
5.	<code> int stage;</code>
6.	
7.	<code> @Override</code>
8.	<code> protected void onCreate(Bundle savedInstanceState) {</code>
9.	<code> super.onCreate(savedInstanceState);</code>
10.	<code> setContentview(R.layout.activity_level);</code>
11.	<code> Bundle extras = getIntent().getExtras();</code>



Tabel 5.33 Implementasi Algoritma Menentukan Level Permainan (lanjutan)

Algoritma: Menentukan Level Permainan	
Baris	Kode Program
12.	if (extras != null) {
13.	stage = extras.getInt("stage");
14.	}
15.	
16.	QuizzesHandler quizzesHandler = new QuizzesHandler(this);
17.	quizzes = quizzesHandler.getQuizzesByStage(stage);
18.	rvLevel = (RecyclerView) findViewById(R.id.rvLevel);
19.	rvLevel.setAdapter(new LevelAdapter(this, stage, quizzes));
20.	rvLevel.setLayoutManager(new GridLayoutManager(this, 5));
21.	}
22.	}
23.	
24.	public class LevelAdapter extends
25.	RecyclerView.Adapter<LevelAdapter.LevelViewHolder> {
26.	Context context;
27.	int stage, currentStage, currentLevel;
28.	
29.	public LevelAdapter(Context context, int stage, ArrayList<Quiz>
30.	levels) {
31.	super();
32.	this.context = context;
33.	this.stage = stage;
34.	currentStage = mSharedPreferences.getInt("currentStage", 1);
35.	currentLevel = mSharedPreferences.getInt("currentLevel", 1);
36.	}
37.	@Override
38.	public void onBindViewHolder(@NonNull LevelViewHolder viewHolder,
39.	int i) {
40.	viewHolder.txtLevel.setText("" + quiz.getLevel());
41.	final Intent toQuizActivity;
42.	switch (quiz.getStage()) {
43.	case 1:
44.	toQuizActivity = new Intent(context,
45.	QuizStageOneActivity.class);
46.	break;
47.	case 2:
48.	toQuizActivity = new Intent(context,
49.	QuizStageTwoActivity.class);
50.	break;
51.	case 3:
52.	toQuizActivity = new Intent(context,
53.	QuizStageThreeActivity.class);
54.	break;
55.	default:
56.	toQuizActivity = new Intent(context, StageActivity.class);
57.	break;
58.	if (quiz.getStage() <= currentStage) {
59.	viewHolder.txtLevel.setTextColor(Color.GREEN);
60.	viewHolder.vLock.setVisibility(View.GONE);
61.	viewHolder.imgLock.setVisibility(View.GONE);
62.	viewHolder.txtLevel.setOnClickListener(new
63.	View.OnClickListener() {
64.	@Override
	public void onClick(View v) {



Tabel 5.33 Implementasi Algoritma Menentukan Level Permainan (lanjutan)

Baris	Kode Program
ke-	
65.	context.startActivity(toQuizActivity);
66.	}
67.	});
68.	}else if(quiz.getStage() == currentStage && quiz.getLevel() <= currentLevel){
69.	if(quiz.getLevel() < currentLevel){
70.	viewHolder.txtLevel.setTextColor(Color.GREEN);
71.	}
72.	viewHolder.vLock.setVisibility(View.GONE);
73.	viewHolder.imgLock.setVisibility(View.GONE);
74.	viewHolder.txtLevel.setOnClickListener(new
75.	View.OnClickListener(){
76.	@Override
77.	public void onClick(View v) {
78.	context.startActivity(toQuizActivity);
79.	}
80.	});
81.	}
82.	}
83.	
84.	public class LevelViewHolder extends RecyclerView.ViewHolder {
85.	
86.	public TextView txtLevel;
87.	public View vLock;
88.	public ImageView imgLock;
89.	
90.	public LevelViewHolder(@NonNull View itemView) {
91.	super(itemView);
92.	this.txtLevel = (TextView)
93.	itemView.findViewById(R.id.txtLevel);
94.	this.vLock = (View) itemView.findViewById(R.id.vLock);
95.	this.imgLock = (ImageView)
96.	itemView.findViewById(R.id.imgLock);
97.	}

Penjelasan dari algoritma pada Tabel 5.33 adalah sebagai berikut:

- Baris 3 sampai 5 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *quizzes*, *rvLevel* dan *stage*.
- Baris 7 sampai 22 merupakan fungsi untuk memuat halaman dari halaman *level* permainan. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk mendapatkan nilai dari variabel *stage* berdasarkan halaman sebelumnya dan untuk mendapatkan data permainan dari basis data.
- Baris 26 dan 27 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *context*, *stage*, *currentStage* dan *currentLevel*.
- Baris 29 sampai 35 merupakan fungsi untuk mendefinisikan nilai dari variabel *context*, *stage*, *currentStage* dan *currentLevel*.



- Baris 37 sampai 82 merupakan fungsi untuk menampilkan *level* yang didapatkan dari basis data, untuk menentukan tahapan yang dipilih oleh pengguna berdasarkan halaman sebelumnya, untuk mengirimkan nilai dari variabel *stage* dan *level* terhadap halaman selanjutnya dan untuk memberikan perintah ketika *level* ditekan maka akan membuka halaman selanjutnya.

- Baris 86 sampai 88 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *txtLevel*, *vLock* dan *imgLock*.

- Baris 90 sampai 96 merupakan fungsi untuk mendefinisikan variabel yang terdiri atas variabel *txtLevel*, *vLock* dan *imgLock*.

6. Implementasi algoritma menampilkan permainan tahapan latihan

Tabel 5.34 Implementasi Algoritma Menampilkan Permainan Tahapan Latihan

Algoritma: Menampilkan Permainan Tahapan Latihan	
Baris ke-	Kode Program
1.	<code>private TextView textScreen, textQuestion, textTitle;</code>
2.	<code>private Button btnSound;</code>
3.	<code>private StagesHandler stagesHandler;</code>
4.	<code>private QuizzesHandler quizzesHandler;</code>
5.	<code>private ArrayList<Quiz> quizzes;</code>
6.	<code>private Quiz quiz;</code>
7.	<code>private TextToSpeech tts;</code>
8.	
9.	<code>@Override</code>
10.	<code>protected void onCreate(Bundle savedInstanceState) {</code>
11.	<code>super.onCreate(savedInstanceState);</code>
12.	<code>setContentView(R.layout.activity_quiz_stage_zero);</code>
13.	
14.	<code>textQuestion = (TextView) findViewById(R.id.textQuestion);</code>
15.	<code>textScreen = (TextView) findViewById(R.id.textScreen);</code>
16.	<code>textTitle = (TextView) findViewById(R.id.textTitle);</code>
17.	<code>btnSound = (Button) findViewById(R.id.btnSound);</code>
18.	
19.	<code>Bundle bundle = getIntent().getExtras();</code>
20.	<code>if (bundle != null) {</code>
21.	<code>if (bundle.containsKey("quizzes")) {</code>
22.	<code>ArrayListWrapper<Quiz> quizzes = (ArrayListWrapper<Quiz>)</code>
23.	<code>bundle.getSerializable("quizzes");</code>
24.	<code>this.quizzes = quizzes.getData();</code>
25.	<code>this.quiz = this.quizzes.get(0);</code>
26.	<code>this.quizzes.remove(0);</code>
27.	<code>else {</code>
28.	<code>finish();</code>
29.	<code>}</code>
30.	
31.	<code>tts = new TextToSpeech(QuizStageZeroActivity.this, new</code>
32.	<code>TextToSpeech.OnInitListener() {</code>
33.	<code>@Override</code>
34.	<code>public void onInit(int status) {</code>
35.	<code>if (status != TextToSpeech.ERROR) {</code>
36.	<code>tts.setLanguage(Locale.getDefault());</code>
37.	<code>}</code>
38.	<code>});</code>
39.	



Tabel 5.34 Implementasi Algoritma Menampilkan Permainan Tahapan Latihan (lanjutan)

Algoritma: Menampilkan Permainan Tahapan Latihan	
Baris ke-	Kode Program
40.	<code>btnSound.setOnClickListener(new View.OnClickListener() {</code>
41.	<code> @Override</code>
42.	<code> public void onClick(View v) {</code>
43.	<code> tts.speak(quiz.getAnswer(), TextToSpeech.QUEUE_FLUSH, null,</code>
44.	<code> "reado!-" + quiz.getAnswer());</code>
45.	<code> }.</code>
46.	<code>}</code>

Penjelasan dari algoritma pada Tabel 5.34 adalah sebagai berikut:

- Baris 3 sampai 5 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *textScreen*, *textQuestion*, *textTitle*, *btnSound*, *stagesHandler*, *quizzes*, *quiz* dan *tts*.

Baris 9 sampai 46 merupakan fungsi untuk memuat halaman. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan variabel, mendapatkan data permainan dari basis data, mendefinisikan nilai dari variabel *tts* yaitu dengan menggunakan fungsi dari *Text To Speech* untuk mengatur bahasa yang digunakan yaitu menggunakan fungsi *setLanguage()* dan memberikan perintah ketika tombol dari *btnSound* ditekan yaitu dengan menggunakan fungsi *speak()* untuk menyuarakan cara pembacaan dari kata yang harus ditebak.

- 7. Implementasi algoritma menampilkan permainan tahapan tebak kata berdasarkan gambar

Tabel 5.35 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	
Baris ke-	Kode Program
1.	<code>private TextView textScreen, textQuestion, textTitle;</code>
2.	<code>private ImageView imgGuess;</code>
3.	<code>private StagesHandler stagesHandler;</code>
4.	<code>private QuizzesHandler quizzesHandler;</code>
5.	<code>private Quiz quiz;</code>
6.	
7.	<code>@Override</code>
8.	<code>protected void onCreate(Bundle savedInstanceState) {</code>
9.	<code> super.onCreate(savedInstanceState);</code>
10.	<code> setContentView(R.layout.activity_quiz_stage_one);</code>
11.	
12.	<code> textQuestion = (TextView) findViewById(R.id.textQuestion);</code>
13.	<code> textScreen = (TextView) findViewById(R.id.textScreen);</code>
14.	<code> textTitle = (TextView) findViewById(R.id.textTitle);</code>
15.	<code> imgGuess = (ImageView) findViewById(R.id.imgGuess);</code>
16.	
17.	<code> quizzesHandler = new QuizzesHandler(this);</code>
18.	<code> stagesHandler = new StagesHandler(this);</code>
19.	



Tabel 5.35 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar (lanjutan)

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar		
Baris ke-	Kode Program	
20.	Drawable quizImage = AssetReader.getDrawableFromAssets(this, quiz.getMedia());	
21.		
22.	Bundle bundle = getIntent().getExtras();	
23.	if(bundle != null){	
24.	this.stage = bundle.getInt("stage");	
25.	this.level = bundle.getInt("level");	
26.	quiz = quizzesHandler.getQuiz(stage, level);	
27.	}	
28.		
29.	textScreen.setText("Level " + quiz.getStage() + " - " + quiz.getLevel());	
30.	imgGuess.setImageDrawable(quizImage);	
31.	imgGuess.startAnimation(smalltobig);	
32.	}	

Penjelasan dari algoritma pada Tabel 5.35 adalah sebagai berikut:

- Baris 1 sampai 5 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *textScreen*, *textQuestion*, *textTitle*, *imgGuess*, *stagesHandler*, *quizzesHandler* dan *quiz*.
- Baris 7 sampai 32 merupakan fungsi untuk memuat halaman. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk mendapatkan data permainan dalam bentuk gambar dari basis data, untuk mendapatkan nilai dari tahapan dan *level* dari halaman sebelumnya serta untuk menampilkan gambar yang harus ditebak pada permainan tahapan tebak kata berdasarkan gambar.

8. Implementasi algoritma menampilkan permainan tahapan tebak kata berdasarkan suara

Tabel 5.36 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara		
Baris ke-	Kode Program	
1.	private TextView textScreen, textQuestion, textTitle;	
2.	private Button btnSound;	
3.	private StagesHandler stagesHandler;	
4.	private QuizzesHandler quizzesHandler;	
5.	private Quiz quiz;	
6.	private TextToSpeech tts;	
7.		
8.	@Override	
9.	protected void onCreate(Bundle savedInstanceState) {	
10.	super.onCreate(savedInstanceState);	
11.	setContentView(R.layout.activity_quiz_stage_two);	
12.		
13.	textQuestion = (TextView) findViewById(R.id.textQuestion);	
14.	textScreen = (TextView) findViewById(R.id.textScreen);	
15.	textTitle = (TextView) findViewById(R.id.textTitle);	
16.	btnSound = (Button) findViewById(R.id.btnSound);	

Tabel 5.36 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara (lanjutan)

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	
Baris ke-	Kode Program
17.	
18.	<code>quizzesHandler = new QuizzesHandler(this);</code>
19.	<code>stagesHandler = new StagesHandler(this);</code>
20.	
21.	<code>Bundle bundle = getIntent().getExtras();</code>
22.	<code>if (bundle != null) {</code>
23.	<code>int stage = bundle.getInt("stage");</code>
24.	<code>int level = bundle.getInt("level");</code>
25.	<code>quiz = quizzesHandler.getQuiz(stage, level);</code>
26.	<code>}</code>
27.	
28.	<code>tts = new TextToSpeech(QuizStageTwoActivity.this, new</code>
29.	<code>TextToSpeech.OnInitListener() {</code>
30.	<code>@Override</code>
31.	<code>public void onInit(int status) {</code>
32.	<code>if (status != TextToSpeech.ERROR) {</code>
33.	<code>tts.setLanguage(Locale.getDefault());</code>
34.	<code>}</code>
35.	<code>};</code>
36.	
37.	<code>textScreen.setText("Level " + quiz.getStage() + " - " +</code>
38.	<code>quiz.getLevel());</code>
39.	<code>btnSound.setOnClickListener(new View.OnClickListener() {</code>
40.	<code>@Override</code>
41.	<code>public void onClick(View v) {</code>
42.	<code>tts.speak(quiz.getAnswer(), TextToSpeech.QUEUE_FLUSH, null,</code>
43.	<code>"reado!-" + quiz.getAnswer());</code>
44.	<code>});</code>
45.	<code>}</code>

Penjelasan dari algoritma pada Tabel 5.36 adalah sebagai berikut:

Baris 1 sampai 6 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *textScreen*, *textQuestion*, *textTitle*, *btnSound*, *stagesHandler*, *quizzesHandler*, *quiz* dan *tts*.

Baris 8 sampai 45 merupakan fungsi untuk memuat halaman. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk mendapatkan data permainan dari basis data, untuk mendapatkan nilai dari tahapan dan *level* dari halaman sebelumnya, untuk mendefinisikan nilai dari variabel *tts* yaitu dengan menggunakan fungsi dari *Text To Speech* untuk mengatur bahasa yang digunakan yaitu menggunakan fungsi *setLanguage()*, untuk menampilkan tulisan *level* dan tahapan serta untuk memberikan perintah ketika tombol dari *btnSound* ditekan yaitu dengan menggunakan fungsi *speak()* untuk menyuarkan cara pembacaan dari kata yang harus ditebak.

9. Implementasi algoritma menampilkan permainan tahapan tebak kata berdasarkan video



Tabel 5.37 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	
Baris ke-	Kode Program
1.	private TextView textScreen, textQuestion, textTitle;
2.	private MediaController mediaController;
3.	private StagesHandler stagesHandler;
4.	private QuizzesHandler quizzesHandler;
5.	private Quiz quiz;
6.	
7.	private YouTubePlayerView youtubePlayerView;
8.	private YouTubePlayer youtubePlayer;
9.	
10.	public static final String API_KEY;
11.	
12.	public static String VIDEO_ID;
13.	
14.	@Override
15.	protected void onCreate(Bundle savedInstanceState) {
16.	super.onCreate(savedInstanceState);
17.	setContentView(R.layout.activity_quiz_stage_three);
18.	
19.	mediaController = new MediaController(this);
20.	stagesHandler = new StagesHandler(this);
21.	quizzesHandler = new QuizzesHandler(this);
22.	
23.	youtubePlayerView = (YouTubePlayerView)
24.	findViewById(R.id.videoView);
25.	youtubePlayerView.initialize(API_KEY, this);
26.	Bundle bundle = getIntent().getExtras();
27.	if (bundle != null) {
28.	int stage = bundle.getInt("stage");
29.	int level = bundle.getInt("level");
30.	quiz = quizzesHandler.getQuiz(stage, level);
31.	}
32.	
33.	textScreen.setText("Level " + quiz.getStage() + " - " +
34.	quiz.getLevel());
35.	}
36.	@Override
37.	public void onInitializationSuccess(YouTubePlayer.Provider
38.	provider, final YouTubePlayer player, boolean wasRestored) {
39.	int timeStart = quiz.getTimeStart();
40.	final int duration = quiz.getDuration();
41.	this.youtubePlayer = player;
42.	VIDEO_ID = quiz.getMedia();
43.	
44.	player.setPlayerStyle(YouTubePlayer.PlayerStyle.CHROMELESS);
45.	player.setPlayerStateChangeListener(playerStateChangeListener);
46.	player.setPlaybackEventListener(playbackEventListener);
47.	
48.	if (!wasRestored) {
49.	player.loadVideo(VIDEO_ID, timeStart);
50.	}
51.	
52.	final Handler handler = new Handler();
53.	handler.postDelayed(new Runnable() {
54.	public void run() {
55.	if (player != null) {
56.	try {



Tabel 5.37 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video (lanjutan)

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	
Baris ke-	Kode Program
57.	if (player.getCurrentTimeMillis() <= duration) {
58.	handler.postDelayed(this, 1000);
59.	} else {
60.	handler.removeCallbacks(this);
61.	player.pause();
62.	}
63.	} catch (IllegalStateException e) {
64.	}
65.	}
66.	}
67.	}, 1000);
68.	}
69.	
70.	@Override
71.	public void onInitializationFailure(YouTubePlayer.Provider
	provider, YouTubeInitializationResult youtubeInitializationResult)
72.	{
	Toast.makeText(this, "Failure to Initialize!",
	Toast.LENGTH_LONG).show();
73.	}
74.	
75.	private YouTubePlayer.PlaybackEventListener playbackEventListener
	= new YouTubePlayer.PlaybackEventListener() {
76.	@Override
77.	public void onPlaying() {
78.	
79.	}
80.	
81.	@Override
82.	public void onPaused() {
83.	
84.	}
85.	
86.	@Override
87.	public void onStopped() {
88.	
89.	}
90.	
91.	@Override
92.	public void onBuffering(boolean b) {
93.	
94.	}
95.	
96.	@Override
97.	public void onSeekTo(int i) {
98.	
99.	}
100.	};
101.	
102.	private YouTubePlayer.PlayerStateChangeListener
	playerStateChangeListener = new
	YouTubePlayer.PlayerStateChangeListener() {
103.	@Override
104.	public void onLoading() {
105.	
106.	}
107.	
108.	@Override
109.	public void onLoaded(String s) {

Tabel 5.37 Implementasi Algoritma Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video (lanjutan)

Algoritma: Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	
Baris ke-	Kode Program
110.	
111.	}
112.	
113.	@Override
114.	public void onAdStarted() {
115.	
116.	}
117.	
118.	@Override
119.	public void onVideoStarted() {
120.	
121.	}
122.	
123.	@Override
124.	public void onVideoEnded() {
125.	
126.	}
127.	
128.	@Override
129.	public void onError(YouTubePlayer.ErrorReason errorReason) {
130.	
131.	}
132.	};
133.	
134.	@Override
135.	protected void onStop() {
136.	if (youTubePlayer != null) {
137.	youTubePlayer.release();
138.	youTubePlayer = null;
139.	}
140.	super.onStop();
141.	}

Penjelasan dari algoritma pada Tabel 5.37 adalah sebagai berikut:

Baris 1 sampai 12 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *textScreen*, *textQuestion*, *textTitle*, *mediaController*, *stagesHandler*, *quizzesHandler*, *quiz*, *youTubePlayerView*, *youTubePlayer*, *API_KEY* dan *VIDEO_ID*.

Baris 14 sampai 34 merupakan fungsi untuk memuat halaman. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel *mediaController*, *stagesHandler* dan *quizzesHandler*, untuk menginisialisasikan nilai dari variabel *youTubePlayerView* dengan menggunakan nilai dari variabel *API_KEY*, untuk mendapatkan nilai tahapan dan *level* berdasarkan halaman sebelumnya, untuk mendapatkan data permainan dari basis data dan untuk menampilkan tulisan tahapan dan *level*.

Baris 36 sampai 68 merupakan fungsi untuk mendeklarasikan dan mendefinisikan nilai dari variabel *timeStart* dan *duration*, untuk mendefinisikan nilai variabel *youTubePlayer* dan *VIDEO_ID*, untuk memuat



video dengan menggunakan variabel *YouTubePlayer* serta untuk memutar video dengan menggunakan variabel *handler* dan menjalankan fungsi yaitu *run()*.

- Baris 70 sampai 100 merupakan fungsi untuk menginisialisasikan ketika terjadi kegagalan untuk memuat dan memutar video pada halaman permainan.
- Baris 102 sampai 132 merupakan fungsi untuk memberikan perintah ketika memutar video dengan beberapa fungsi yaitu fungsi *onLoading()*, *onLoaded()*, *onAdStarted()*, *onVideoStarted()*, *onVideoEnded()* dan *onError()*.
- Baris 134 sampai 141 merupakan fungsi untuk memberhentikan video yang sedang ditampilkan.

10. Implementasi algoritma menyediakan mekanisme untuk jawaban dari permainan

Tabel 5.38 Implementasi Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan

Algoritma: Menyediakan Mekanisme untuk Jawaban dari Permainan	
Baris ke-	Kode Program
1.	<code>private int counter, maxPresCounter;</code>
2.	<code>private EditText editText;</code>
3.	<code>private Button btnReset;</code>
4.	<code>private Quiz quiz;</code>
5.	<code>private StagesHandler stagesHandler;</code>
6.	<code>private QuizzesHandler quizzesHandler;</code>
7.	<code>private SharedPreferences mSharedPreferences;</code>
8.	
9.	<code>@Override</code>
10.	<code>protected void onCreate(Bundle savedInstanceState) {</code>
11.	<code>super.onCreate(savedInstanceState);</code>
12.	
13.	<code>editText = (EditText) findViewById(R.id.editText);</code>
14.	<code>btnReset = (Button) findViewById(R.id.btnReset);</code>
15.	
16.	<code>maxPresCounter = quiz.getQuestion().length();</code>
17.	
18.	<code>quiz.setQuestion(shuffleString(quiz.getQuestion().toCharArray()));</code>
19.	<code>for (char key : quiz.getQuestion().toCharArray()) {</code>
20.	<code>addView(layoutParent, key, editText);</code>
21.	<code>}</code>
22.	
23.	<code>btnReset.setOnClickListener(new View.OnClickListener() {</code>
24.	<code>@Override</code>
25.	<code>public void onClick(View v) {</code>
26.	<code>editText.setText("");</code>
27.	<code>counter = 0;</code>
28.	<code>layoutParent.removeAllViews();</code>
29.	<code>quiz.setQuestion(shuffleString(quiz.getQuestion().toCharArray()));</code>
30.	<code>for (char key : quiz.getQuestion().toCharArray()) {</code>
31.	<code>addView(layoutParent, key, editText);</code>
32.	<code>}</code>
33.	<code>}</code>
34.	<code>});</code>
35.	<code>}</code>
36.	
37.	<code>private void doValidate() {</code>
38.	<code>if (editText.getText().toString().equals(quiz.getAnswer())) {</code>



Tabel 5.38 Implementasi Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan (lanjutan)

Algoritma: Menyediakan Mekanisme untuk Jawaban dari Permainan	
Baris ke-	Kode Program
39.	
40.	final int maxStage = stagesHandler.getStagesCount();
41.	final int maxLevel =
	quizzesHandler.getQuizzesCountByStage(quiz.getStage());
42.	
43.	int currentStage, currentLevel;
44.	currentStage = mSharedPreferences.getInt("currentStage", 1);
45.	currentLevel = mSharedPreferences.getInt("currentLevel", 1);
46.	SharedPreferences.Editor sharedPreferencesEditor =
	mSharedPreferences.edit();
47.	if(quiz.getStage() == currentStage && quiz.getLevel() ==
	currentLevel){
48.	if (currentLevel < maxLevel){
49.	sharedPreferencesEditor.putInt("currentLevel",
	quiz.getLevel() + 1);
50.	}else if(currentLevel == maxLevel){
51.	if(currentStage < maxStage){
52.	sharedPreferencesEditor.putInt("currentStage",
	currentStage + 1);
53.	}
54.	}
55.	}
56.	sharedPreferencesEditor.apply();
57.	
58.	// Remove linear
59.	layoutParent.animate()
60.	.alpha(0)
61.	.setDuration(200)
62.	.setListener(new AnimatorListenerAdapter() {
63.	@Override
64.	public void onAnimationEnd(Animator animation) {
65.	super.onAnimationEnd(animation);
66.	layoutParent.setVisibility(View.GONE);
67.	}
68.	});
69.	
70.	Snackbar snackbar = Snackbar
71.	.make(layoutParent, R.string.correct,
	Snackbar.LENGTH_INDEFINITE)
72.	.setAction(R.string.next, new View.OnClickListener() {
73.	@Override
74.	public void onClick(View view) {
75.	int maxLevel =
	quizzesHandler.getQuizzesCountByStage(quiz.getStage());
76.	if(quiz.getLevel() == maxLevel){
77.	Intent toNextStage = new
	Intent(QuizStageThreeActivity.this, StageActivity.class);
78.	toNextStage.putExtra("stage", quiz.getStage() + 1);
79.	toNextStage.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
80.	startActivity(toNextStage);
81.	finish();
82.	}else{
83.	Intent toNextQuiz = new
	Intent(QuizStageThreeActivity.this, QuizStageThreeActivity.class);
84.	toNextQuiz.putExtra("stage", quiz.getStage());
85.	toNextQuiz.putExtra("level", quiz.getLevel() + 1);
86.	startActivity(toNextQuiz);
87.	finish();
88.	}



Tabel 5.38 Implementasi Algoritma Menyediakan Mekanisme untuk Jawaban dari Permainan (lanjutan)

Baris ke-	Kode Program
89.	}
90.	});
91.	snackbar.setActionTextColor(Color.WHITE);
92.	View sbView = snackbar.getView();
93.	TextView textView = (TextView)
94.	sbView.findViewById(android.support.design.R.id.snackbar_text);
95.	textView.setTextColor(Color.GREEN);
96.	snackbar.show();
97.	else if(maxPresCounter <= counter && !editText.getText().toString().equals(quiz.getAnswer())){
98.	Snackbar snackbar = Snackbar
99.	.make(layoutParent, R.string.wrong,
100.	Snackbar.LENGTH INDEFINITE)
101.	.setAction(R.string.retry, new View.OnClickListener() {
102.	@Override
103.	public void onClick(View view) {
104.	editText.setText("");
105.	counter = 0;
106.	layoutParent.removeAllViews();
107.	quiz.setQuestion(shuffleString(quiz.getQuestion().
108.	toCharArray()));
109.	for (char key : quiz.getQuestion().toCharArray()) {
110.	addView(layoutParent, key, editText);
111.	}
112.	});
113.	snackbar.setActionTextColor(Color.RED);
114.	View sbView = snackbar.getView();
115.	TextView textView = (TextView)
116.	sbView.findViewById(android.support.design.R.id.snackbar_text);
117.	textView.setTextColor(Color.YELLOW);
118.	snackbar.show();
119.	}
120.	private String shuffleString(char[] charArr){
121.	Random rnd = new Random();
122.	for (int i = charArr.length - 1; i > 0; i--) {
123.	int index = rnd.nextInt(i+1);
124.	char a = charArr[index];
125.	charArr[index] = charArr[i];
126.	charArr[i] = a;
127.	return new String(charArr);
128.	}

Penjelasan dari algoritma pada Tabel 5.38 adalah sebagai berikut:

- Baris 1 sampai 7 merupakan algoritma untuk mendeklarasikan variabel yang terdiri atas variabel *counter*, *maxPresCounter*, *editText*, *btnReset*, *quiz*, *stagesHandler*, *quizzesHandler* dan *mSharedPreferences*.
- Baris 9 sampai 35 merupakan fungsi untuk memuat halaman. Fungsi ini terdiri atas algoritma-algoritma yaitu algoritma untuk mendefinisikan nilai dari variabel, untuk mendapatkan data permainan dari basis data dan untuk memberikan perintah ketika tombol yaitu *btnReset* ditekan.



- Baris 37 sampai 117 merupakan fungsi untuk menentukan jawaban yang dijawab oleh pengguna dengan jawaban yang sebenarnya berdasarkan data yang didapatkan pada basis data, untuk mendeklarasikan dan mendefinisikan nilai dari variabel *maxStage*, *maxLevel*, *currentStage*, *currentLevel* dan *sharedPreferencesEditor*, untuk memuat halaman ketika pengguna ingin melanjutkan permainan berdasarkan tahapan dan *level* dan untuk mengacak pilihan alfabet untuk jawaban yang harus dipilih oleh pengguna.
- Baris 119 sampai 128 merupakan fungsi untuk mengacak pilihan alfabet untuk jawaban yang harus dipilih oleh pengguna.

5.2.6 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang sudah dilakukan sebelumnya pada perancangan perangkat lunak. Implementasi ini dilakukan dengan menggunakan bahasa pemrograman XML untuk mengatur tampilan dari aplikasi Reado. Implementasi antarmuka ini terdiri atas 11 implementasi antarmuka dari tampilan aplikasi Reado. Adapun 11 implementasi antarmuka dalam bahasa pemrograman XML dari pengembangan aplikasi Reado, antara lain sebagai berikut:

1. Implementasi antarmuka tampilan aplikasi untuk logo aplikasi Reado

Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan logo dari aplikasi Reado dapat dilihat pada Gambar 5.37.



Gambar 5.37 Implementasi Antarmuka Tampilan Aplikasi untuk Logo Aplikasi Reado

2. Implementasi antarmuka tampilan aplikasi untuk halaman utama aplikasi Reado



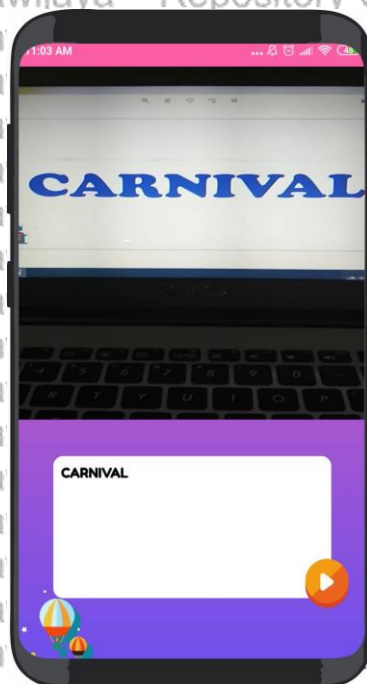
Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan halaman utama dari aplikasi Reado dapat dilihat pada Gambar 5.38.



Gambar 5.38 Implementasi Antarmuka Tampilan Aplikasi untuk Halaman Utama Aplikasi Reado

3. Implementasi antarmuka tampilan aplikasi untuk halaman deteksi kata

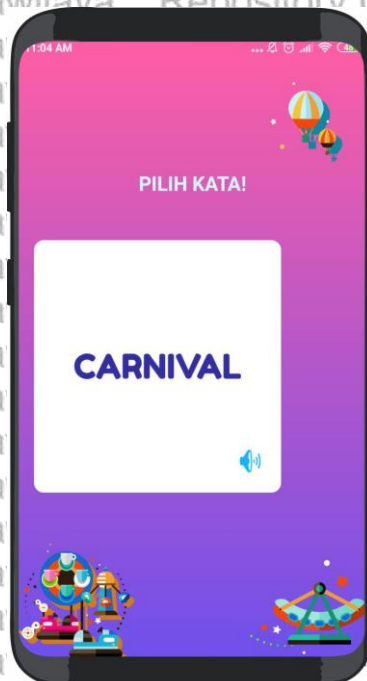
Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan halaman deteksi kata dari aplikasi Reado dapat dilihat pada Gambar 5.39.



Gambar 5.39 Implementasi Antarmuka Tampilan Aplikasi Halaman Deteksi Kata

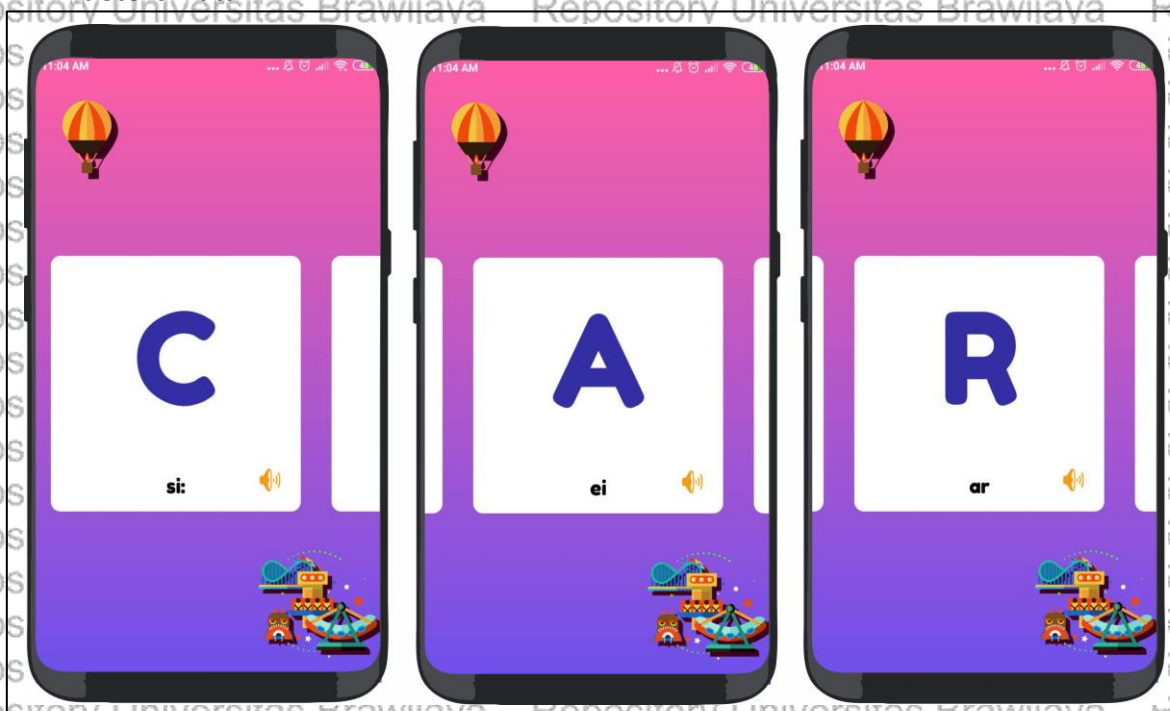
4. Implementasi antarmuka tampilan aplikasi untuk halaman kata dari hasil deteksi kata

Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan halaman kata dari hasil deteksi kata dapat dilihat pada Gambar 5.40.



Gambar 5.40 Implementasi Antarmuka Tampilan Aplikasi Halaman Kata

5. Implementasi antarmuka tampilan aplikasi untuk halaman alfabet dari hasil deteksi kata



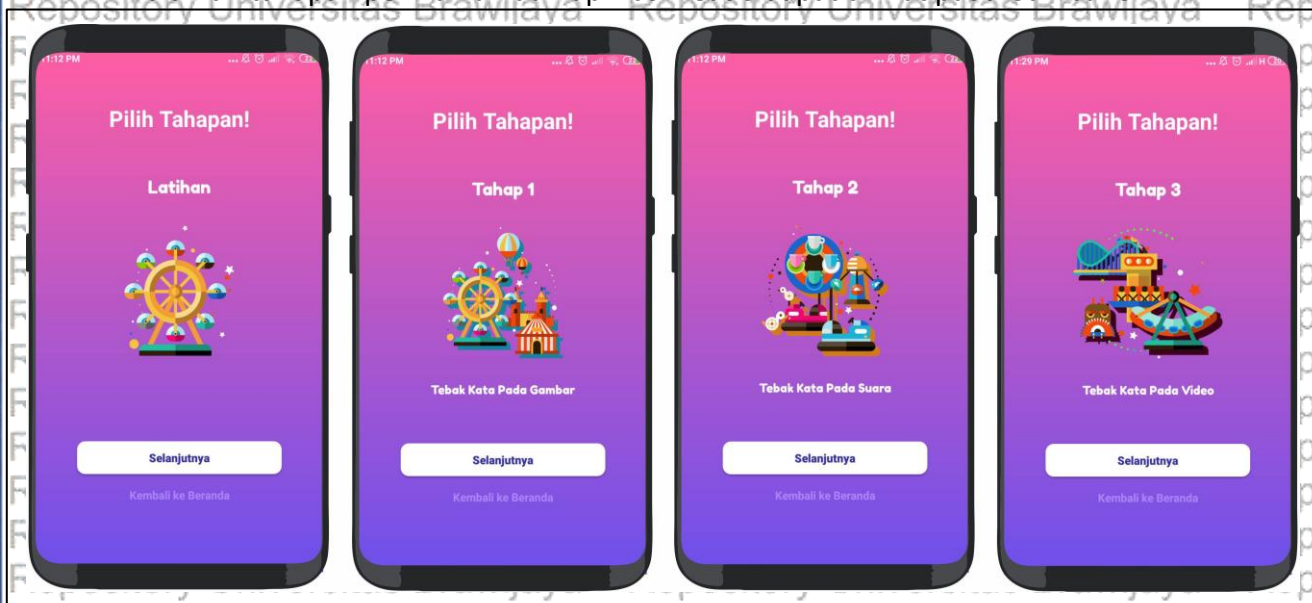
Gambar 5.41 Implementasi Antarmuka Tampilan Aplikasi Halaman Alfabet



Berdasarkan Gambar 5.41 merupakan implementasi antarmuka tampilan aplikasi untuk menampilkan halaman alfabet dari hasil deteksi kata.

6. Implementasi antarmuka tampilan aplikasi untuk halaman tahapan permainan

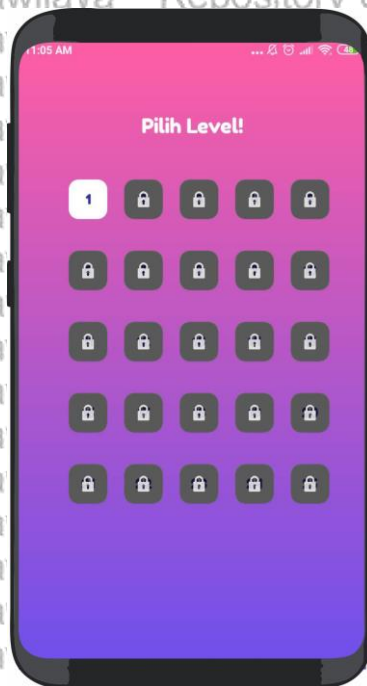
Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan halaman tahapan permainan dari aplikasi Reado dapat dilihat pada Gambar 5.42.



Gambar 5.42 Implementasi Antarmuka Tampilan Aplikasi Tahapan Permainan

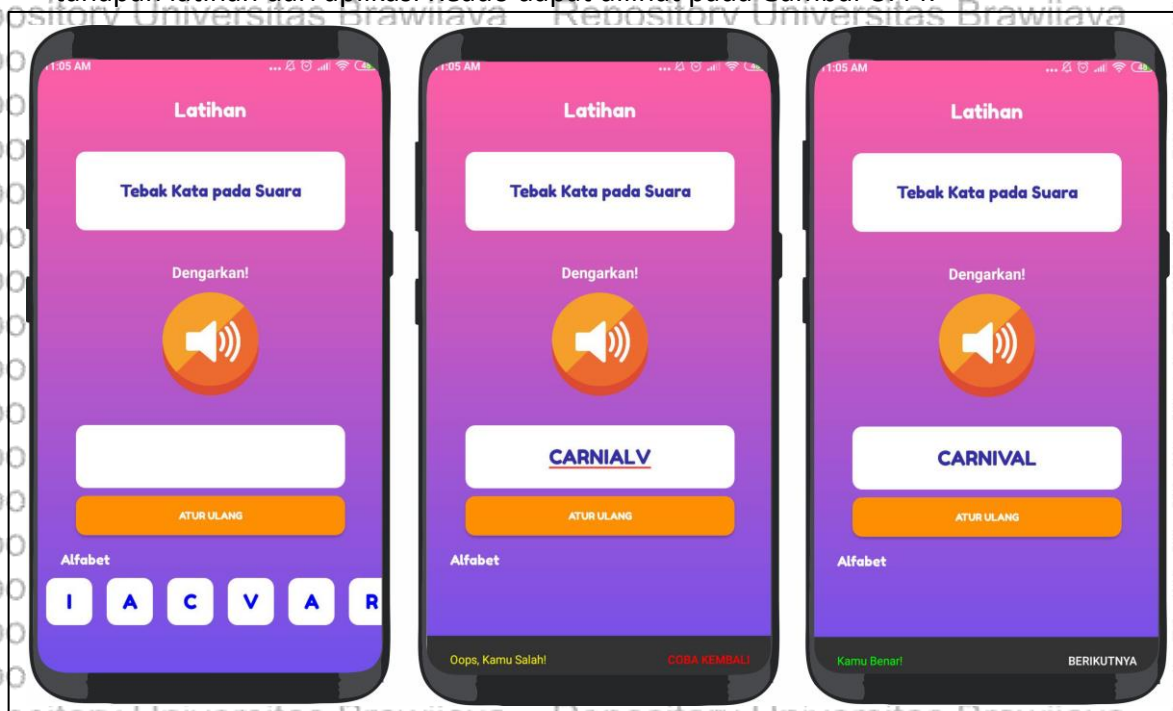
7. Implementasi antarmuka tampilan aplikasi untuk halaman *level* permainan

Adapun implementasi antarmuka tampilan aplikasi untuk menampilkan halaman *level* permainan dari aplikasi Reado dapat dilihat pada Gambar 5.43.



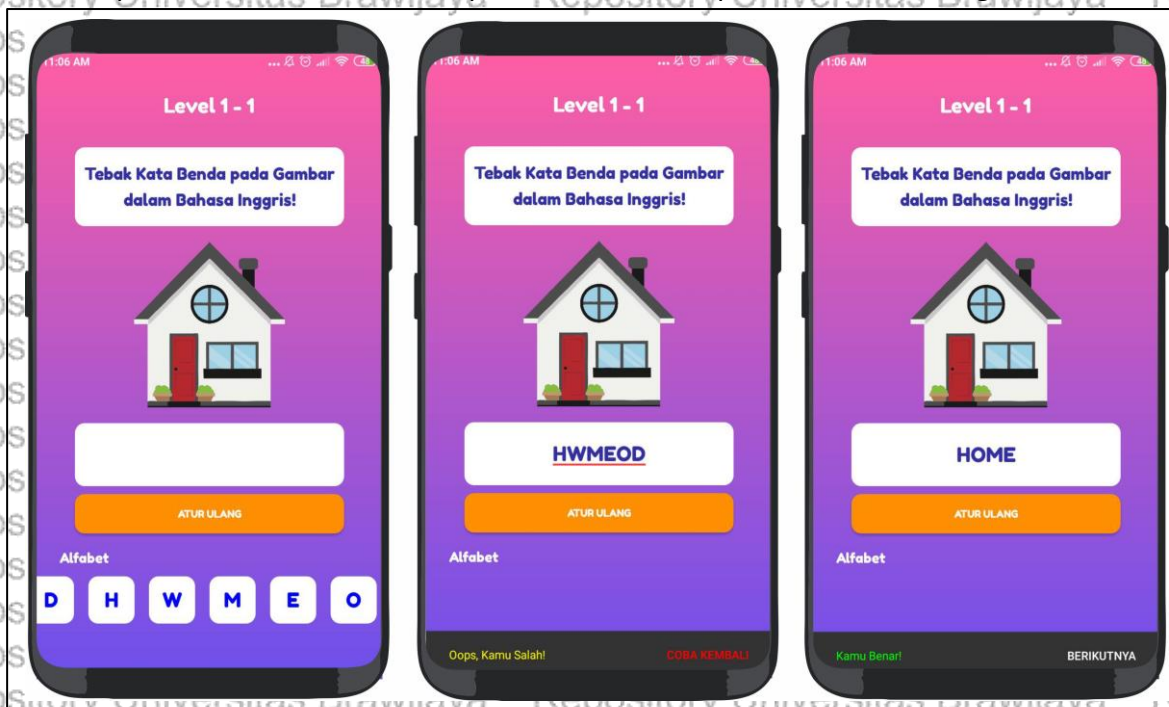
Gambar 5.43 Implementasi Antarmuka Tampilan Aplikasi *Level* Permainan

8. Implementasi antarmuka tampilan aplikasi halaman permainan tahapan latihan
- Adapun implementasi antarmuka tampilan aplikasi untuk halaman permainan tahapan latihan dari aplikasi Readoo dapat dilihat pada Gambar 5.44.



Gambar 5.44 Implementasi Antarmuka Tampilan Aplikasi Tahapan Latihan

9. Implementasi antarmuka tampilan halaman tahapan tebak kata dari gambar

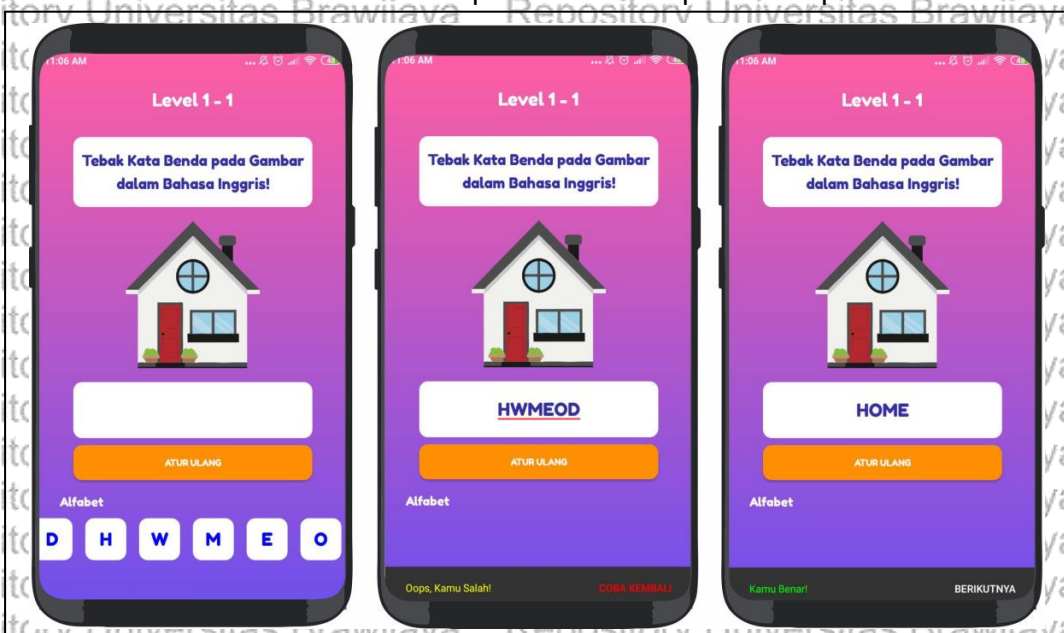


Gambar 5.45 Implementasi Antarmuka Tampilan Tebak Kata dari Gambar

Berdasarkan Gambar 5.45 merupakan implementasi antarmuka tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar.

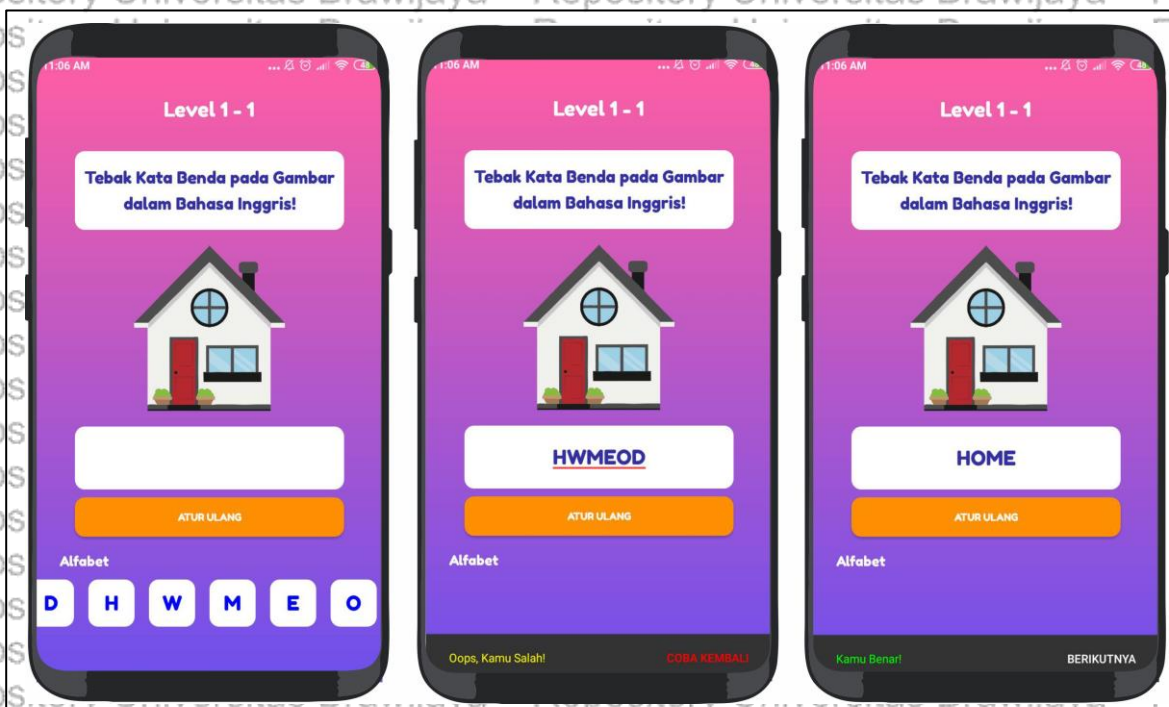
10. Implementasi antarmuka tampilan halaman tahapan tebak kata dari suara

Adapun implementasi antarmuka tampilan aplikasi untuk permainan tahapan tebak kata berdasarkan suara dari aplikasi Reado dapat dilihat pada Gambar 5.46.



Gambar 5.46 Implementasi Antarmuka Tampilan Tebak Kata dari Suara

11. Implementasi antarmuka tampilan halaman tahapan tebak kata berdasarkan video



Gambar 5.47 Implementasi Antarmuka Tampilan Tebak Kata dari Video

Berdasarkan Gambar 5.47 merupakan implementasi antarmuka tampilan aplikasi untuk halaman permainan tahapan tebak kata berdasarkan gambar.



BAB 6 PENGUJIAN

Pada bagian pengujian dari penelitian skripsi ini, akan dijelaskan terkait dengan pengujian-pengujian yang terdiri atas pengujian *black box* dan pengujian *usability* atau kegunaan aplikasi. Pengujian *usability* dilakukan yang menggunakan teknik *equivalence partitioning* dan pengujian *usability* dilakukan berdasarkan parameter pengukuran dari aspek efisiensi, efektivitas dan kepuasan pengguna.

6.1 Pengujian *Black Box*

Pengujian *black box* dilakukan dengan menggunakan teknik *equivalence partitioning*. Pengujian ini bertujuan untuk menguji validasi pada kebutuhan-kebutuhan fungsional dari aplikasi Reado. Pengujian ini juga bertujuan dalam menguji apakah implementasi dari aplikasi sudah berfungsi sesuai dengan kebutuhan fungsional dan skenario *use case* yang telah dibuat sebelumnya pada tahap perancangan. Pengujian *black box* dengan menggunakan teknik *equivalence partitioning* yang dilakukan dapat dilihat pada Tabel 6.1. Pada pengujian ini, terdapat 10 kebutuhan fungsional pada aplikasi Reado yang diujikan. Kebutuhan fungsional tersebut, antara lain sebagai berikut:

1. Sistem harus mampu mendeteksi kata
2. Sistem harus mampu menampilkan kata-kata dari hasil deteksi kata
3. Sistem harus mampu menampilkan alfabet dari kata berdasarkan hasil deteksi kata
4. Sistem harus mampu menentukan tahapan permainan
5. Sistem harus mampu menentukan *level* permainan
6. Sistem harus mampu menampilkan permainan tahapan latihan
7. Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan gambar
8. Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan suara
9. Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan video
10. Sistem harus mampu menyediakan mekanisme untuk jawaban dari permainan

Pengujian *black box* dengan teknik *equivalence partitioning* menerapkan partisi-partisi dari setiap kebutuhan fungsional pada aplikasi. Pengujian ini terdiri atas beberapa bagian yaitu ID kasus uji, deskripsi modul, spesifikasi modul, langkah uji, masukan, kelas *equivalence* yang benar, kelas *equivalence* yang tidak benar, data uji, hasil ekspektasi dan hasil aktual. Dari setiap masing-masing kebutuhan diujikan dan ketika hasil pengujian sama seperti hasil ekspektasi yang sudah ditentukan sebelumnya maka pengujian dari kebutuhan fungsional sistem tersebut memiliki status pada hasil aktual adalah terverifikasi.

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
1	Mendeteksi kata	Sistem harus mampu menyediakan mekanisme untuk mendeteksi kata-kata yang dapat ditangkap oleh kamera secara <i>realtime</i> atau waktu sebenarnya.	1. Sistem menampilkan halaman untuk mendeteksi kata.	Gambar melalui kamera	Input gambar yang terdiri dari kata	Input gambar yang tidak terdiri dari kata	Gambar yang terdiri dari kata "ON"	Sistem menampilkan kata "ON"	Terverifikasi
			2. Aktor mengarahkan kamera terhadap kata yang ingin dideteksi dengan benar.				Gambar yang tidak terdiri dari kata-kata apapun	Sistem tidak menampilkan kata-kata apapun	Terverifikasi
			3. Sistem mendeteksi kata dari gambar yang ditangkap.	Tombol untuk setuju	menekan tombol setuju	4. Sistem menampilkan kata-kata yang dideteksi secara <i>realtime</i> pada halaman.	Kata yang disetujui yaitu kata "ON"	Sistem mengirimkan kata yang terdeteksi yaitu kata "ON"	Terverifikasi
5. Aktor melihat kata-kata yang dapat dideteksi secara <i>realtime</i> .									
6. Aktor menekan tombol untuk menyetujui kata yang sudah dideteksi.									
7. Sistem mengirimkan kata yang terdeteksi untuk ke halaman selanjutnya.									

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
2	Menampilkan kata-kata dari hasil deteksi kata	Sistem harus mampu menampilkan kata dari hasil deteksi kata dan ditampilkan cara pembacaan kata-kata tersebut dalam bentuk suara.	1. Sistem menampilkan kata-kata dari hasil deteksi kata. 2. Aktor melihat kata-kata dari hasil deteksi kata. 3. Aktor menekan tombol. 4. Sistem menampilkan cara pembacaan kata. 5. Aktor mendengarkan cara pembacaan kata.	Tombol suara	Input menekan tombol suara	Input tidak menekan tombol suara	Kata yang dilafalkan adalah "ON" dalam Bahasa Inggris	Sistem menampilkan cara pembacaan kata yaitu "ON" dalam Bahasa Inggris	Terverifikasi
3	Menampilkan alfabet dari kata berdasarkan hasil deteksi kata	Sistem harus mampu menampilkan alfabet dari kata berdasarkan hasil deteksi kata dan ditampilkan cara pembacaan dalam bentuk suara dan tulisan.	1. Sistem menampilkan alfabet. 2. Aktor melihat alfabet. 3. Aktor melihat cara pembacaan yaitu tulisan. 4. Aktor menekan tombol suara. 5. Sistem menampilkan cara pembacaan yaitu suara. 6. Aktor mendengarkan cara pembacaan alfabet.	Tombol suara	Input menekan tombol suara	Input tidak menekan tombol suara	Alfabet yang dilafalkan yaitu "O" Alfabet yang dilafalkan yaitu "N"	Sistem menampilkan cara pembacaan alfabet yaitu "O" Sistem menampilkan cara pembacaan alfabet yaitu "N"	Terverifikasi Terverifikasi

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
4	Menentukan tahapan permainan	Sistem harus mampu menyediakan mekanisme untuk menentukan tahapan permainan yang terdiri atas latihan, tebak kata berdasarkan gambar, tebak kata berdasarkan suara dan tebak kata berdasarkan video.	1. Sistem menampilkan tahapan-tahapan permainan pada halaman yang terdiri atas latihan, tebak kata berdasarkan gambar, suara dan video.				Tahapan permainan yang dipilih yaitu tahapan latihan	Sistem menentukan tahapan latihan	Terverifikasi
			2. Aktor melihat tahapan-tahapan permainan.			Input tidak menekan tombol memilih tahapan	Tahapan yang dipilih yaitu tebak kata dari gambar	Sistem menentukan tahapan tebak kata dari gambar	Terverifikasi
			3. Aktor menekan salah satu tahapan permainan dan memerintahkan sistem untuk memproses.	Tombol memilih tahapan	Input menekan tombol memilih tahapan		Tahapan yang dipilih yaitu tebak kata dari gambar	Sistem menentukan tahapan tebak kata dari gambar	Terverifikasi
			4. Sistem menentukan tahapan permainan yang dipilih aktor.				Tahapan yang dipilih yaitu tebak kata dari suara	Sistem menentukan tahapan tebak kata dari suara	Terverifikasi
							Tahapan yang dipilih yaitu tebak kata dari video	Sistem menentukan tahapan tebak kata dari video	Terverifikasi

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
5	Menentukan level permainan	Sistem harus mampu menyediakan mekanisme untuk menentukan level permainan berdasarkan pilihan aktor yang terdiri atas level pada tahapan permainan tebak kata berdasarkan gambar, suara dan video.	1. Sistem menampilkan level permainan pada halaman yang terdiri atas level tahapan tebak kata berdasarkan gambar, suara dan video.				Level yang dipilih yaitu level 1	Sistem menentukan level yaitu level 1	Terverifikasi
			2. Aktor melihat level permainan.				Level yang dipilih yaitu level 2	Sistem menentukan level yaitu level 2	Terverifikasi
			3. Aktor menekan salah satu level permainan dan tombol memerintahkan sistem untuk memproses.	Input menekan tombol level		Input tidak menekan tombol level	Level yang dipilih yaitu level 3	Sistem menentukan level yaitu level 3	Terverifikasi
			4. Sistem menentukan level permainan yang dipilih aktor.				Level yang dipilih yaitu level 4	Sistem menentukan level yaitu level 4	Terverifikasi
							Level yang dipilih yaitu level 5	Sistem menentukan level yaitu level 5	Terverifikasi

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
6	Menampilkan permainan tahapan latihan	Sistem harus mampu menampilkan permainan tahapan latihan.	1. Sistem menampilkan halaman permainan untuk tahapan latihan yang terdiri atas tombol suara. 2. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 3. Sistem menampilkan cara pembacaan kata yang harus ditebak dalam bentuk suara. 4. Aktor mendengarkan cara pembacaan kata.	Tombol suara	Input menekan tombol suara	Input tidak menekan tombol suara	Kata yang dilafalkan adalah "ON" dalam Bahasa Inggris	Sistem menampilkan cara pembacaan kata yaitu "ON" dalam Bahasa Inggris	Terverifikasi
7	Menampilkan permainan tahapan tebak kata berdasarkan gambar	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan gambar.	1. Sistem menampilkan halaman permainan yang terdiri atas gambar yang memvisualisasikan kata yang harus ditebak. 2. Aktor melihat gambar dari kata yang harus ditebak.				-	Sistem menampilkan gambar yang harus ditebak.	Terverifikasi

Tabel 6.1 Pengujian Black Box dengan Menggunakan Teknik Equivalence Partitioning pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas Equivalence yang Benar	Kelas Equivalence yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
8	Menampilkan permainan tahapan tebak kata berdasarkan suara	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan suara.	1. Sistem menampilkan halaman permainan yang terdiri atas tombol suara. 2. Aktor menekan tombol suara dan memerintahkan sistem untuk memproses. 3. Sistem menampilkan cara pembacaan kata yang harus ditebak dalam bentuk suara. 4. Aktor mendengarkan cara pembacaan kata.	Tombol suara	Input menekan tombol suara	Input tidak menekan tombol suara	Kata yang dilafalkan adalah "LION" dalam Bahasa Inggris	Sistem menampilkan cara pembacaan kata yaitu "LION" dalam Bahasa Inggris	Terverifikasi
9	Menampilkan permainan tahapan tebak kata berdasarkan video	Sistem harus mampu menampilkan permainan tahapan tebak kata berdasarkan video.	1. Sistem menampilkan halaman permainan yang terdiri atas video yang memvisualisasikan kata yang harus ditebak. 2. Aktor melihat video dari kata yang harus ditebak.	-	-	-	-	Sistem menampilkan video yang memvisualisasikan gambar yang harus ditebak.	Terverifikasi

Tabel 6.1 Pengujian Black Box dengan Menggunakan Teknik Equivalence Partitioning pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas Equivalence yang Benar	Kelas Equivalence yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
10	Menyediakan mekanisme untuk jawaban dari permainan	Sistem harus mampu menyediakan mekanisme untuk jawaban dan menentukan jawaban yang benar atau salah dari permainan disetiap tahapan. Jawaban yang harus dipilih terdiri atas alfabet-alfabet dari kata yang harus ditebak.	1. Sistem menampilkan halaman.				Jawaban yang dipilih yaitu "HEMO", pada tebak kata dari gambar level 1	Sistem menampilkan jawaban yaitu "HEMO"	Terverifikasi
			2. Aktor menekan tombol pilihan jawaban alfabet.		Input pilihan jawaban yang salah	Input tidak memilih jawaban			
			3. Sistem menampilkan jawaban.						
			4. Aktor menekan tombol pilihan jawaban mengatur ulang.	Tombol pilihan jawaban		Input tidak memilih jawaban yang benar	Jawaban yang dipilih yaitu "HOME", pada tebak kata dari gambar level 1	Sistem menampilkan jawaban yaitu "HOME"	Terverifikasi
			5. Sistem menampilkan alfabet jawaban aktor		Input pilihan jawaban yang benar				
			6. Aktor melihat jawaban.						
			7. Sistem melakukan validasi.						
			8. Sistem menampilkan hasil jawaban.						
			9. Aktor melihat hasil jawaban.		Input	Input tidak menekan tombol mengatur ulang	Jawaban yang sudah dipilih yaitu "HEMO" harus diubah	Sistem menampilkan jawaban yang sudah diatur ulang yaitu "" atau kosong	Terverifikasi
			10. Aktor menekan tombol selanjutnya.	Tombol mengatur ulang					
			11. Sistem menampilkan halaman kembali.						

Tabel 6.1 Pengujian Black Box dengan Menggunakan Teknik Equivalence Partitioning pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas Equivalence yang Benar	Kelas Equivalence yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
10	Menyediakan mekanisme untuk jawaban dari permainan	Sistem harus mampu menyediakan mekanisme untuk jawaban dan menentukan jawaban yang benar atau salah dari permainan disetiap tahapan. Jawaban yang harus dipilih terdiri atas alfabet-alfabet dari kata yang harus ditebak.	1. Sistem menampilkan halaman.				Jawaban yang dipilih yaitu “HEMO”, pada tebak kata dari gambar level 1	Sistem menampilkan hasil jawaban yaitu “Oops, Kamu Salah!” dan tombol yaitu “Coba Kembali”	Terverifikasi
			2. Aktor menekan tombol pilihan jawaban alfabet.	Input pilihan jawaban yang salah	Input tidak memilih jawaban		Jawaban yang dipilih yaitu “HOME”, pada tebak kata dari gambar level 1	Sistem menampilkan hasil jawaban yaitu “Kamu Benar!” dan tombol yaitu “Berikutnya”	Terverifikasi
			3. Sistem menampilkan jawaban.		Input pilihan jawaban yang salah				
			4. Aktor menekan tombol mengatur ulang.						
			5. Sistem menampilkan jawaban aktor.	Tombol pilihan jawaban alfabet					
			6. Aktor melihat jawaban.						
			7. Sistem melakukan validasi.						
			8. Sistem menampilkan hasil jawaban.	Input pilihan jawaban yang benar					
			9. Aktor melihat hasil jawaban.						
			10. Aktor menekan tombol hasil jawaban.						
			11. Sistem menampilkan halaman kembali.						

Tabel 6.1 Pengujian *Black Box* dengan Menggunakan Teknik *Equivalence Partitioning* pada Aplikasi Reado (lanjutan)

ID Kasus Uji	Deskripsi Modul	Spesifikasi Modul	Langkah Uji	Masukan	Kelas <i>Equivalence</i> yang Benar	Kelas <i>Equivalence</i> yang Tidak Benar	Data Uji	Hasil Ekspetasi	Hasil Aktual
10	Menyediakan mekanisme untuk jawaban dari permainan	Sistem harus mampu menyediakan mekanisme untuk jawaban dan menentukan jawaban yang benar atau salah dari permainan disetiap tahapan. Jawaban yang harus dipilih terdiri atas alfabet-alfabet dari kata yang harus ditebak.	1. Sistem menampilkan halaman. 2. Aktor menekan tombol pilihan jawaban alfabet. 3. Sistem menampilkan jawaban. 4. Aktor menekan tombol mengatur ulang. 5. Sistem menampilkan jawaban aktor. 6. Aktor melihat jawaban. 7. Sistem melakukan validasi. 8. Sistem menampilkan hasil jawaban. 9. Aktor melihat hasil jawaban. 10. Aktor menekan tombol hasil jawaban. 11. Sistem menampilkan halaman kembali.	Tombol hasil jawaban	Input menekan tombol "Coba Kembali"	Input tidak menekan tombol "Coba Kembali"	Hasil jawaban yang dipilih yaitu salah	Sistem menampilkan halaman tahapan permainan kembali dengan level yang sama dengan sebelumnya	Terverifikasi
					Input menekan tombol "Berikutnya"	Input tidak menekan tombol "Berikutnya"	Hasil jawaban yang dipilih yaitu benar	Sistem menampilkan halaman tahapan permainan kembali dengan level selanjutnya	Terverifikasi

Dari hasil pengujian *black box* yang sudah dilakukan dengan menggunakan teknik *equivalence partitioning* terhadap kebutuhan-kebutuhan fungsional pada aplikasi Reado, hasil tersebut dapat dilihat pada Tabel 6.2. Berdasarkan hasil pengujian tersebut, semua kebutuhan fungsional pada aplikasi Reado memiliki status terverifikasi. Dari hasil tersebut, hal ini menyatakan bahwa kebutuhan-kebutuhan yang sudah diimplementasikan pada aplikasi Reado sudah memenuhi semua kebutuhan-kebutuhan dari pengguna.

Tabel 6.2 Kebutuhan Fungsional dari Hasil Pengujian *Black Box*

No.	Kode	Nama Fungsional	Status
1.	SRS-RD-01	Mendeteksi Kata	Terverifikasi
2.	SRS-RD-02	Menampilkan Kata-Kata dari Hasil Deteksi Kata	Terverifikasi
3.	SRS-RD-03	Menampilkan Alfabet dari Kata Berdasarkan Hasil Deteksi Kata	Terverifikasi
4.	SRS-RD-04	Menentukan Tahapan Permainan	Terverifikasi
5.	SRS-RD-05	Menentukan <i>Level</i> Permainan	Terverifikasi
6.	SRS-RD-06	Menampilkan Permainan Tahapan Latihan	Terverifikasi
7.	SRS-RD-07	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Gambar	Terverifikasi
8.	SRS-RD-08	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Suara	Terverifikasi
9.	SRS-RD-09	Menampilkan Permainan Tahapan Tebak Kata Berdasarkan Video	Terverifikasi
10.	SRS-RD-10	Menyediakan Mekanisme untuk Jawaban dari Permainan	Terverifikasi

6.2 Pengujian *Usability*

Pengujian *usability* atau kegunaan aplikasi digunakan untuk menilai tingkat kegunaan aplikasi berdasarkan aspek efisiensi, efektivitas dan kepuasan pengguna. Pengujian *usability* ini diukur berdasarkan hasil observasi dari subjek penelitian yaitu anak-anak. Pengujian ini dilakukan dengan 22 anak-anak dari kegiatan sosial Komunitas Katalis Pendidikan yaitu pada kelas 1 sampai kelas 6 SD. Subjek penelitian yaitu anak-anak di dalam pengujian ini dapat dilihat pada Tabel 6.3. Dari setiap anak-anak diharuskan untuk melakukan tugas-tugas yang diberikan dan



mengisikan kuesioner pada tahapan pengujian ini. Terdapat 7 tugas yang harus diselesaikan oleh partisipan dalam penggunaan aplikasi Reado yang mana secara detail terdapat pada Tabel 6.4.

Tabel 6.3 Subjek Penelitian dari Pengujian Aplikasi Reado

No.	Subjek Penelitian	Kelas
1	S1	1
2	S2	1
3	S3	1
4	S4	1
5	S5	1
6	S6	2
7	S7	2
8	S8	2
9	S9	2
10	S10	2
11	S11	3
12	S12	3
13	S13	3
14	S14	3
15	S15	4
16	S16	4
17	S17	4
18	S18	5
19	S19	5
20	S20	5
21	S21	6
22	S22	6

Tabel 6.4 Tugas Pengguna untuk Pengujian Aplikasi Reado

Tugas Pengguna untuk Pengujian Aplikasi Reado	
Tugas	Deskripsi Tugas Pengguna
T1	Anak-anak diminta untuk mendeteksi kata yang ada disekitar dengan aplikasi.



Tabel 6.4 Tugas Pengguna untuk Pengujian Aplikasi Reado (lanjutan)

Tugas Pengguna untuk Pengujian Aplikasi Reado	
Tugas	Deskripsi Tugas Pengguna
T2	Anak-anak diminta untuk mendengarkan pelafalan kata dalam bahasa Inggris yang telah diambil pada aplikasi.
T3	Anak-anak diminta untuk mendengarkan pelafalan alfabet.
T4	Anak-anak diminta untuk memainkan permainan tebak kata tahapan latihan.
T5	Anak-anak diminta untuk memainkan permainan tebak kata tahapan tebak kata berdasarkan gambar.
T6	Anak-anak diminta untuk memainkan permainan tebak kata tahapan tebak kata berdasarkan suara.
T7	Anak-anak diminta untuk memainkan permainan tebak kata tahapan tebak kata berdasarkan video.

6.2.1 Data Pengujian

Pada tahapan dari pengujian *usability* ini, data-data pengujian yang diambil, antara lain sebagai berikut:

a. Keberhasilan Pengguna dalam Menyelesaikan Tugas

Keberhasilan pengguna ditunjukkan ketika pengguna berhasil atau tidak berhasil dalam menyelesaikan setiap tugas yang diberikan kepada pengguna. Apabila pengguna berhasil menyelesaikan tugas maka akan bernilai 1, sedangkan jika pengguna tidak berhasil maka akan bernilai 0. Adapun keberhasilan pengguna dalam menyelesaikan tugas dapat dilihat pada Tabel 6.5.

Tabel 6.5 Keberhasilan Pengguna Dalam Menyelesaikan Tugas

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S1	1	1	1	1	1	1	1
S2	1	1	1	0	1	1	1
S3	1	1	1	0	1	1	1
S4	1	1	1	0	1	1	1



Tabel 6.5 Keberhasilan Pengguna Dalam Menyelesaikan Tugas (lanjutan)

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S5	1	1	1	1	1	1	1
S6	1	1	1	1	1	1	0
S7	1	1	1	1	1	1	1
S8	1	1	1	1	1	1	1
S9	1	1	1	1	1	1	1
S10	1	1	1	1	1	1	1
S11	1	1	1	1	1	1	0
S12	1	1	1	1	1	1	1
S13	1	1	1	1	1	1	1
S14	1	1	1	0	1	1	1
S15	1	1	1	1	1	1	1
S16	1	1	1	1	1	1	1
S17	1	1	1	1	1	1	1
S18	1	1	1	1	1	1	1
S19	1	1	1	1	1	1	1
S20	1	1	1	1	1	1	1
S21	1	1	1	1	1	1	1
S22	1	1	1	1	1	1	1

b. Waktu yang Dibutuhkan untuk Menyelesaikan Tugas

Waktu untuk menyelesaikan setiap tugas pada pengujian dicatat dan ditunjukkan dalam satuan detik. Waktu penyelesaian dihitung dengan melihat waktu pada hasil rekaman video pada saat melakukan pengujian. Adapun waktu yang dibutuhkan pengguna untuk menyelesaikan setiap tugas dapat dilihat pada Tabel 6.6.

Tabel 6.6 Waktu yang Dibutuhkan untuk Menyelesaikan Tugas

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S1	21	7	7	29	13	15	53
S2	4	4	7	12*	11	16	33



Tabel 6.6 Waktu yang Dibutuhkan untuk Menyelesaikan Tugas (lanjutan)

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S3	5	3	8	26*	18	10	51
S4	7	7	2	26*	10	7	46
S5	9	2	2	18	11	16	56
S6	15	4	6	40	45	18	58
S7	3	3	2	10	22	22	32
S8	7	4	4	24	20	13	26
S9	7	7	3	11	9	27	45
S10	6	3	2	10	10	11	36
S11	21	4	3	32	11	15	38*
S12	8	2	6	15	19	24	40
S13	6	4	2	7	8	8	22
S14	22	5	3	28*	9	19	27
S15	12	17	6	12	39	16	28
S16	8	4	3	9	13	14	16
S17	7	3	2	25	8	7	23
S18	6	5	3	7	11	16	56
S19	17	3	2	7	11	12	29
S20	4	2	3	4	31	5	17
S21	7	6	2	10	11	15	20
S22	13	2	3	12	6	9	18

c. Jumlah Kesalahan Pengguna dari Setiap Tugas

Jumlah kesalahan pengguna dari setiap tugas ditunjukkan dengan banyaknya kesalahan-kesalahan pengguna dari pengaksesan halaman, penekanan tombol serta kesalahan di dalam menjawab permainan pada aplikasi. Adapun jumlah kesalahan pengguna dapat dilihat pada Tabel 6.7.

Tabel 6.7 Jumlah Kesalahan Pengguna dari Setiap Tugas

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S1	0	0	0	1	0	0	1



Tabel 6.7 Jumlah Kesalahan Pengguna dari Setiap Tugas (lanjutan)

Subjek Penelitian	Tugas Pengguna						
	T1	T2	T3	T4	T5	T6	T7
S2	0	0	0	1	0	1	0
S3	0	0	0	3	1	0	1
S4	0	0	0	1	0	0	2
S5	0	0	0	0	0	0	0
S6	0	0	0	1	2	0	1
S7	0	0	0	0	2	2	1
S8	0	0	0	1	0	0	0
S9	0	0	0	1	0	2	2
S10	0	0	0	0	0	0	2
S11	0	0	0	1	0	0	0
S12	0	0	0	0	0	4	1
S13	0	0	0	0	0	0	0
S14	1	0	0	1	0	0	1
S15	0	0	0	0	0	0	0
S16	0	0	0	0	0	0	0
S17	0	0	0	2	0	0	0
S18	0	0	0	1	0	0	2
S19	0	0	0	0	0	0	0
S20	0	0	0	0	0	0	0
S21	0	0	0	0	0	0	0
S22	0	0	0	0	0	0	0
Total	1	0	0	14	5	9	14

d. Hasil Kuesioner

Hasil dari kuesioner didapatkan setelah anak-anak sudah menggunakan aplikasi Reado. Anak-anak diharuskan untuk mengisi kuesioner yang sudah disediakan yang terdiri atas 10 pernyataan. Setiap pernyataan akan dinilai dengan skor yang terdiri dari 5 yaitu skor 1 untuk jawaban “Sangat Tidak Setuju”, 2 untuk jawaban “Tidak Setuju”, 3 untuk jawaban “Cukup Setuju”, 4 untuk jawaban “Setuju” dan 5 untuk jawaban “Sangat Setuju”. Adapun hasil dari kuesioner yang dilakukan dapat dilihat pada Tabel 6.8.



Tabel 6.8 Hasil Kuesioner

Subjek Penelitian	Pernyataan Kuesioner									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
S1	4	3	5	3	5	2	4	3	5	5
S2	5	3	4	3	5	3	4	1	4	1
S3	5	3	5	2	5	2	3	1	5	1
S4	5	2	4	3	5	2	4	2	5	2
S5	3	1	5	2	4	1	4	1	5	2
S6	3	1	5	3	3	1	5	2	4	5
S7	4	3	5	1	5	1	3	2	5	2
S8	4	3	5	2	5	2	5	2	5	2
S9	5	2	5	2	5	1	5	2	5	1
S10	4	2	5	1	5	2	5	2	5	2
S11	5	2	5	1	5	1	5	1	5	5
S12	4	1	5	2	4	2	4	1	5	2
S13	5	1	5	3	5	1	4	1	4	3
S14	4	3	5	3	3	3	4	2	4	1
S15	5	2	5	3	4	2	5	2	4	1
S16	5	2	3	3	4	2	5	2	4	1
S17	3	1	5	2	5	1	4	1	4	2
S18	4	2	5	2	4	1	5	2	5	2
S19	1	1	3	1	3	2	3	2	3	2
S20	3	2	5	2	3	2	3	2	3	2
S21	5	1	5	1	5	1	5	1	5	1
S22	4	2	4	2	4	2	4	2	4	4

6.2.2 Aspek Efisiensi

Di dalam menentukan tingkat kegunaan aplikasi berdasarkan aspek efisiensi dibutuhkan data-data pengujian dengan menggunakan data dari Tabel 6.5 dan Tabel 6.6. Data pada Tabel 6.5 merupakan data yang menunjukkan keberhasilan pengguna dalam menyelesaikan tugas, sedangkan untuk data pada Tabel 6.6 merupakan data yang menunjukkan waktu pengguna dalam menyelesaikan tugas. Aspek efisiensi digunakan untuk mengukur tingkat kegunaan aplikasi dari masing-masing kelas pada subjek penelitian yang terdiri atas kelas 1, 2, 3, 4, 5, dan 6.



Adapun keterangan-keterangan untuk menentukan nilai dari efisiensi relatif keseluruhan dari setiap masing-masing kelas, antara lain sebagai berikut:

$$\begin{aligned} R &= 10 \\ N &= 7 \end{aligned}$$

Berikut merupakan perhitungan efisiensi relatif keseluruhan untuk menentukan aspek efisiensi dalam mengukur tingkat kegunaan aplikasi ReadO:

$$\begin{aligned} \text{Efisiensi relatif keseluruhan (1 SD)} &= \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \% \\ &= \frac{(1 \times 21) + \dots + (1 \times 26)}{21 + 7 + \dots + 16 + 56} \times 100 \% \\ &= \frac{508}{572} \times 100 \% \\ &= 0,8881 \times 100 \% \\ &= 88,81\% \approx 89 \% \end{aligned}$$

$$\begin{aligned} \text{Efisiensi relatif keseluruhan (2 SD)} &= \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \% \\ &= \frac{(1 \times 15) + \dots + (1 \times 36)}{15 + 4 + \dots + 11 + 36} \times 100 \% \\ &= \frac{497}{565} \times 100 \% \\ &= 0,8796 \times 100 \% \\ &= 87,96\% \approx 88 \% \end{aligned}$$

$$\begin{aligned} \text{Efisiensi relatif keseluruhan (3 SD)} &= \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \% \\ &= \frac{(1 \times 21) + \dots + (1 \times 27)}{21 + 4 + \dots + 19 + 27} \times 100 \% \\ &= \frac{342}{408} \times 100 \% \\ &= 0,8382 \times 100 \% \\ &= 83,8235\% \approx 84 \% \end{aligned}$$

$$\begin{aligned} \text{Efisiensi relatif keseluruhan (4 SD)} &= \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \% \\ &= \frac{(1 \times 12) + \dots + (1 \times 23)}{12 + 17 + \dots + 7 + 23} \times 100 \% \\ &= \frac{272}{272} \times 100 \% \end{aligned}$$



$$= 1 \times 100 \%$$

$$= 100 \%$$

$$\text{Efisiensi relatif keseluruhan (5 SD)} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \%$$

$$= \frac{(1 \times 6) + \dots + (1 \times 17)}{6 + 5 + \dots + 5 + 17} \times 100 \%$$

$$= \frac{251}{251} \times 100 \%$$

$$= 1 \times 100 \%$$

$$= 100 \%$$

$$\text{Efisiensi relatif keseluruhan (6 SD)} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times 100 \%$$

$$= \frac{(1 \times 7) + \dots + (1 \times 18)}{7 + 6 + \dots + 9 + 18} \times 100 \%$$

$$= \frac{134}{134} \times 100 \%$$

$$= 1 \times 100 \%$$

$$= 100 \%$$

Dari aspek efisiensi, nilai hasil perhitungan dengan menggunakan perhitungan efisiensi relatif keseluruhan dari setiap masing-masing kelas dapat dilihat pada Tabel 6.9. Berdasarkan Tabel 6.9 ditunjukkan bahwa nilai efisiensi relatif keseluruhan yang didapatkan adalah kelas 1 SD yaitu 89%, kelas 2 SD yaitu 88%, kelas 3 SD yaitu 84%, kelas 4 SD yaitu 100%, kelas 5 SD yaitu 100% dan kelas 6 SD yaitu 100%. Dari hasil perhitungan tersebut, maka didapatkan tingkat kepercayaan dari setiap masing-masing kelas adalah kelas 1 SD yaitu “normal”, kelas 2 SD yaitu “normal”, kelas 3 SD yaitu “normal”, kelas 4 SD yaitu “baik”, kelas 5 SD yaitu “baik”, dan kelas 6 SD yaitu “baik”. Jika diambil rata-rata dari pengujian ini, maka didapatkan bahwa penilaian berdasarkan aspek efisiensi pada aplikasi Read memiliki nilai rata-rata yaitu 93,5% dan memiliki tingkat kepercayaan yaitu “baik”.

Tabel 6.9 Hasil Nilai Efisiensi Relatif Keseluruhan dari Setiap Kelas

Kelas	Nilai Efisiensi Relatif Keseluruhan (persentase)	Tingkat Kepercayaan
1	89	Baik
2	88	Normal
3	84	Normal
4	100	Baik



Tabel 6.9 Hasil Nilai Efisiensi Relatif Keseluruhan dari Setiap Kelas (lanjutan)

Kelas	Nilai Efisiensi Relatif Keseluruhan (persentase)	Tingkat Kepercayaan
5	100	Baik
6	100	Baik
Rata-Rata	93,5	Baik

6.2.3 Aspek Efektivitas

Pengukuran pada aspek efektivitas dilakukan dengan pengukuran tingkat kesalahan dari penggunaan aplikasi Reado berdasarkan masing-masing kelas. Pengukuran ini dilakukan dengan menggunakan data-data yaitu total kesalahan dan total kesempatan. Total kesalahan merupakan jumlah kesalahan yang dilakukan oleh pengguna yaitu data pengujian pada Tabel 6.7. Total kesempatan adalah jumlah kesempatan yang dimiliki oleh pengguna untuk menyelesaikan setiap tugas yang diberikan. Langkah-langkah pada kesempatan pengguna untuk menyelesaikan tugas merupakan kesatuan langkah pada saat pengguna menyelesaikan setiap tugas secara berurutan. Adapun total kesempatan untuk menyelesaikan tugas pengguna secara rinci dijelaskan pada Tabel 6.10.

Tabel 6.10 Total Kesempatan untuk Menyelesaikan Tugas Pengguna

Tugas Pengguna	Langkah Penyelesaian Tugas	Jumlah Langkah
T1	Langkah 1: Memilih tombol Ambil Kata	2
	Langkah 2: Menekan tombol untuk menyetujui kata yang dideteksi	
T2	Langkah 1: Memilih tombol Suara untuk mendengarkan kata	1
T3	Langkah 1: Memilih kata dari hasil deteksi	2
	Langkah 2: Memilih tombol Suara untuk mendengarkan alfabet	
T4	Langkah 1: Memilih tahapan Latihan	4
	Langkah 2: Memilih tombol Suara untuk mendengarkan kata yang harus ditebak	
	Langkah 3: Memilih jawaban	
	Langkah 4: Memilih tombol Berikutnya	



Tabel 6.10 Total Kesempatan untuk Menyelesaikan Tugas Pengguna (lanjutan)

Tugas Pengguna	Langkah Penyelesaian Tugas	Jumlah Langkah
T5	Langkah 1: Memilih tahapan tebak kata dari gambar	4
	Langkah 2: Memilih /level/ pada tahapan	
	Langkah 3: Memilih jawaban dari pilihan jawaban	
	Langkah 4: Memilih tombol Berikutnya	
T6	Langkah 1: Memilih tahapan tebak kata dari suara	5
	Langkah 2: Memilih /level/ pada tahapan	
	Langkah 3: Memilih tombol Suara untuk mendengarkan kata yang harus ditebak	
	Langkah 4: Memilih jawaban dari pilihan jawaban	
	Langkah 5: Memilih tombol Berikutnya	
T7	Langkah 1: Memilih tahapan tebak kata dari video	4
	Langkah 2: Memilih /level/ pada tahapan	
	Langkah 3: Memilih jawaban dari pilihan jawaban	
	Langkah 4: Memilih tombol Berikutnya	

Adapun perhitungan tingkat kesalahan setiap tugas dari salah satu kelas yaitu 1 SD untuk menentukan aspek efektivitas dalam mengukur tingkat kegunaan aplikasi, antara lain sebagai berikut:

$$\text{Tingkat Kesalahan } T1 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T1 = \frac{0}{2 \times 5}$$

$$T1 = 0$$

$$\text{Tingkat Kesalahan } T2 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T2 = \frac{0}{1 \times 5}$$



$$T2 = 0$$

$$\text{Tingkat Kesalahan } T3 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T3 = \frac{0}{2 \times 5}$$

$$T3 = 0$$

$$\text{Tingkat Kesalahan } T4 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T4 = \frac{6}{4 \times 5}$$

$$T4 = 0,3$$

$$\text{Tingkat Kesalahan } T5 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T5 = \frac{1}{4 \times 5}$$

$$T5 = 0,05$$

$$\text{Tingkat Kesalahan } T6 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T6 = \frac{1}{5 \times 5}$$

$$T6 = 0,04$$

$$\text{Tingkat Kesalahan } T7 = \frac{\text{Total Kesalahan}}{\text{Kesempatan} \times \text{Total Partisipan}}$$

$$T7 = \frac{4}{4 \times 5}$$

$$T7 = 0,2$$

Adapun rata-rata hasil perhitungan tingkat kesalahan setiap tugas dari salah satu kelas yaitu 1 SD, antara lain sebagai berikut:

$$\text{Rata - Rata} = \frac{\text{Tingkat Kesalahan } (T1 + T2 + T3 + T4 + T5 + T6 + T7)}{7}$$

$$\text{Rata - Rata} = \frac{0 + 0 + 0 + 0,3 + 0,05 + 0,04 + 0,2}{7}$$

$$\text{Rata - Rata} = \frac{0,59}{7}$$

$$\text{Rata - Rata} = 0,0842 \approx 0,08$$

Berdasarkan perhitungan yang dilakukan didapatkan hasil perhitungan tingkat kesalahan setiap tugas dari setiap masing-masing kelas dan rata-rata dari perhitungan tersebut dapat dilihat pada Tabel 6.11. Berdasarkan Tabel 6.11 ditunjukkan bahwa rata-rata dari nilai tingkat kesalahan yang didapatkan adalah kelas 1 SD yaitu 0,08, kelas 2 SD yaitu 0,12, kelas 3 SD yaitu 0,08, kelas 4 SD yaitu



0,02, kelas 5 SD yaitu 0,04, dan kelas 6 SD yaitu 0. Jika diambil rata-rata dari pengujian ini, maka didapatkan bahwa penilaian berdasarkan aspek efektivitas pada aplikasi Reado memiliki nilai rata-rata yaitu 0,06. Dari hasil rata-rata tersebut, maka nilai 0,06 menunjukkan bahwa nilai tersebut berada di bawah rata-rata tingkat kesalahan aplikasi yaitu 0,7, sehingga hal ini menyatakan bahwa hasil pengukuran tingkat kesalahan aplikasi Reado adalah **sangat baik**.

Tabel 6.11 Hasil Nilai Tingkat Kesalahan dari Setiap Kelas

Kelas	Nilai Tingkat Kesalahan Setiap Tugas							Rata-Rata Nilai Tingkat Kesalahan
	T1	T2	T3	T4	T5	T6	T7	
1	0	0	0	0,3	0,05	0,04	0,2	0,08
2	0	0	0	0,15	0,2	0,16	0,3	0,12
3	0,125	0	0	0,125	0	0,2	0,125	0,08
4	0	0	0	0,167	0	0	0	0,02
5	0	0	0	0,083	0	0	0,167	0,04
6	0	0	0	0	0	0	0	0
Rata-Rata								0,0567 $\approx$ 0,06

6.2.4 Aspek Kepuasan Pengguna

Dalam melakukan pengujian pada aspek kepuasan pengguna, pengujian dilakukan dengan menggunakan kuesioner SUS. Dengan melakukan kuesioner SUS kepada 10 subjek penelitian yaitu anak-anak yang mana terdiri atas 10 pernyataan. Di dalam menentukan aspek kepuasan pengguna dibutuhkan data pengujian dengan menggunakan data dari Tabel 6.8. Data pada Tabel 6.8 merupakan data yang menunjukkan hasil dari kuesioner SUS setelah menggunakan aplikasi Reado. Berikut merupakan perhitungan nilai SUS untuk salah satu subjek penelitian yaitu S1 dalam menentukan nilai SUS berdasarkan nilai dari pernyataan kuesioner SUS:

$$\text{Nilai SUS} = \sum_{i=1}^{\text{ganjil}=5} \sum_{j=1}^{\text{genap}=5} (K_{\text{ganjil}=j} - 1) + (5 - K_{\text{genap}=i}) \times 2,5$$

$$\text{Nilai SUS} = ((K1 - 1) + (5 - K2) + (K3 - 1) + (5 - K4) + (K5 - 1) + (5 - K6) + (K7 - 1) + (5 - K8) + (K9 - 1) + (5 - K10)) \times 2,5$$

$$\text{Nilai SUS S1} = ((4 - 1) + (5 - 3) + (5 - 1) + (5 - 3) + (5 - 1) + (5 - 2) + (4 - 1) + (5 - 3) + (5 - 1) + (5 - 5)) \times 2,5$$

$$\text{Nilai SUS S1} = (3 + 2 + 4 + 2 + 4 + 3 + 3 + 2 + 4 + 0) \times 2,5$$

$$\text{Nilai SUS S1} = 27 \times 2,5$$

$$\text{Nilai SUS S1} = 67,5$$



Dari pengujian untuk aspek kepuasan pengguna, digunakan kuesioner SUS untuk menentukan tingkat kegunaan aplikasi dari aspek kepuasan pengguna. Berdasarkan perhitungan yang dilakukan, didapatkan hasil perhitungan dari hasil kuesioner SUS yang dapat dilihat pada Tabel 6.12.

Tabel 6.12 Nilai SUS

Subjek Penelitian	Pernyataan Kuesioner SUS										Nilai SUS
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
S1	7,5	5	10	5	10	7,5	7,5	5	10	0	67,5
S2	10	5	7,5	5	10	5	7,5	10	7,5	10	77,5
S3	10	5	10	7,5	10	7,5	5	10	10	10	85
S4	10	7,5	7,5	5	10	7,5	7,5	7,5	10	7,5	80
S5	5	10	10	7,5	7,5	10	7,5	10	10	7,5	85
S6	5	10	10	5	5	10	10	7,5	7,5	0	70
S7	7,5	5	10	10	10	10	5	7,5	10	7,5	82,5
S8	7,5	5	10	7,5	10	7,5	10	7,5	10	7,5	82,5
S9	10	7,5	10	7,5	10	10	10	7,5	10	10	92,5
S10	7,5	7,5	10	10	10	7,5	10	7,5	10	7,5	87,5
S11	10	7,5	10	10	10	10	10	10	10	0	87,5
S12	7,5	10	10	7,5	7,5	7,5	7,5	10	10	7,5	85
S13	10	10	10	5	10	10	7,5	10	7,5	5	85
S14	7,5	5	10	5	5	5	7,5	7,5	7,5	10	70
S15	10	7,5	10	5	7,5	7,5	10	7,5	7,5	10	82,5
S16	10	7,5	5	5	7,5	7,5	10	7,5	7,5	10	77,5
S17	5	10	10	7,5	10	10	7,5	10	7,5	7,5	85
S18	7,5	7,5	10	7,5	7,5	10	10	7,5	10	7,5	85
S19	0	10	5	10	5	7,5	5	7,5	5	7,5	62,5
S20	5	7,5	10	7,5	5	7,5	5	7,5	5	7,5	67,5
S21	10	10	10	10	10	10	10	10	10	10	100
S22	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	2,5	70
Rata-rata Nilai SUS											75,5

Berdasarkan hasil perhitungan yang sudah dilakukan, didapatkan rata-rata nilai SUS dari setiap masing-masing kelas yang dapat dilihat pada Tabel 6.13. Berdasarkan Tabel 6.13 ditunjukkan bahwa rata-rata dari nilai SUS setiap kelas yang

didapatkan adalah kelas 1 SD yaitu 79, kelas 2 SD yaitu 83, kelas 3 SD yaitu 81,88, kelas 4 SD yaitu 81,67, kelas 5 SD yaitu 71,67, dan kelas 6 SD yaitu 85. Jika diambil rata-rata dari pengujian ini, maka didapatkan bahwa penilaian berdasarkan aspek kepuasan pengguna pada aplikasi Reado memiliki nilai rata-rata yaitu 80,37. Dari hasil rata-rata tersebut, maka nilai 80,37 menunjukkan bahwa nilai tersebut apabila dilihat berdasarkan *adjective rating*, menunjukkan bahwa aspek kepuasan pengguna yaitu melebihi nilai dari 70 dan memiliki *adjective rating* yaitu “baik”.

Tabel 6.13 Hasil Nilai SUS dari Setiap Kelas

Kelas	Jumlah Sampel	Rata-Rata Hasil Nilai SUS dari Setiap Kelas
1	5	79
2	5	83
3	4	81,88
4	3	81,67
5	3	71,67
6	2	85
Rata-Rata	22 Anak	80,37

BAB 7 PENUTUP

7.1 Kesimpulan

Penggunaan metode *design sprint* dalam pengembangan aplikasi Reado terdiri atas beberapa tahapan pengembangan. Tahapan pengembangan ini yaitu tahapan *understand*, *ideate*, *decide*, *prototype* dan *test* yang dilakukan selama 5 hari. Tahapan dilakukan untuk membuat prototipe berdasarkan kebutuhan pengguna. Dari hasil analisis kebutuhan, perancangan, implementasi dan pengujian dengan menggunakan metode *design sprint*, maka dapat diambil kesimpulan bahwa:

1. Tahapan analisis kebutuhan menggunakan metode *design sprint* yaitu tahapan *understand* pada hari pertama. Dari hasil analisis kebutuhan ini menghasilkan 10 kebutuhan fungsional dan 1 kebutuhan non fungsional pada aplikasi Reado. Kebutuhan fungsional yang didapatkan yaitu sistem harus mampu untuk mendeteksi kata, menampilkan kata-kata dari hasil deteksi, menampilkan alfabet dari kata berdasarkan hasil deteksi, menentukan tahapan permainan, menentukan *level* permainan, menampilkan permainan tahapan latihan, menampilkan permainan tahapan tebak kata berdasarkan gambar, menampilkan permainan tahapan tebak kata berdasarkan suara, menampilkan permainan tahapan tebak kata berdasarkan video dan menyediakan mekanisme untuk jawaban dari permainan. Selain itu, kebutuhan non fungsional aplikasi Reado yaitu *usability* atau kegunaan aplikasi.
2. Perancangan yang dilakukan bertujuan untuk menghasilkan prototipe tervalidasi yang menggunakan metode *design sprint* yaitu tahapan *ideate*, *decide*, *prototype* dan *test* pada hari kedua sampai kelima. Perancangan ini terdiri atas 5 perancangan yaitu perancangan arsitektur aplikasi, perancangan diagram, perancangan basis data, perancangan algoritma dan perancangan antarmuka. Perancangan arsitektur aplikasi menghasilkan skema perancangan aplikasi Reado yang menerapkan fitur dari Mobile Vision API dan YouTube Android Player API. Selanjutnya yaitu perancangan diagram menghasilkan 10 perancangan diagram *sequence* berdasarkan jumlah dari kebutuhan fungsional aplikasi Reado dan 1 perancangan diagram *class* yang menggambarkan relasi antar kelas dari aplikasi Reado. Selain itu, perancangan basis data dan algoritma berdasarkan jumlah dari kebutuhan fungsional aplikasi Reado. Terakhir yaitu perancangan antarmuka yang dilakukan untuk menghasilkan prototipe tervalidasi yang terdiri dari 11 perancangan antarmuka yaitu perancangan antarmuka untuk halaman logo aplikasi, halaman utama, halaman deteksi kata, halaman kata, halaman alfabet, halaman tahapan permainan, halaman *level* permainan, halaman tahapan latihan, halaman tebak kata dari gambar, halaman tebak kata dari suara dan halaman tebak kata dari video.
3. Tahapan implementasi menghasilkan implementasi yang terdiri atas implementasi basis data, algoritma dan antarmuka. Implementasi basis data dan algoritma menghasilkan 10 implementasi berdasarkan jumlah dari kebutuhan fungsional aplikasi Reado. Selain itu, implementasi dari antarmuka



menghasilkan 11 implementasi yang terdiri atas implementasi antarmuka untuk halaman logo aplikasi, halaman utama, halaman deteksi kata, halaman kata, halaman alfabet, halaman tahapan permainan, halaman *level* permainan, halaman tahapan latihan, halaman tebak kata dari gambar, halaman tebak kata dari suara dan halaman tebak kata dari video.

4. Dari tahapan pengujian yang dilakukan terdiri atas pengujian *black box* dan pengujian *usability*. Pada pengujian *black box* dengan menggunakan teknik *equivalence partitioning* menghasilkan 10 pengujian dari kebutuhan fungsional aplikasi Reado yang telah lulus pengujian dan sudah terverifikasi. Berdasarkan hasil pengujian *black box* ini didapatkan kesimpulan bahwa aplikasi Reado sudah memenuhi kebutuhan-kebutuhan dari pengguna yang sudah dianalisis sebelumnya. Selain itu, pada pengujian *usability* atau kegunaan aplikasi menghasilkan 3 nilai dari tingkat kegunaan aplikasi yaitu terdiri atas nilai dari aspek efisiensi, efektivitas dan kepuasan pengguna.
5. Dalam menentukan tingkat kegunaan aplikasi Reado, dilakukan pengujian terhadap anak-anak yang terdiri dari 22 partisipan. Partisipan tersebut merupakan anak-anak dari kegiatan sosial Komunitas Katalis Pendidikan dengan tingkatan pendidikan Sekolah Dasar yaitu kelas 1 sampai kelas 6. Tingkat kegunaan aplikasi Reado dengan menggunakan metode *Design Sprint* diukur dari aspek yaitu efisiensi, efektivitas dan kepuasan pengguna. Dari aspek efisiensi yaitu memiliki nilai rata-rata **93,5%** yang mana nilai tersebut masuk ke dalam rentang 90%-100% yaitu **baik** dan menunjukkan bahwa efisiensi dari aplikasi Reado memiliki nilai yang baik. Selain itu, dari aspek efektivitas yaitu **sangat baik** dikarenakan nilai dari aspek efektivitas yaitu **0,06** berada dibawah rata-rata tingkat kesalahan dari aplikasi yaitu 0,7 yang mana hal ini menunjukkan bahwa kesalahan yang dilakukan pengguna saat menggunakan aplikasi masih dalam tahap yang wajar. Selanjutnya, dari aspek kepuasan pengguna yaitu **baik** dengan nilai yang melebihi 70 yaitu **80,37** yang menunjukkan bahwa nilai kepuasan pengguna dari aplikasi Reado yaitu baik. Maka dari itu, dapat disimpulkan bahwa tingkat kegunaan aplikasi Reado didapatkan nilai aspek efisiensi yaitu **baik**, nilai aspek efektivitas yaitu **sangat baik** dan nilai aspek kepuasan pengguna yaitu **baik**.

7.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya agar menghasilkan sistem yang lebih baik, antara lain sebagai berikut:

1. Perhitungan terhadap aspek pengukuran yang digunakan yaitu aspek efisiensi, efektivitas dan kepuasan pengguna untuk mengukur tingkat kegunaan aplikasi didapatkan dari sumber yang berbeda.
2. Penambahan aspek pengukuran lainnya untuk mengukur tingkat kegunaan dari aplikasi.
3. Penggunaan metode pengembangan aplikasi lain yang didapatkan dari sumber berbeda untuk pengembangan aplikasi yang lebih baik.



DAFTAR REFERENSI

Arikunto, S. 2006. *Prosedur Penelitian Suatu Pendekatan Praktek*. Jakarta: Rineka Cipta.

Asosiasi Penyelenggara Jasa Internet Indonesia. 2016. *Jenis Layanan Internet yang Digunakan*. [Online] Tersedia di: <<https://databoks.katadata.co.id/datapublish/2016/10/25/apjii-93-juta-pengguna-internet-pakai-perangkat-mobile>> [Diakses pada 4 November 2019].

Baharrudin, M., Wardani, N.H., Herlambang, A.D. 2018. Analisis SUability pada Sistem Informasi FILKOM Apps Fakultas Ilmu Komputer Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(10), 4179 - 4183.

Bangor, A., Kortum, P. dan Miller, J. 2009. *Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*. *Journal of Usability Studies*, 4(3), 114 – 123.

Brooke, J. 1986. *SUS – A Quick and Dirty Usability Scale*. United Kingdom: Redhatch Consulting Ltd.

Djalal, F. dan Supriadi, D. 2002. *Reformasi Pendidikan Dalam Konteks Otonomi Daerah*. Yogyakarta: Adicita karya Nusa.

EF EPI. 2019. *Indeks Kecakapan Bahasa Inggris EF Peringkat 100 Negara dan Wilayah Menurut Kecakapan Berbahasa Inggris*. Jakarta Pusat: Education First.

Ferreira, V.G. dan Canedo, E.D. 2019. *Using Design Sprint as a Facilitator in Active Learning for Students in The Requirements Engineering Course: an Experience Report*. ACM ISBN 978-1-4503-5933-7/19/04. doi: 10.1145/3297280.3297463.

International Organization for Standardization., 1998. *Guidance on Usability : ISO 9241 -11*.

International Organization for Standardization. 2010. *Ergonomics of Human-System Interaction ISO 9241-210: Human-Centred Design For Interactive Systems*.

Knapp, J., Zeratsky, J. dan Kowitz, B. 2016. *Sprint: How To Solve Big Problems and Test New Ideas in Just Five Days*. Simon and Schuster.

Mayo, M.J. 2007. Games for Science and Engineering Education. *Communications of The ACM*, 50(7): 31-35.

Mifsud, J. 2015. *Usability Metrics-A Guide To Quantify The Usability Of Any System-Usability Geek*. [Online] Tersedia di: <<https://usabilitygeek.com/usabilitymetrics-a-guide-to-quantify-systemusability/>> [Diakses pada 27 Juli 2019].



Mustaqbal, M.S., Firdaus, R.F. dan Rahmadi H. 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 31-36.

Prensky, M. 2003. Digital Game-Based Learning. *ACM Computers in Entertainment*, 1(1), 1-4.

Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach*, McGrawHill, New York.

Quan, Z. 2016. Introducing Mobile DDL (Data-Driven Learning) for Vocabulary Learning: an Experiment for Academic English. *Journal of Computers in Education*, 3(3), 273-287. doi:10.1007/s40692-016-0067-0.

Rabbianty, E.N. 2015. Pembelajaran Bahasa Inggris Pada Anak Usia Dini di Sekolah Alam Excellentia Pamekasan Madura. *Okara*, 2, 84-102.

Rubin, J., dan Chisnell, D. 2008. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests (2nd ed.)*. Indianapolis, IN: Wiley Publishing, Inc.

Rummel, B., 2015. *Quick UX Assessment? Start with System Usability Scale*. [Online] Tersedia di <<https://experience.sap.com/skillup/quick-ux-assessment-start-with-the-system-usability-scale/>> [Diakses 1 Agustus 2019].

Sauro, J. Dan Kindlund, E. 2005. *Making Sense of Usability Metrics: Usability and Six Sigma*. Montreal, Kanada 27 Juni – 1 Juli 2005. USA : UPA (Usability Professionals Association).

Sergeev, A. 2010. *User Interface Design, UX Research and Usability Evaluation*. [Online] Tersedia di: <<http://ui-designer.net/uideigner/usability/satisfaction.html>> [Diakses 27 Juli 2019].

Sugiono. 2007. *Metodologi Penelitian Bisnis*. Jakarta: PT. Gramedia.

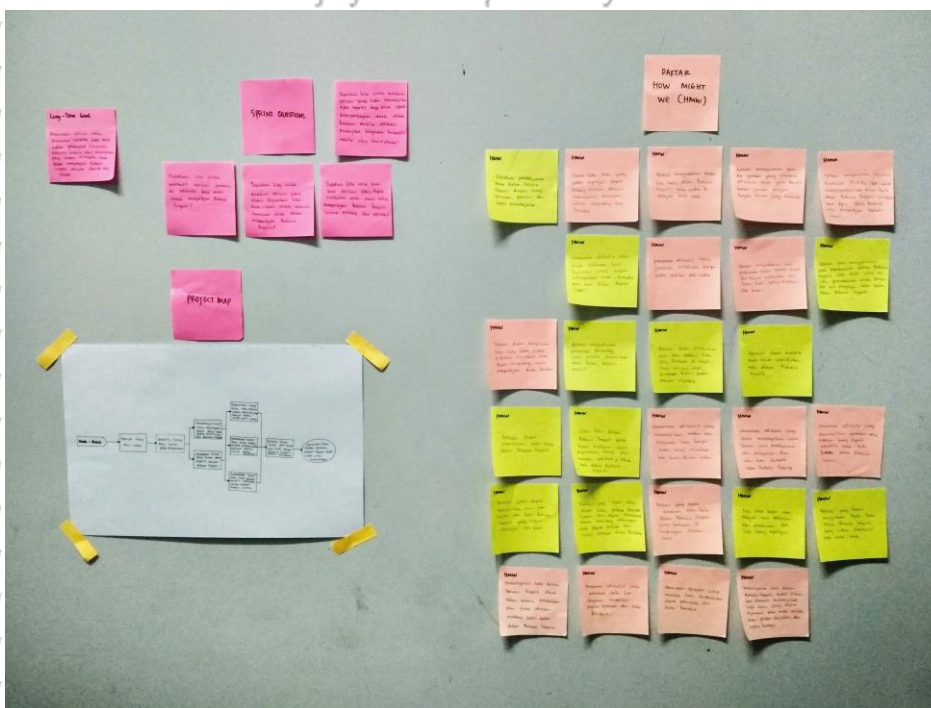
Tolle, H., Pinandito, A., Kharisma A.P. dan Dewi, R.K. 2017. *Pengembangan Aplikasi Perangkat Bergerak*. Malang: Universitas Brawijaya Press.

Umar, H. 2003. *Metodologi Penelitian untuk Skripsi dan Tesis Bisnis*. Depok: PT RajaGrafindo Persada.

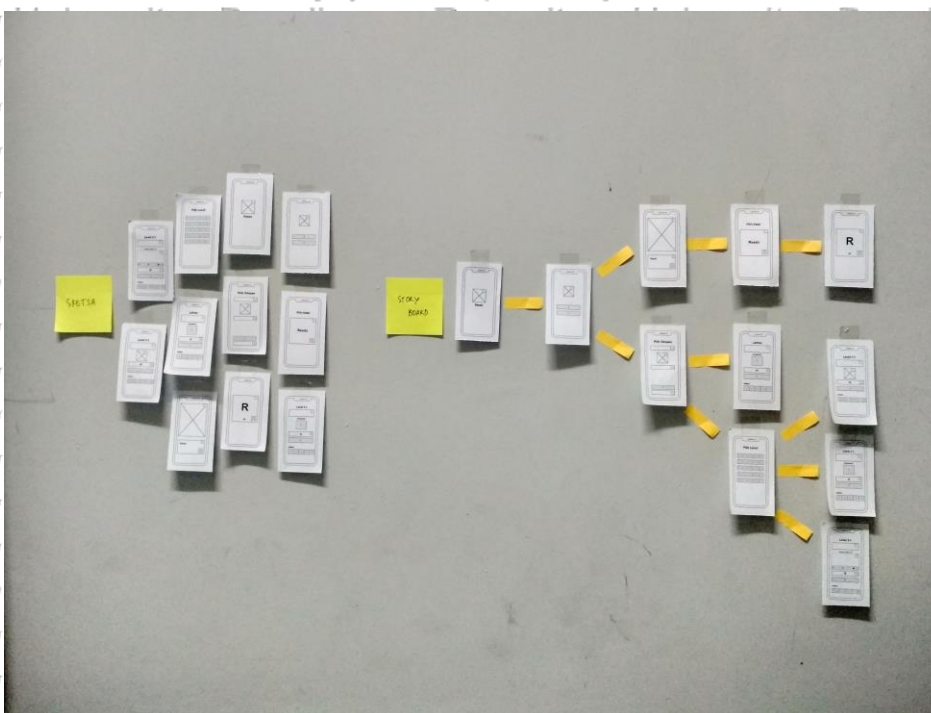
We Are Social. 2017. *Sistem Operasi Perangkat Mobile*. [Online] Tersedia di: <<https://databoks.katadata.co.id/datapublish/2017/10/09/73-perangkat-mobile-global-menggunakan-android>> [Diakses pada 4 November 2019].

LAMPIRAN A PELAKSANAAN PENGEMBANGAN APLIKASI

A.1 Dokumentasi Pelaksanaan Pengembangan Aplikasi Reado untuk Analisis Kebutuhan



A.2 Dokumentasi Pelaksanaan Pengembangan Aplikasi Reado untuk Perancangan





LAMPIRAN B HASIL WAWANCARA

B.1 Hasil Wawancara dari Pengajar yaitu Guru Sekolah

1. Guru Sekolah : Pak Tomi

P : Pembelajaran untuk mata pelajaran Bahasa Inggris sudah diajarkan sejak kelas berapa?

J : Untuk mata pelajaran Bahasa Inggris sudah diajarkan sejak kelas 1 SD dan akan terus diajarkan sampai kelas 6 SD.

P : Apa saja permasalahan yang dihadapi oleh anak-anak di sekolah ketika mempelajari mata pelajaran Bahasa Inggris?

J : Permasalahan yang dihadapi anak yaitu sulit untuk memahami kalimat-kalimat Bahasa Inggris dan sulit untuk membaca kata-kata tertentu dalam Bahasa Inggris. Misalnya di dalam buku terdapat kata yang baru pertama kali dia ketahui, dia tidak tahu cara untuk membacanya. Biasanya permasalahan-permasalahan ini biasa terjadi untuk anak-anak yang duduk di kelas 1 sampai 4 SD dan hanya ada beberapa anak di kelas 5 dan 6 SD yang sulit untuk membaca kata dalam Bahasa Inggris. Hal ini dikarenakan anak kelas 5 dan 6 SD sering mendengar kata-kata Bahasa Inggris dalam permainan atau *game* sehingga tahu cara pembacaannya.

P : Materi pembelajaran seperti apa yang digunakan guru untuk anak-anak dapat mempelajari mata pelajaran Bahasa Inggris?

J : Materi pembelajaran yang diajarkan oleh guru disekolah yaitu melalui buku pelajaran yaitu dengan membaca cerita dalam Bahasa Inggris, menulis kata-kata Bahasa Inggris, dan belajar komunikasi dengan melakukan dialog antar siswa dalam Bahasa Inggris. Semua materi sudah terdapat di dalam buku dari mata pelajaran Bahasa Inggris.

P : Bagaimana metode pengajaran yang diterapkan untuk mengajarkan anak-anak dalam mata pelajaran Bahasa Inggris?

J : Metode pengajaran yang digunakan bukan hanya mengajarkan anak melalui buku namun juga menggunakan gambar-gambar yang dapat merepresentasikan kata-kata dalam Bahasa Inggris. Misalkan gambar-gambar hewan yang dapat merepresentasikan kata-kata hewan dalam Bahasa Inggris. Selain itu, pengajaran dilakukan melalui sebuah permainan seperti lempar bola untuk melanjutkan cerita di dalam Bahasa Inggris. Hal ini sangat membantu anak dalam mempelajari Bahasa Inggris, karena dengan bermain anak-anak dapat lebih cepat untuk mempelajarinya.

P : Apakah pihak sekolah mengizinkan penggunaan perangkat elektronik yang digunakan oleh guru untuk menunjang pembelajaran bagi anak-anak?

J : Ya, pihak sekolah mengizinkan anak-anak dan guru untuk menggunakan perangkat elektronik disekolah yaitu komputer dan *smartphone*.



P : Mengapa pihak sekolah memberikan izin untuk penggunaan perangkat elektronik tersebut dan seperti apa penggunaan dari perangkat elektronik yang digunakan tersebut?

J : Penggunaan perangkat elektronik seperti komputer dan *smartphone* sangat membantu sekali dalam mengajarkan anak. Selain dengan tujuan untuk membangun karakter pelajar yang melek terhadap teknologi, penggunaan perangkat elektronik juga sangat meningkatkan pemahaman anak terhadap pembelajaran karena menggunakan media pembelajaran baru selain buku sehingga membuat anak memiliki rasa penasaran yang tinggi untuk memahami teknologi juga belajar hal yang baru.

P : Bagaimana cara guru dalam mengantisipasi ketergantungan anak untuk menggunakan perangkat elektronik yang disediakan oleh pihak sekolah?

J : Cara antisipasi untuk penggunaan perangkat elektronik yaitu dengan adanya pengawasan langsung dari guru atau pengajar dan anak-anak diberikan waktu tertentu untuk penggunaan perangkat elektronik. Pemberian waktu tertentu dilakukan untuk membuat anak tidak ketergantungan untuk bermain menggunakan perangkat elektronik yang diberikan sehingga hal ini ditujukan juga untuk dapat melatih kedisiplinan anak.

Yang bertanda tangan dibawah ini:

Nama : Bpk. Tomi

Jabatan : Kepala Laboratorium Teknologi Informasi dan Komunikasi (TIK) dan Guru di SDI Moh Hatta

Menyatakan bahwa hasil wawancara diatas dinyatakan benar dan tidak direkayasa :

Yang bertanda tanda tangan,

Bpk. Tomi



B.2 Hasil Wawancara dari Pengajar yaitu Relawan Mengajar

2. Pengajar dari Komunitas : Dhea Paramitha

P : Apa latar belakang dari diadakannya kegiatan sosial ini oleh Komunitas Katalis Pendidikan?

J : Latar belakang dari kegiatan sosial yang diadakan oleh Komunitas Katalis Pendidikan ialah Sosial-Pendidikan. Bukan hanya mengajar namun kita juga bersosial kepada masyarakat sekitar yang membutuhkan.

P : Siapakah yang menjadi target dari diadakannya kegiatan sosial ini?

J : Untuk saat ini, target dari diadakannya kegiatan sosial ini adalah anak-anak yang membutuhkan biaya untuk dapat bersekolah di sekolah yang terbaik. Semua program kerja kami fokuskan untuk mengajar, mendidik keterampilan serta pendidikan karakter dari anak-anak yang membutuhkan untuk dapat bersekolah di sekolah yang bagus. Anak-anak dalam kegiatan sosial ini akan dididik sehingga dapat disekolahkan dengan dana bantuan yang dikumpulkan oleh komunitas.

P : Apa saja mata pelajaran yang diajarkan oleh pengajar dari komunitas untuk anak-anak?

J : Mata pelajaran yang diajarkan sangat banyak. Sama seperti mata pelajaran seperti pada sekolah pada umumnya, seperti matematika, Bahasa Inggris dan lain-lain.

P : Apakah ada mata pelajaran yang sulit untuk dipahami dan dipelajari oleh anak-anak?

J : Tidak terlalu banyak mata pelajaran yang sulit untuk dipahami oleh anak-anak, karena anak-anak dalam kegiatan sosial ini dapat memahami pelajaran dengan cepat. Namun mungkin lebih kepada mata pelajaran Bahasa Inggris yang sulit dipahami anak-anak, karena bukan merupakan bahasa ibu seperti Bahasa Indonesia ataupun Bahasa Jawa sehingga sulit untuk mempelajari dan memahaminya. Karena bahasa pengantar yang digunakan oleh relawan mengajar juga adalah Bahasa Indonesia, sehingga anak-anak tidak terbiasa dengan mata pelajaran Bahasa Inggris.

P : Apa saja permasalahan yang dihadapi oleh anak-anak ketika mempelajari mata pelajaran Bahasa Inggris?

J : Kesulitan yang dihadapi, mungkin lebih kepada sulit untuk membaca kata atau kalimat dalam Bahasa Inggris. Lalu sulit memahami arti dari kalimat dalam Bahasa Inggris.

P : Materi pembelajaran seperti apa yang digunakan oleh pengajar dari komunitas untuk mengajarkan anak-anak mempelajari mata pelajaran Bahasa Inggris?

J : Materi pembelajaran yang digunakan yaitu membaca cerita-cerita seperti dongeng yang dituliskan dalam Bahasa Inggris. Lalu juga, video-video



pembelajaran Bahasa Inggris di Youtube. Serta buku-buku pembelajaran Bahasa Inggris untuk anak-anak dapat membaca dan memahami pelajaran Bahasa Inggris.

P : Bagaimana metode pengajaran yang diterapkan untuk mengajarkan anak-anak dalam mata pelajaran Bahasa Inggris?

J : Metode pengajaran yang diterapkan yaitu metode mengajar belajar dari buku, lalu menonton video, bermain dan pemberian hadiah ketika anak-anak mampu menyelesaikan tugas dengan baik. Pemberian hadiah ini dilakukan untuk dapat menciptakan semangat bagi anak untuk menyelesaikan tugas yang diperintahkan.

P : Apakah pihak komunitas mengizinkan untuk penggunaan perangkat elektronik yang digunakan oleh pengajar untuk menunjang pembelajaran bagi anak-anak?

J : Pihak komunitas tidak mengizinkan pengajar untuk penggunaan perangkat elektronik.

P : Mengapa pihak komunitas tidak memberikan izin untuk penggunaan perangkat elektronik tersebut?

J : Hal ini dikarenakan tidak adanya perangkat elektronik yang cukup dari komunitas, karena dikhawatirkan kekurangan perangkat elektronik dapat menimbulkan masalah bagi anak baik bertengkar karena tidak menggunakan perangkat elektronik yang sama dan masalah lainnya. Sehingga hal ini dilakukan demi terciptanya lingkungan belajar yang kondusif. Maka dari hal ini, kita sangat menginginkan anak-anak dapat bersekolah di sekolah terbaik yaitu SDI Mohammad Hatta yang memiliki fasilitas yang lengkap dan bahkan menerapkan *smart class* untuk anak-anak melalui perangkat elektronik.

Yang bertanda tangan dibawah ini:

Nama : Dhea Paramitha

Jabatan : Pengajar dari Komunitas Katalis Pendidikan

Menyatakan bahwa hasil wawancara diatas dinyatakan benar dan tidak direkayasa:

Yang bertanda tanda tangan,

Dhea Paramitha

LAMPIRAN C PELAKSANAAN PENGUJIAN APLIKASI

C.1 Dokumentasi Pelaksanaan Pengujian *Usability*





LAMPIRAN D SURAT KETERANGAN PENGUJIAN

D.1 Surat Permohonan Data Skripsi



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS ILMU KOMPUTER

Jl. Veteran No.8, Malang, 65145, Indonesia

Telp. : +62-341-577911; Fax : +62-341-577911

http://filkom.ub.ac.id

E-mail : filkom@ub.ac.id

Nomor : 34 /UN10.F15.05/PP/2019

09 JAN 2020

Hal : *Permohonan Data Skripsi*

Yth. Kepala Sekolah SDI Mohammad Hatta
Jl. Flamboyan No.30, Lowokwaru, Kec. Lowokwaru
Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Rachmalia Dewi
NIM : 165150201111236
Judul Skripsi : Pengembangan Aplikasi Reado: Aplikasi Pembelajaran Bahasa Inggris Bagi Anak Pada Kegiatan Sosial Komunitas Katalis Pendidikan Dengan Metode Design Sprint
Dosen Pembimbing : 1. Agi Putra Kharisma, S.T., M.T.
2. Candra Dewi, S.Kom., M.Sc
Prodi : Teknik Informatika

Guna melakukan survey atau pengambilan data untuk skripsi mahasiswa tersebut di Instansi Bapak / Ibu, jenis survey atau data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data : Data Pengujian Usability untuk Penggunaan Aplikasi Reado
Waktu : 10 Januari 2020

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Kepala Tata Usaha



Tembusan:

1. Ketua Program Studi Sistem Informasi
2. Mahasiswa yang bersangkutan

D.2 Surat Permohonan Pengisian Kuesioner

SURAT PERMOHONAN PENGISIAN KUESIONER

Kepada Yth.
Kepala Sekolah
SD ISLAM MOHAMMAD HATTA
Di tempat,

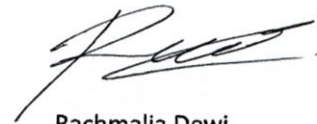
Dengan Hormat,

Sehubungan untuk memenuhi kelengkapan penyusunan skripsi, saya bermaksud untuk melakukan penelitian di SD ISLAM MOHAMMAD HATTA dengan Judul "Pengembangan Aplikasi Reado: Aplikasi Pembelajaran Bahasa Inggris Bagi Anak Pada Kegiatan Sosial Komunitas Katalis Pendidikan Dengan Metode *Design Sprint*". Sebagai salah satu syarat untuk mengikuti ujian sarjana pada Universitas Brawijaya. Maka dengan segala kerendahan hati penulis, memohon kesediaan pihak sekolah untuk mengizinkan Siswa/i SD Islam Mohammad Hatta dalam melakukan pengisian kuesioner yang telah dilampirkan.

Penelitian ini semata-mata bersifat ilmiah, dan hanya dipergunakan untuk keperluan penyusunan skripsi. Di samping itu juga, diharapkan hasil penelitian ini dapat memberikan masukan bagi penulis.

Saya memohon kesediaan Siswa/i dari kegiatan sosial Komunitas Katalis Pendidikan yang bersekolah di SD Islam Mohammad Hatta untuk menjawab semua pertanyaan yang ada secara jujur dan terbuka, mengingat data yang saya perlukan sangat besar sekali artinya. Penelitian menjamin kerahasiaan identitas dan setiap jawaban responden. Atas segala bantuan dan partisipasi yang telah diberikan, saya ucapkan terima kasih.

Hormat Saya,



Rachmalia Dewi

D.3 Surat Keterangan Pelaksanaan Pengujian



YAYASAN BINA INSAN KAMIL [YANAICA] INDONESIA
SD ISLAM MOHAMMAD HATTA

Terakreditasi A

NIS : 100630 – NSS : 102056104009 – NPSN : 20533897

Jl. Simpang Flamboyan no. 30 Malang 65141, Tlp. (0341) 413003

Website : sdimohammadhatta.sch.id e-mail : sdimh@sdimohammadhatta.sch.id



SURAT KETERANGAN

Nomor : 181/S.Ket/KS/SDI-MH/I/2020

Yang bertandatangan di bawah ini :

Nama : Suyanto, S.Pd., M.K.Pd

NIY : 992069022

Jabatan : Kepala SD Islam Mohammad Hatta

Menerangkan dengan sesungguhnya bahwa :

Nama : Rachmalia Dewi

NIM : 165150201111236

Program Studi : Teknik Informatika

Telah melaksanakan penelitian di SD Islam Mohammad Hatta pada tanggal 10 Januari 2020 guna penyusunan skripsi dengan judul *“Pengembangan Aplikasi Reado: Aplikasi Pembelajaran Bahasa Inggris Bagi Anak Pada Kegiatan Sosial Komunitas Katalis Pendidikan Dengan Metode Design Sprint”*

Demikian surat keterangan ini dibuat dengan sebenarnya dan dipergunakan sebagaimana mestinya.

Malang, 04 Jumadhil Akhir 1441 H

29 Januari 2020 M

Kepala Sekolah,



Suyanto, S.Pd., M.KPd

MH Cerdas Berimtaq Sahabat Lingkungan

@sdi_mohammad_hatta

sdi mohammad hatta

081990006100