

**RANCANG BANGUN APLIKASI *MOBILE* ANDROID KAMUS  
BAHASA LAMPUNG BERBASIS KOMUNITAS  
DENGAN METODE *SCRUM***

**(Skripsi)**

**Oleh**

**RIDHO AHMAD FAUZI  
NPM 2115061001**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

**RANCANG BANGUN APLIKASI *MOBILE* ANDROID KAMUS  
BAHASA LAMPUNG BERBASIS KOMUNITAS  
DENGAN METODE *SCRUM***

Oleh

**RIDHO AHMAD FAUZI**

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar  
**SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika  
Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

## **ABSTRAK**

### **RANCANG BANGUN APLIKASI *MOBILE* ANDROID KAMUS BAHASA LAMPUNG BERBASIS KOMUNITAS DENGAN METODE *SCRUM***

**Oleh**

**RIDHO AHMAD FAUZI**

Provinsi Lampung memiliki keberagaman suku yang menetap di dalamnya. Sayangnya hal tersebut membawa dampak juga terhadap bahasa Lampung. Saat ini bahasa Lampung terbatas penggunaannya dan bahasa lain menjadi lebih dominan. Sehingga diperlukan revitalisasi bahasa dengan mendokumentasikan aspek bahasa seperti penulisan kamus. Penelitian ini bertujuan untuk merancang aplikasi Android kamus bahasa Lampung berbasis komunitas yang dapat melestarikan bahasa Lampung. Aplikasi ini dikembangkan menggunakan bahasa pemrograman React Native dengan kerangka kerja Expo dan menggunakan *Scrum* sebagai metode pengembangan perangkat lunak. Aplikasi diuji menggunakan *Black Box* dan *User Acceptance Test (UAT)*. Hasil penelitian menunjukkan bahwa aplikasi berhasil dikembangkan menggunakan metode *Scrum* dalam 9 *sprint* dengan total 19 *product backlog*. Hasil pengujian menunjukkan seluruh fungsionalitas berjalan sesuai harapan dan dapat diterima oleh pengguna, dengan keberhasilan pengujian *Black Box* mencapai 100% dan rata-rata nilai UAT sebesar 95,02% (4,75 dari 5) dari tiga kelompok pengguna.

Kata kunci: Bahasa Lampung, *Blacbox*, React Native Expo, *Scrum*, *User Acceptance Test (UAT)*

## **ABSTRACT**

### ***DESIGN AND DEVELOPMENT OF A COMMUNITY-BASED ANDROID MOBILE APPLICATION FOR A LAMPUNG LANGUAGE DICTIONARY USING THE SCRUM METHOD***

*By*

**RIDHO AHMAD FAUZI**

*The province of Lampung is home to a diverse population of ethnic groups. Unfortunately, this diversity has impacted the use of the Lampung language. Currently, the use of the Lampung language is limited, while other languages have become more dominant. Therefore, language revitalization is necessary by documenting linguistic aspects such as compiling dictionaries. This study aims to design a community-based Android dictionary application for the Lampung language to support its preservation. The application was developed using the React Native programming language with the Expo framework and employed the Scrum method for software development. The application was tested using Black Box and User Acceptance Testing (UAT). The results showed that the application was successfully developed using the Scrum method over nine sprints with a total of 19 product backlogs. Testing results indicated that all functionalities worked as expected and were well received by users, with Black Box testing achieving 100% success and UAT yielding an average score of 95.02% (4.75 out of 5) from three user groups.*

*Keywords: Blacbox, Lampung language, React Native Expo, Scrum, User Acceptance Test (UAT)*

Judul Skripsi : **RANCANG BANGUN APLIKASI  
MOBILE ANDROID KAMUS BAHASA  
LAMPUNG BERBASIS KOMUNITAS  
DENGAN METODE SCRUM**

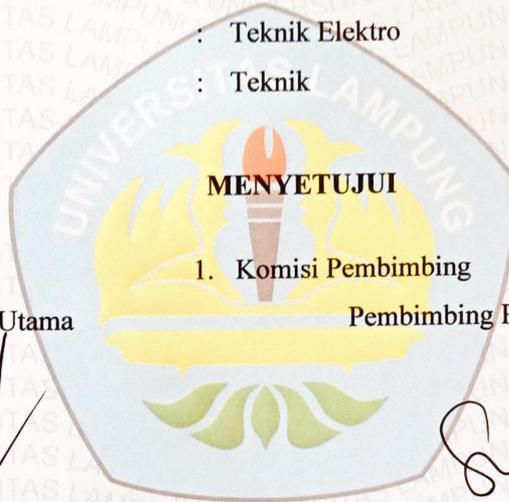
Nama Mahasiswa : **Ridho Ahmad Fauzi**

Nomor Pokok Mahasiswa : 2115061001

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

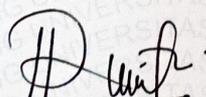
  
**Mahendra Pratama, S.T., M. Eng.**  
NIP 199112152019031013

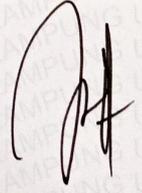
  
**Ir. Trisya Septiana, S.T., M.T., IPM**  
NIP 199009212019032025

2. Mengetahui

Ketua Jurusan  
Teknik Elektro

Ketua Program Studi  
Teknik Informatika

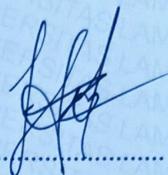
  
**Herlinawati, S.T., M.T.**  
NIP 197103141999032001

  
**Yessi Mulyani, S.T., M.T.**  
NIP 197312262000122001

## MENGESAHKAN

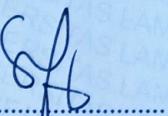
1. Tim Penguji

Ketua : **Mahendra Pratama, S.T., M. Eng.**



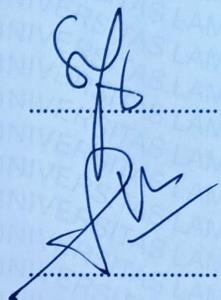
.....

Sekretaris : **Ir. Trisya Septiana, S.T., M.T., IPM**

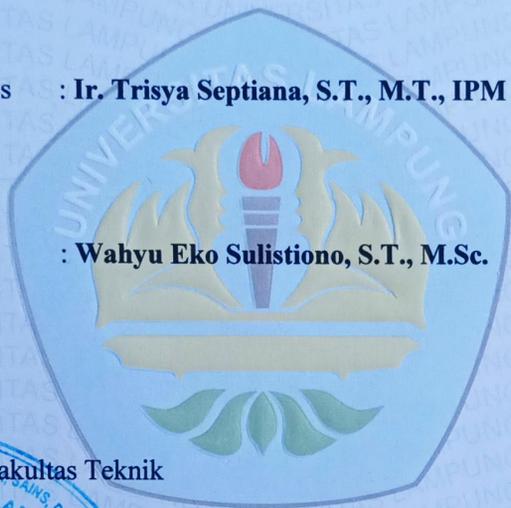


.....

Penguji : **Wahyu Eko Sulistiono, S.T., M.Sc.**



.....



2. Dekan Fakultas Teknik



**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.**

NIP 197509282001121002

Tanggal Lulus Ujian Skripsi: **2 Juni 2025**

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul “Rancang Bangun Aplikasi *Mobile* Android Kamus Bahasa Lampung Berbasis Komunitas Dengan Metode *Scrum*” sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 18 Juni 2025

Penulis,



Ridho Ahmad Fauzi

NPM. 2115061001

## RIWAYAT HIDUP



Penulis dilahirkan di Gisting pada tanggal 14 Juli 2003 sebagai anak sulung dari dua bersaudara, dari pasangan Bapak Lardi dan Ibu Yuntizar. Penulis menyelesaikan jenjang pendidikan formal di SDS Muhammadiyah 35 Solear pada tahun 2015, kemudian melanjutkan ke SMPN 1 Cisoka dan lulus pada tahun 2018, serta menamatkan pendidikan menengah atas di SMAN 1 Kabupaten Tangerang pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur seleksi SNPMTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti kegiatan Magang Bersertifikat dari Kementerian Pendidikan dan Budaya di mitra PT Suitmedia Kreasi Indonesia sebagai *Frontend Web Developer* pada tahun 2024.
2. Mengikuti kegiatan Studi Independen Bersertifikat dari Kementerian Pendidikan dan Budaya di mitra Dicoding Indonesia pada tahun 2023.
3. Berperan sebagai mentor dalam program Studi Independen Bersertifikat Batch 7 di mitra Dicoding Indonesia.
4. Berperan sebagai mentor dalam program *Coding Camp Powered By DBS Foundation 2025* yang diselenggarakan oleh Dicoding Indonesia dan DBS Foundation.
5. Berperan sebagai Asisten Laboratorium Teknik Komputer tahun akademik 2023/2024.

## **MOTTO**

*“The only way to do great work is to love what you do”*

**(Steve Jobs)**

*“Don’t end the book early because of a bad page”*

**(Nelson Mandela)**

*“Don’t end the book early because of a bad page”*

**(James)**

*“Teruslah bernafas walau kau tahu akan mati”*

**(Penulis)**

## **PERSEMBAHAN**

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, saya sebagai penulis dapat menyelesaikan penelitian dan skripsi ini.

Kupersembahkan skripsi ini kepada:

### **Kedua Orang Tua**

“Kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, dan nasihat kepada penulis dalam proses penyelesaian skripsi ini”

## SANWACANA

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, serta hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Rancang Bangun Aplikasi *Mobile* Android Kamus Bahasa Lampung Berbasis Komunitas dengan Metode *Scrum*”. Dalam proses penyusunan skripsi ini, penulis banyak menerima dukungan, bimbingan, serta bantuan baik secara moril maupun materil dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih yang mendalam, penulis menyampaikan penghargaan sebesar-besarnya kepada:

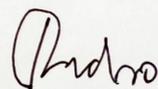
1. Kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, dan nasihat kepada penulis sepanjang masa perkuliahan;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung yang telah membantu proses kelancaran pengerjaan penelitian;
5. Bapak Mahendra Pratama, S.T., M. Eng. selaku Pembimbing Utama yang telah membimbing dan memberikan saran kepada penulis selama mengerjakan penelitian dan penulisan skripsi;
6. Ibu Ir. Trisya Septiana, S.T., M.T., IPM. selaku Pembimbing Pendamping yang telah membimbing dan memberikan saran bagi penulis dalam menyelesaikan penelitian dan penulisan skripsi;
7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc. selaku Penguji yang telah memberikan berbagai saran dan masukan yang membangun terhadap penelitian ini;

8. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM. selaku dosen pembimbing akademik yang selalu memberikan dukungan dan bimbingan supaya menjadi lebih dalam perkuliahan;
9. Bapak Dr. Munaris, M.Pd. dan Ibu Jesika Wulandari, S. Pd. sebagai dosen Prodi Pendidikan Bahasa Lampung yang telah bersedia meluangkan waktu untuk menjadi penguji aplikasi;
10. Seluruh Dosen Program Studi Teknik Informatika Universitas Lampung yang telah banyak memberikan ilmu kepada penulis.
11. Andre Gilang Firmansyah, Dede Kurniawan, dan Nyoman Eka Swardita sebagai rekan satu tim dalam penelitian yang telah bekerja sama dan saling mendukung dalam penyelesaian pengembangan aplikasi;
12. Rekan-rekan yang tidak dapat disebutkan satu per satu yang telah bersedia meluangkan waktu untuk menjadi penguji aplikasi;
13. Rekan-rekan dalam grup 'Rido Jamet', yaitu Muhkito, Agustin, Nabila, Afif, dan Gibran, yang telah memberikan dukungan besar kepada penulis dalam menyelesaikan skripsi;
14. Rekan-rekan dalam grup 'Teh Kotak' yang tidak dapat disebutkan satu persatu, yang telah memberikan dukungan besar kepada penulis dalam menyelesaikan skripsi;
15. Seluruh pihak yang terlibat dalam penelitian ini yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 18 Juni 2025

Penulis,



Ridho Ahmad Fauzi

## DAFTAR ISI

	Halaman
<b>ABSTRAK .....</b>	<b>i</b>
<b>PERSEMBAHAN.....</b>	<b>viii</b>
<b>SANWACANA .....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvi</b>
<b>DAFTAR TABEL .....</b>	<b>xx</b>
<b>I. PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Tujuan Penelitian.....	4
1.4    Manfaat Penelitian .....	4
1.5    Batasan Masalah.....	4
1.6    Sistematika Penulisan .....	5
<b>II. TINJAUAN PUSTAKA .....</b>	<b>6</b>
2.1    Dasar Teori .....	6
2.1.1    Bahasa Lampung.....	6
2.1.2    Revitalisasi Bahasa.....	6
2.1.3 <i>Scrum</i> .....	8
2.1.4 <i>Unified Modeling Language (UML)</i> .....	11
2.1.5 <i>Black Box Testing</i> .....	12

2.1.6	<i>User Acceptance Test (UAT)</i> .....	12
2.1.7	Visual Studio Code.....	13
2.1.8	Git.....	13
2.1.9	GitLab .....	13
2.1.10	Android .....	14
2.1.11	Node JS .....	14
2.1.12	React Native .....	14
2.1.13	React Native Expo.....	16
2.1.14	<i>Application Programming Interface (API)</i> .....	16
2.1.15	Google Play Store .....	17
2.1.16	Taiga.....	17
2.2	Penelitian Terkait.....	18
2.2.1	Cawa Lampung : Kamus Bahasa Indonesia Lampung Dialek A Berbasis Android.....	18
2.2.2	Rancang Bangun Aplikasi Kamus Bahasa dan Aksara Lampung Dialek A dan Dialek O Berbasis Android .....	18
2.2.3	Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan <i>Prototype</i> (Studi Kasus : Toko Uda Fajri) .....	19
2.2.4	Aplikasi Manajemen Surat Masuk dan Surat Keluar Berbasis <i>Mobile</i> Di Perum Bulog Subdivre .....	19
2.2.5	Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode <i>Scrum</i> pada Aplikasi Presensi Karyawan Glints Academy .....	20
2.2.6	Penerapan Metode <i>SCRUM</i> dalam Pengembangan Sistem Informasi Layanan Kawasan .....	20
2.2.7	<i>Comparative Analysis Of Scrum And Kanban Methodologies In Agile Software Development</i> .....	20

2.2.8	<i>A Comparative Analysis of Native vs React Native Mobile App Development</i> .....	21
2.2.9	<i>Software Testing with the approach of Black Box Testing on the Academic Information System</i> .....	21
2.2.10	<i>Performing User Acceptance Test with System Usability Scale for Graduation Application</i> .....	22
<b>III. METODOLOGI PENELITIAN</b> .....		<b>23</b>
3.1	Waktu dan Tempat Penelitian.....	23
3.2	Alat Penelitian.....	25
3.3	Struktur Tim Kamus Bahasa Lampung .....	26
3.4	Tahapan Penelitian .....	27
3.4.1	Analisis Kebutuhan .....	28
3.4.2	<i>Product Backlog</i> .....	53
3.4.3	<i>Sprint Planning</i> .....	53
3.4.4	<i>Sprint</i> .....	53
3.4.5	<i>Daily Scrum</i> .....	54
3.4.6	<i>Sprint Review</i> .....	54
3.4.7	<i>Retrospective</i> .....	54
3.4.8	<i>User Acceptance Test (UAT)</i> .....	54
<b>IV. HASIL DAN PEMBAHASAN</b> .....		<b>56</b>
4.1	<i>Product Backlog</i> .....	56
4.2	<i>Sprint 1</i> .....	58
4.2.1	<i>Sprint Planning</i> .....	58
4.2.2	<i>Sprint</i> .....	58
4.2.3	<i>Sprint Review</i> .....	59
4.2.4	<i>Retrospective</i> .....	59
4.3	<i>Sprint 2</i> .....	59

4.3.1	<i>Sprint Planning</i> .....	59
4.3.2	<i>Sprint</i> .....	59
4.3.3	<i>Sprint Review</i> .....	61
4.3.4	<i>Retrospective</i> .....	61
4.4	<i>Sprint 3</i> .....	61
4.4.1	<i>Sprint Planning</i> .....	61
4.4.2	<i>Sprint</i> .....	62
4.4.3	<i>Sprint Review</i> .....	65
4.4.4	<i>Retrospective</i> .....	65
4.5	<i>Sprint 4</i> .....	65
4.5.1	<i>Sprint Planning</i> .....	65
4.5.2	<i>Sprint</i> .....	66
4.5.3	<i>Sprint Review</i> .....	73
4.5.4	<i>Retrospective</i> .....	73
4.6	<i>Sprint 5</i> .....	74
4.6.1	<i>Sprint Planning</i> .....	74
4.6.2	<i>Sprint</i> .....	74
4.6.3	<i>Sprint Review</i> .....	79
4.6.4	<i>Retrospective</i> .....	79
4.7	<i>Sprint 6</i> .....	79
4.7.1	<i>Sprint Planning</i> .....	79
4.7.2	<i>Sprint</i> .....	81
4.7.3	<i>Sprint Review</i> .....	93
4.7.4	<i>Retrospective</i> .....	93
4.8	<i>Sprint 7</i> .....	94
4.8.1	<i>Sprint Planning</i> .....	94

4.8.2	<i>Sprint</i> .....	95
4.8.3	<i>Sprint Review</i> .....	113
4.8.4	<i>Retrospective</i> .....	114
4.9	<i>Sprint 8</i> .....	114
4.9.1	<i>Sprint Planning</i> .....	114
4.9.2	<i>Sprint</i> .....	115
4.9.3	<i>Sprint Review</i> .....	120
4.9.4	<i>Retrospective</i> .....	120
4.10	<i>Sprint 9</i> .....	120
4.10.1	<i>Sprint Planning</i> .....	120
4.10.2	<i>Sprint</i> .....	120
4.10.3	<i>Sprint Review</i> .....	122
4.10.4	<i>Retrospective</i> .....	123
4.11	<i>User Acceptance Test (UAT)</i> .....	123
4.12	Perilisan .....	133
<b>V. KESIMPULAN DAN SARAN</b> .....		<b>136</b>
5.1	Kesimpulan .....	136
5.2	Saran .....	136
<b>DAFTAR PUSTAKA</b> .....		<b>137</b>
<b>LAMPIRAN</b> .....		<b>142</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Alur Kerangka Kerja <i>Scrum</i> .....	8
Gambar 2.2 Arsitektur Aplikasi React Native .....	15
Gambar 3.1 Struktur Tim Kamus Bahasa Lampung .....	26
Gambar 3.2 Tahapan Penelitian .....	27
Gambar 3.3 <i>Use Case</i> .....	34
Gambar 3.4 <i>Activity Diagram</i> daftar .....	37
Gambar 3.5 <i>Activity Diagram</i> masuk.....	38
Gambar 3.6 <i>Activity Diagram</i> lupa kata sandi .....	39
Gambar 3.7 <i>Activity Diagram</i> menerjemahkan kata dan pelaporan .....	40
Gambar 3.8 <i>Activity Diagram</i> melihat blog .....	41
Gambar 3.9 <i>Activity Diagram</i> menambahkan kata baru .....	42
Gambar 3.10 <i>Activity Diagram</i> mengelola draf.....	43
Gambar 3.11 <i>Activity Diagram</i> merevisi kata .....	44
Gambar 3.12 <i>Activity Diagram</i> melihat riwayat penambahan kata .....	45
Gambar 3.13 <i>Activity Diagram</i> tinjau kata (menerima).....	46
Gambar 3.14 <i>Activity Diagram</i> tinjau kata (menolak).....	47
Gambar 3.15 <i>Activity Diagram</i> tinjau kata (merevisi).....	48
Gambar 3.16 <i>Activity Diagram</i> tinjau laporan .....	49
Gambar 3.17 <i>Activity Diagram</i> melihat riwayat tinjau kata .....	50
Gambar 3.18 <i>Activity Diagram</i> melihat riwayat tinjau laporan .....	50
Gambar 3.19 <i>Activity Diagram</i> sunting profil .....	51
Gambar 3.20 <i>Activity Diagram</i> atur ulang kata sandi .....	52
Gambar 3.21 Arsitektur Sistem .....	52
Gambar 4.1 Inisiasi awal proyek.....	58
Gambar 4.2 Slicing section pencarian kata .....	60

Gambar 4.3 Hasil implementasi section pencarian kata dan bar navigasi .....	60
Gambar 4.4 Hasil implementasi section banner CTA .....	62
Gambar 4.5 Hasil implementasi section statistics.....	62
Gambar 4.6 Hasil implementasi section leaderboard.....	63
Gambar 4.7 Hasil implementasi section blog .....	63
Gambar 4.8 Slicing halaman profil .....	64
Gambar 4.9 Hasil implementasi halaman profil .....	64
Gambar 4.10 Hasil implementasi halaman dasbor.....	66
Gambar 4.11 Hasil implementasi halaman profil pengguna .....	67
Gambar 4.12 Slicing halaman edit profil .....	67
Gambar 4.13 Hasil implementasi halaman edit profil .....	68
Gambar 4.14 Slicing halaman kontribusi.....	68
Gambar 4.15 Hasil implementasi halaman kontribusi .....	69
Gambar 4.16 Slicing halaman daftar riwayat kontribusi .....	69
Gambar 4.17 Hasil implementasi halaman riwayat kontribusi .....	70
Gambar 4.18 Hasil implementasi halaman login .....	71
Gambar 4.19 Hasil implementasi halaman register .....	71
Gambar 4.20 Hasil implementasi halaman laporan .....	72
Gambar 4.21 Hasil implementasi halaman blog .....	72
Gambar 4.22 Slicing halaman detail blog.....	72
Gambar 4.23 Hasil implementasi halaman detail blog .....	73
Gambar 4.24 Hasil implementasi halaman daftar tinjau kata .....	75
Gambar 4.25 Slicing halaman detail tinjau kata .....	75
Gambar 4.26 Hasil implementasi halaman detail tinjau kata.....	76
Gambar 4.27 Hasil implementasi halaman konfirmasi email .....	76
Gambar 4.28 Slicing halaman konfirmasi OTP .....	77
Gambar 4.29 Hasil implementasi halaman konfirmasi OTP .....	77
Gambar 4.30 Hasil implementasi halaman kata sandi baru .....	78
Gambar 4.31 Hasil implementasi halaman daftar draf.....	78
Gambar 4.32 Hasil implementasi halaman detail draf.....	79
Gambar 4.33 Kode program fungsi login .....	81
Gambar 4.34 Slicing integrasi API halaman login.....	82

Gambar 4.35 Kode program fungsi register.....	82
Gambar 4.36 Slicing integrasi API halaman register .....	83
Gambar 4.37 Kode program fungsi komunikasi API metode get .....	83
Gambar 4.38 Kode program fungsi komunikasi API metode post .....	84
Gambar 4.39 Kode program fungsi komunikasi API metode put.....	85
Gambar 4.40 Kode program fungsi komunikasi API metode delete .....	85
Gambar 4.41 Kode program pemanggilan fungsi komunikasi API metode get ...	86
Gambar 4.42 Kode program fungsi untuk data komponen dropdown.....	86
Gambar 4.43 Integrasi API halaman profil .....	87
Gambar 4.44 Integrasi API halaman daftar blog.....	87
Gambar 4.45 Integrasi API halaman detail blog .....	88
Gambar 4.46 Integrasi API halaman konfirmasi email.....	89
Gambar 4.47 Integrasi API halaman konfirmasi OTP .....	89
Gambar 4.48 Integrasi API halaman kata sandi baru.....	90
Gambar 4.49 Hasil implementasi halaman daftar riwayat tinjau kata .....	90
Gambar 4.50 Slicing halaman detail riwayat tinjau kata .....	91
Gambar 4.51 Hasil Implementasi halaman detail riwayat tinjau kata .....	91
Gambar 4.52 Integrasi API halaman detail riwayat tinjau kata .....	92
Gambar 4.53 Integrasi API section terjemah kata.....	93
Gambar 4.54 Integrasi API section leaderboard .....	96
Gambar 4.55 Integrasi API halaman kontribusi.....	96
Gambar 4.56 Integrasi API daftar tinjau kata.....	97
Gambar 4.57 Integrasi API detail tinjau kata .....	98
Gambar 4.58 Integrasi API halaman daftar draf .....	98
Gambar 4.59 Integrasi API halaman detail draf.....	99
Gambar 4.60 Hasil Implementasi halaman ubah draf.....	100
Gambar 4.61 Integrasi API halaman update draf.....	100
Gambar 4.62 Hasil Implementasi halaman daftar kata direvisi .....	101
Gambar 4.63 Integrasi API halaman daftar revisi .....	101
Gambar 4.64 Hasil Implementasi halaman detail revisi .....	102
Gambar 4.65 Hasil Implementasi halaman detail revisi .....	103
Gambar 4.66 Integrasi API halaman revisi .....	103

Gambar 4.67 Integrasi API halaman daftar riwayat kontribusi.....	104
Gambar 4.68 Integrasi API halaman detail riwayat kontribusi .....	105
Gambar 4.69 Hasil Implementasi halaman daftar kata sedang ditinjau.....	105
Gambar 4.70 Integrasi API daftar kata sedang ditinjau .....	106
Gambar 4.71 Hasil Implementasi halaman detail kata sedang ditinjau .....	107
Gambar 4.72 Integrasi API halaman daftar sedang ditinjau .....	107
Gambar 4.73 Hasil Implementasi halaman FAQ .....	108
Gambar 4.74 Hasil Implementasi halaman kredit.....	108
Gambar 4.75 Hasil Implementasi halaman tentang aplikasi.....	108
Gambar 4.76 Hasil Implementasi halaman syarat dan ketentuan .....	109
Gambar 4.77 Integrasi API halaman detail profil .....	109
Gambar 4.78 Integrasi API halaman update profil.....	110
Gambar 4.79 Integrasi API halaman ubah kata sandi .....	110
Gambar 4.80 Hasil implementasi halaman notifikasi .....	111
Gambar 4.81 Integrasi API halaman daftar notifikasi.....	111
Gambar 4.82 Integrasi API push notification.....	112
Gambar 4.83 Hasil Implementasi halaman daftar kata .....	112
Gambar 4.84 Integrasi API halaman daftar kata .....	113
Gambar 4.85 Integrasi API laporkan kata .....	115
Gambar 4.86 Integrasi API halaman daftar laporan.....	115
Gambar 4.87 Hasil Implementasi halaman detail laporan .....	116
Gambar 4.88 Integrasi API halaman detail laporan .....	117
Gambar 4.89 Hasil Implementasi halaman daftar riwayat tinjau laporan.....	117
Gambar 4.90 Integrasi API halaman riwayat tinjau laporan .....	118
Gambar 4.91 Hasil Implementasi halaman detail riwayat tinjau laporan .....	119
Gambar 4.92 Integrasi API halaman detail riwayat tinjau laporan .....	119
Gambar 4.93 Penyiapan informasi aplikasi .....	133
Gambar 4.94 Penyiapan pengujian tertutup .....	133
Gambar 4.95 Penyiapan rilis aplikasi ke produksi.....	134
Gambar 4.96 Rilis aplikasi ke produksi .....	135
Gambar 4.97 Hasil rilis aplikasi di Play Store .....	135

## DAFTAR TABEL

	Halaman
Tabel 2.1 Elemen antarmuka pengguna pada berbagai <i>platform</i> .....	16
Tabel 3.1 Waktu Penelitian.....	23
Tabel 3.2 Alat Penelitian .....	25
Tabel 3.3 Perbandingan dengan Aplikasi Serupa .....	30
Tabel 3.4 Kebutuhan Fungsional.....	31
Tabel 3.5 Kebutuhan Non-fungsional .....	33
Tabel 3.6 Definisi Aktor <i>Use Case</i> .....	35
Tabel 3.7 Definisi <i>Use Case</i> .....	35
Tabel 4.1 <i>User stories</i> .....	56
Tabel 4.2 <i>Sprint backlog</i> dan <i>task sprint</i> kedua .....	58
Tabel 4.3 <i>Sprint backlog</i> dan <i>task sprint</i> kedua .....	59
Tabel 4.4 <i>Sprint backlog</i> dan <i>task sprint</i> ketiga .....	61
Tabel 4.5 <i>Sprint backlog</i> dan <i>task sprint</i> keempat .....	65
Tabel 4.6 <i>Sprint backlog</i> dan <i>task sprint</i> kelima .....	74
Tabel 4.7 <i>Sprint backlog</i> dan <i>task sprint</i> keenam .....	80
Tabel 4.8 <i>Sprint backlog</i> dan <i>task sprint</i> ketujuh.....	94
Tabel 4.9 <i>Sprint backlog</i> dan <i>task sprint</i> kedelapan.....	114
Tabel 4.10 <i>Sprint backlog</i> dan <i>task sprint</i> kesembilan.....	120
Tabel 4.11 Hasil pengujian <i>Black Box</i> pertama.....	121
Tabel 4.12 Hasil pengujian <i>Black Box</i> kedua.....	121
Tabel 4.13 Bobot penilaian pengguna .....	124
Tabel 4.14 Kategori interpretasi skor .....	124
Tabel 4.15 Hasil kuesioner peran peninjau .....	125
Tabel 4.16 Hasil kuesioner peran kontributor .....	127
Tabel 4.17 Hasil kuesioner peran publik.....	130
Tabel 4.18 Hasil akumulasi akhir UAT .....	132

## I. PENDAHULUAN

### 1.1 Latar Belakang

Indonesia adalah negara kepulauan yang terdiri dari 17.504 pulau besar maupun kecil dan dikenal sebagai negara multikultural dengan keberagaman suku, agama, ras, budaya, dan bahasa [1]. Keberagaman ini menjadikan Indonesia memiliki 1.340 suku bangsa yang tersebar di 38 provinsi [2]. Umumnya, setiap suku bangsa tersebut memiliki bahasa daerah tersendiri yang digunakan sebagai sarana komunikasi sekaligus menjadi identitas khas dari daerah tersebut. Berdasarkan data dari Badan Pengembangan dan Pembinaan Bahasa pada tahun 2019, terdapat sekitar 801 bahasa daerah yang digunakan di seluruh Indonesia [3].

Dari sisi jumlah, penutur bahasa-bahasa daerah di Nusantara sangat beragam. Beberapa bahasa daerah masih memiliki penutur yang cukup banyak, di sisi lain ada bahasa-bahasa daerah hanya memiliki sedikit penutur dan dapat dianggap sebagai bahasa minoritas, bahkan sedang berada di ambang kepunahan [4]. Kepunahan bahasa daerah di Nusantara disebabkan oleh berbagai faktor, seperti dominasi bahasa mayoritas di wilayah penggunaannya, masyarakat penutur yang bilingual atau bahkan multilingual, pengaruh globalisasi, perpindahan penduduk atau migrasi, perkawinan antar suku, bencana alam dan musibah, kurangnya apresiasi terhadap bahasa etnik sendiri, rendahnya penggunaan bahasa daerah dalam berbagai aspek kehidupan, dan faktor ekonomi [4].

Provinsi Lampung ialah satu di antara provinsi di Indonesia dengan lokasi yang sangat strategis. Lokasinya yang terletak di ujung selatan Pulau Sumatera menjadikan Lampung sebagai gerbang utama bagi siapa pun yang ingin memasuki Pulau Sumatera. Kondisi ini menjadikan Lampung ramai didatangi oleh pendatang dari berbagai suku. Keberagaman ini tidak hanya memperkaya komposisi

masyarakat di Lampung, tetapi juga membawa dampak terhadap masyarakat asli Lampung. Salah satu dampaknya adalah memudarnya penggunaan bahasa Lampung di kalangan masyarakat lokal, karena bahasa Indonesia digunakan oleh pendatang untuk berkomunikasi dengan penutur asli bahasa Lampung, sehingga bahasa Indonesia menjadi dominan dalam masyarakat Lampung [5].

Bahasa Lampung merupakan bahasa daerah sekaligus bahasa ibu bagi masyarakat di Provinsi Lampung. Bahasa ini memiliki dua dialek utama, yaitu Lampung Api yang dikenal sebagai dialek A atau Pesisir dan Lampung Nyo yang dikenal sebagai dialek O atau Abung [6]. Saat ini, bahasa Lampung menghadapi permasalahan berupa penurunan penggunaan dalam komunikasi sehari-hari oleh penuturnya. Menurut Badan Pengembangan dan Pembinaan Bahasa, bahasa Lampung termasuk dalam kategori rentan, yang menunjukkan bahwa jumlah penuturnya semakin sedikit [7]. Saat ini, bahasa Lampung terbatas penggunaannya hanya dalam konteks keluarga, di kampung-kampung penduduk asli, dan saat permusyawaratan adat. Bahkan, bahasa Lampung belum berhasil beradaptasi dengan perkembangan teknologi, karena perangkat lunak kamus bahasa Lampung belum menerima kontribusi data yang cukup dari masyarakat penggunanya [5].

Salah satu upaya dalam melakukan revitalisasi bahasa Lampung agar tetap lestari adalah dengan mendokumentasikan aspek bahasa seperti penulisan kamus [8]. Pemanfaatan teknologi, seperti perangkat lunak dan aplikasi digital, dapat mempermudah dalam upaya memperluas cakupan penggunaan bahasa daerah. Penggunaan teknologi ini dapat menjadi solusi untuk mendukung kelestarian bahasa, dengan mengembangkan aplikasi digital seperti kamus berbasis Android.

Saat ini sudah terdapat beberapa aplikasi kamus digital bahasa Lampung yang dapat digunakan, salah satunya adalah aplikasi bernama Kamus Bahasa Lampung, yang sudah memiliki banyak pengguna. Namun, aplikasi tersebut masih memiliki beberapa keterbatasan yang perlu diperbaiki. Salah satu keterbatasannya adalah pada alur kontribusi penambahan kata. Ketika pengguna mengajukan kata untuk dimasukkan ke dalam *database* aplikasi, mereka tidak dapat terlibat langsung dalam proses tersebut. Pengguna hanya dapat memberikan saran penambahan kata tanpa mengetahui atau berpartisipasi dalam proses selanjutnya.

Oleh karena itu, aplikasi kamus bahasa Lampung berbasis Android yang dirancang dalam penelitian ini diharapkan mampu menjawab keterbatasan yang telah disampaikan sebelumnya. Aplikasi ini dirancang dengan basis komunitas, yang melibatkan masyarakat luas, baik masyarakat Lampung maupun individu yang memiliki pengetahuan tentang bahasa Lampung. Kolaborasi dengan mereka diharapkan dapat meningkatkan akurasi kontribusi data dalam aplikasi. Pada aplikasi ini pengguna berkesempatan untuk terlibat langsung dalam proses tersebut, mulai dari penambahan, pembaruan, hingga verifikasi entri kamus. Dengan demikian, pengguna tidak hanya menjadi pemakai, tetapi juga berperan aktif dalam pengembangan dan penyempurnaan *database* aplikasi.

Selain fitur kontribusi, aplikasi ini juga dilengkapi dengan fitur utama lainnya, yaitu fitur pencarian kata. Fitur ini dirancang untuk memudahkan pengguna dalam menemukan arti kata dalam bahasa Lampung. Dengan antarmuka yang intuitif, pengguna dapat mencari kata menggunakan bahasa Indonesia atau bahasa Lampung. Fitur ini tidak hanya membantu pengguna memahami arti kata, tetapi juga menyediakan informasi tambahan, seperti dialek, lokasi penutur, penulisan dalam aksara, dan contoh penggunaan dalam kalimat. Dengan adanya fitur ini, diharapkan pengalaman belajar bahasa Lampung menjadi lebih interaktif, efektif, dan menyenangkan bagi pengguna dari berbagai kalangan. Harapannya aplikasi yang dirancang pada penelitian ini dapat menyediakan akses mudah dan cepat bagi masyarakat dari berbagai kalangan, untuk mempelajari bahasa Lampung. Serta dapat mendukung upaya pelestarian dan pendokumentasian Bahasa Lampung sebagai warisan budaya Indonesia.

## **1.2 Rumusan Masalah**

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara merancang aplikasi kamus bahasa Lampung berbasis komunitas yang dapat melestarikan bahasa Lampung?
2. Apakah aplikasi kamus bahasa Lampung yang dirancang dapat berfungsi dengan baik dan sesuai dengan kebutuhan pengguna?

### **1.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang aplikasi kamus bahasa Lampung berbasis komunitas yang dapat melestarikan bahasa Lampung.
2. Menguji fungsionalitas aplikasi kamus bahasa Lampung supaya dapat berfungsi dengan baik dan sesuai dengan kebutuhan pengguna.

### **1.4 Manfaat Penelitian**

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Penelitian ini berkontribusi pada pelestarian dan dokumentasi bahasa Lampung.
2. Hasil penelitian ini dapat dijadikan acuan untuk penelitian lain yang sejenis di masa mendatang.
3. Penelitian ini bisa menjadi sumber pembelajaran yang bermanfaat bagi masyarakat luas.

### **1.5 Batasan Masalah**

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Penelitian ini berfokus pada pengembangan *frontend* aplikasi Anrdoid.
2. Penelitian ini menggunakan *framework* React Native Expo untuk pengembangan *frontend*.
3. Penelitian ini berfokus pada implementasi fitur-fitur untuk pengguna dengan peran publik, kontributor, dan peninjau. Sehingga, fitur pengelolaan kata, blog, pengguna, dialek, dan wilayah tidak menjadi bagian dari implementasi aplikasi dalam penelitian ini.
4. Sasaran penelitian pengembangan *website* ini adalah penutur asli bahasa Lampung serta masyarakat umum yang tertarik mempelajari bahasa tersebut.

## 1.6 Sistematika Penulisan

### BAB I: PENDAHULUAN

Bab ini berisi mengenai latar belakang penelitian yang menjelaskan dasar pengembangan aplikasi kamus bahasa Lampung berbasis komunitas, tujuan penelitian, rumusan masalah, manfaat penelitian, batasan masalah, dan sistematika penulisan.

### BAB II: TINJAUAN PUSTAKA

Bab ini memuat dasar-dasar teori yang menjadi bahan referensi penelitian ini dan penelitian sebelumnya yang berkaitan dengan penelitian ini.

### BAB III: METODE PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan yaitu metode *Scrum*. Bab ini berisi mengenai waktu penelitian, tempat penelitian, alat penelitian serta metode dan tahap yang dilakukan dalam penelitian ini.

### BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi hasil rancangan dari sistem dengan tahapan metode *Scrum* yaitu *sprint planning*, *sprint*, *daily scrum*, *sprint review*, dan *retrospective*. Serta, penjelasan hasil pengujian aplikasi yang telah dirancang.

### BAB V: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan serta berisi saran yang dapat digunakan sebagai acuan untuk penelitian selanjutnya di masa yang akan datang.

## II. TINJAUAN PUSTAKA

### 2.1 Dasar Teori

#### 2.1.1 Bahasa Lampung

Bahasa Lampung merupakan bahasa daerah sekaligus bahasa ibu bagi masyarakat di Provinsi Lampung. Bahasa Lampung sendiri merupakan salah satu bahasa asli lokal di Indonesia yang memiliki beberapa fungsi strategis bagi masyarakat Lampung, seperti simbol kebanggaan dan identitas, serta sarana komunikasi dalam keluarga dan masyarakat Lampung. Bahasa Lampung termasuk ke dalam rumpun *Malayo-Polynesian*. Bahasa ini memiliki dua dialek utama, yaitu Lampung Api yang dikenal sebagai dialek A atau Pesisir dan Lampung Nyo yang dikenal sebagai dialek O atau Abung. Bahasa Lampung Api dituturkan oleh 827.000 penutur asli dari beberapa wilayah, seperti Sekala Brak, Melinting-Maringgai, Pesisir Rajabasa, Pesisir Teluk, Pesisir Semaka, Pesisir Krui, Belalau, Ranau, Komering, Kayu Agung, Way Kanan, Sungkai, dan Pubian. Sedangkan, bahasa Lampung Nyo dituturkan oleh 180.000 penutur asli dari berbagai wilayah, seperti Abung, Sukadana, Menggala/Tulang Bawang, dan Tulang Bawang Barat. Sehingga, jika diakumulasikan terdapat sekitar 1 juta penutur asli bahasa Lampung [6]. Perbedaan utama antara kedua dialek ini terletak pada fonem akhir kata. Dalam dialek Abung, kata-kata biasanya berakhir dengan fonem /o/, sedangkan dalam dialek Pesisir, fonem akhir tersebut berubah menjadi /a/. Sebagai contoh, kata "dijo" yang berarti "di sini" dalam dialek Abung diucapkan sebagai "dija" dalam dialek Pesisir [9].

#### 2.1.2 Revitalisasi Bahasa

Revitalisasi bahasa adalah sebuah upaya mempertahankan suatu bahasa supaya tetap digunakan dan dihargai, utamanya sebagai kekhasan kelompok dalam komunitas penuturnya. Hal ini dilakukan melalui pendidikan, karya sastra, media

massa, dan berbagai cara lainnya. Revitalisasi bahasa diharapkan mampu secara efektif mempertahankan dan menghidupkan kembali keberadaan bahasa minoritas dalam berbagai aspek kehidupan masyarakat penuturnya atau komunitas asli tempat bahasa tersebut berkembang. Salah satu upaya yang dapat dilakukan untuk revitalisasi bahasa adalah dengan mendokumentasikan seluruh aspek bahasa melalui pengkajian pemetaan, penulisan kamus, pembakuan, pembukuan tata bahasa, penulisan ensiklopedia, serta pencatatan kosakata khazanah budaya [8].

Suatu bahasa dapat dikatakan memiliki vitalitas yang tinggi apabila memiliki banyak penutur dan variasi bahasa tersebut digunakan secara luas. Karakteristik ini menunjukkan bahwa bahasa tersebut cenderung bertahan dan diwariskan dari satu generasi ke generasi berikutnya. Perkembangan pesat teknologi informasi telah mengubah cara berkomunikasi sebagian besar orang, terutama mereka yang tinggal di perkotaan [3].

Berikut ini adalah enam kategori yang digunakan oleh Badan Pengembangan dan Pembinaan Bahasa untuk menentukan status vitalitas bahasa daerah [3].

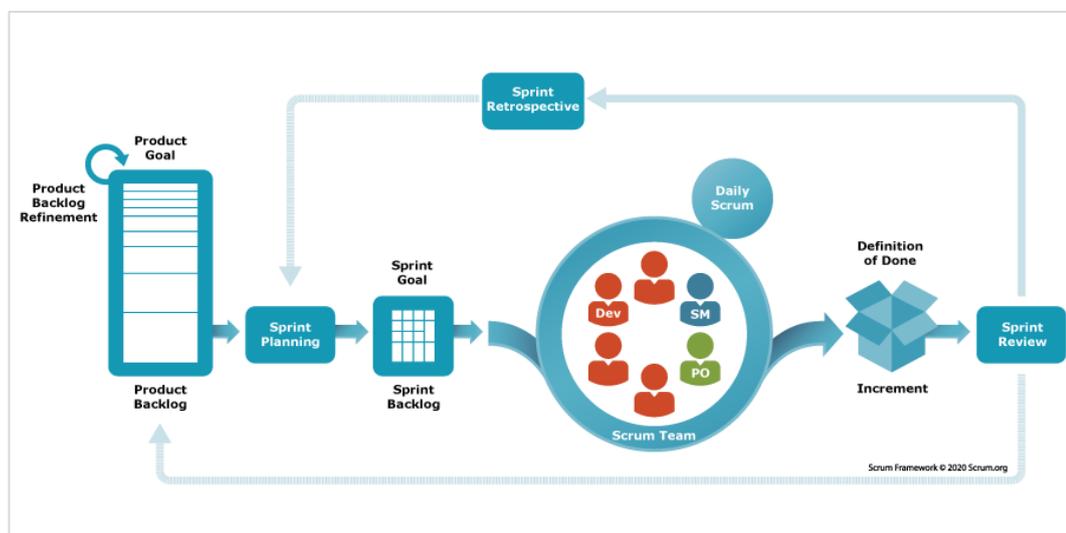
1. Aman, bahasa dikatakan aman jika masih digunakan oleh semua anak dan seluruh anggota kelompok etnis tersebut.
2. Rentan, apabila basa masih digunakan oleh anak-anak dan orang tua, tetapi jumlah penutur sedikit.
3. Mengalami kemunduran, apabila sebagian penutur anak-anak dan kaum tua dan sebagian anak-anak lain tidak menggunakan.
4. Terancam punah, apabila hanya digunakan oleh penutur berusia di atas 20 tahun dan jumlahnya penuturnya sedikit, serta orang-orang yang lebih tua tidak lagi berkomunikasi menggunakan bahasa tersebut dengan anak-anak atau di antara mereka sendiri.
5. Kritis, apabila penuturnya hanya digunakan oleh penutur berusia di atas 40 tahun dan jumlahnya sangat sedikit.
6. Punah, apabila bahasa tidak ada lagi penuturnya.

Menurut Badan Pengembangan dan Pembinaan Bahasa, bahasa Lampung termasuk dalam kategori rentan, yang menunjukkan bahwa jumlah penuturnya semakin

sedikit [7]. Sehingga perlu dilakukan upaya untuk melakukan revitalisasi bahasa Lampung agar tetap lestari.

### 2.1.3 Scrum

*Scrum* adalah metodologi atau kerangka kerja yang terstruktur untuk mengelola pengembangan produk yang kompleks. Kerangka kerja ini menggunakan pendekatan yang berulang dan bertahap untuk mengoptimalkan kondisi yang tidak pasti dan mengendalikan resiko. Tim *scrum* terdiri dari tiga peran utama, yaitu *product owner*, *scrum master*, dan *developer*. Tiga pilar utama *scrum* adalah transparansi, inspeksi, dan adaptasi, yang mendorong tim untuk bekerja secara iteratif, terus-menerus mengevaluasi dan menyempurnakan setiap proses. Transparansi berarti semua aspek penting dari proses harus terlihat oleh pihak yang bertanggung jawab. Inspeksi dilakukan secara berkala untuk memeriksa artefak *scrum* dan perkembangan menuju *sprint goal*, dengan tujuan mendeteksi perbedaan hasil yang tidak diinginkan. Adaptasi terjadi ketika ada penyimpangan dalam proses yang mengakibatkan produk tidak dapat diterima, sehingga proses harus segera disesuaikan untuk meminimalkan kesalahan lebih lanjut [10][11].



Gambar 2.1 Alur Kerangka Kerja Scrum [11]

Pada kerangka kerja *scrum*, terdapat lima *scrum event* yang harus dilakukan dalam setiap iterasi. *Scrum events* tersebut meliputi *sprint*, *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective* [11].

### a. *Sprint*

*Sprint* dapat dikatakan sebagai jantung dalam *scrum*, karena pada *sprint* ide akan diubah menjadi nilai. *Sprint* berlangsung selama satu bulan atau kurang untuk menciptakan konsistensi. Hal-hal yang diperlukan untuk mencapai *product goal*, seperti *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective*, terjadi dalam *sprint*. Terdapat beberapa hal yang perlu diperhatikan selama *sprint* berlangsung meliputi:

1. Selama *sprint*, tidak boleh membuat perubahan besar yang dapat mengganggu tujuan *sprint* yang telah ditetapkan sebelumnya.
2. Kualitas produk atau fitur yang dikembangkan harus tetap terjaga selama proses *sprint*.
3. Jika diperlukan, daftar prioritas pekerjaan atau *product backlog* dapat disempurnakan atau diperbarui selama *sprint*.
4. Jika menemukan hal-hal baru atau kesulitan yang sebelumnya tidak terduga, maka dapat memperjelas atau mengubah lingkup pekerjaan, seperti menambah atau mengurangi fitur yang dikerjakan. Namun, perubahan ini harus dirundingkan dan disetujui oleh *product owner*.

### b. *Sprint Planning*

*Sprint planning* adalah awal yang dilakukan pada setiap *sprint*. Dalam tahapan ini tim *scrum* menentukan apa yang akan dikerjakan selama *sprint*. Pada *sprint planning*, hal-hal yang dilakukan adalah menetapkan *sprint goal* dan memilih *item* dari *product backlog* yang akan dikerjakan.

### c. *Daily Scrum*

*Daily scrum* adalah pertemuan harian yang diadakan untuk mengetahui kemajuan pada *sprint goal*. Selama *daily scrum*, pengembang membahas kemajuan yang telah dicapai pada hari sebelumnya, mengidentifikasi hambatan, dan merencanakan pekerjaan untuk dilakukan pada hari berjalan.

#### d. *Sprint Review*

*Sprint review* adalah pertemuan di akhir *sprint* untuk mengevaluasi hasil kerja dan menentukan langkah berikutnya. Tim *scrum* mempresentasikan hasilnya kepada pemangku kepentingan dan mendiskusikan kemajuan yang telah dilakukan terhadap *product goal*. Dalam pertemuan ini, juga dinilai apa yang telah dicapai dan perubahan yang terjadi.

#### e. *Sprint Retrospective*

*Sprint retrospective* adalah pertemuan di akhir *sprint* untuk mengevaluasi dan merencanakan cara untuk meningkatkan kualitas dan efektivitas tim. Tim *scrum* mendiskusikan apa yang telah berjalan dengan baik, masalah apa yang dihadapi, dan bagaimana menyelesaikan masalah tersebut.

Pada *scrum* juga terdapat *scrum artifacts*. *Scrum artifacts* adalah dokumen yang diperlukan dan dihasilkan selama proses pengembangan produk. *Scrum artifacts* berfungsi sebagai dokumentasi yang mencakup apa yang telah selesai, sedang dikerjakan, dan rencana yang akan dilakukan selanjutnya [11]. Berikut adalah daftar *scrum artifacts* dalam *scrum*:

##### a. *Product Backlog*

*Product backlog* adalah daftar prioritas berisi hal yang diperlukan untuk untuk mengembangkan, memperbaiki, atau memperbarui produk. Setiap *item* dalam *product backlog* dapat diperinci menjadi bagian-bagian lebih kecil melalui proses penyempurnaan (*refinement*) untuk mempermudah dalam memahami pekerjaan yang perlu dilakukan dan apa yang harus dikerjakan berikutnya.

##### b. *Product Goal*

*Product goal* adalah tujuan akhir produk yang memberikan gambaran yang ideal dari produk yang ingin dicapai oleh tim *scrum*. Dengan *product goal*, maka akan membantu tim *scrum* dalam merencanakan pekerjaan dan bertindak. Pada setiap *sprint* tim *scrum* akan meninjau apakah setiap *increment* yang dihasilkan sesuai dengan *product goal*.

c. *Sprint Backlog*

*Sprint backlog* adalah rencana kerja yang disusun oleh tim *scrum* untuk mencapai *sprint goal* dalam suatu *sprint*. *Sprint backlog* berisi pekerjaan yang akan dilakukan selama *sprint* untuk mencapai *sprint goal*. Tujuan dari *sprint backlog* adalah untuk memberikan informasi yang cukup bagi tim *scrum* dalam memonitoring kemajuan yang telah dilakukan.

d. *Sprint Goal*

*Sprint goal* adalah tujuan utama yang ingin dicapai dalam suatu *sprint*. *Sprint goal* ditetapkan pada saat *sprint planning*. Dengan *sprint goal* dapat membantu tim *scrum* untuk tetap berfokus pada tujuan yang sama sepanjang *sprint*.

e. *Increment*

*Increment* dalam Scrum adalah bagian produk yang telah selesai dan telah melalui proses peninjauan untuk memastikan hasil produk telah memenuhi standar yang disepakati (*Definition of Done*). Dalam suatu *sprint*, tim *scrum* bisa menghasilkan lebih dari satu *increment* produk yang sudah selesai dan siap digunakan. Hasil *increment* nantinya juga dipresentasikan pada *sprint review* atau kepada pemangku kepentingan sebelum *sprint* berakhir.

f. *Definition of Done* (DoD)

*Definition of Done* adalah kondisi yang harus dipenuhi agar sebuah *increment* dianggap selesai dan sesuai dengan *product goal*. DoD diperlukan untuk memastikan kualitas produk.

#### **2.1.4 *Unified Modeling Language* (UML)**

*Unified Modeling Language* (UML) adalah metode pemodelan visual yang digunakan dalam perancangan dan pengembangan perangkat lunak berbasis objek. UML berfungsi sebagai standar penulisan, seperti cetak biru, yang mencakup proses bisnis dan penulisan kelas dalam bahasa pemrograman tertentu [12]. Ada beberapa diagram UML yang umum digunakan dalam pengembangan sistem, di antaranya:

- a. *Use Case Diagram* merupakan diagram yang menggambarkan fungsionalitas yang diharapkan dari sistem serta interaksi antara aktor (manusia atau sistem lain) dan sistem itu sendiri.
- b. *Activity Diagram* merupakan diagram yang menunjukkan aliran aktivitas dalam sistem yang sedang berjalan.

### **2.1.5 Black Box Testing**

*Black Box* adalah salah satu metode pengujian perangkat lunak untuk menguji fungsionalitas perangkat lunak. Tujuan *Black Box* adalah menguji aplikasi berdasarkan *input* yang diberikan dan memeriksa apakah *output* yang dihasilkan sesuai dengan yang diharapkan. Pada pengujian ini, penguji tidak perlu mengetahui cara kerja internal atau kode sumber dari perangkat lunak yang sedang diuji. Penguji hanya berfokus pada *input* yang diberikan dan *output* yang dihasilkan, serta memastikan perangkat lunak yang diuji berfungsi sesuai dengan spesifikasi dan persyaratan yang telah ditentukan. Kelebihan *Black Box* adalah dalam proses pengujian tidak memerlukan pemahaman teknis yang mendalam, sehingga penguji tidak perlu memiliki pengetahuan pemrograman. Selain itu, metode ini juga efektif ketika digunakan pada sistem yang lebih besar karena fokus pengujian kepada fungsionalitas yang terlihat [13].

### **2.1.6 User Acceptance Test (UAT)**

User Acceptance Testing (UAT) adalah tahap pengujian perangkat lunak yang dilakukan oleh seseorang yang akan menggunakan suatu sistem (*end user*) untuk mendapatkan keputusan apakah sistem yang dikembangkan telah menjawab kebutuhan pengguna dan dapat memenuhi semua skenario bisnis. Hasil pengujian UAT dapat diperoleh melalui pengumpulan data menggunakan instrumen kuesioner. Penggunaan kuesioner bertujuan untuk mengumpulkan informasi secara lebih mendetail, serta memberikan kenyamanan kepada responden sehingga mereka tidak merasa khawatir memberikan jawaban yang mungkin tidak sesuai dengan ekspektasi. Selain itu, skala *likert* digunakan sebagai metode untuk mengukur pendapat atau persepsi individu [14].

### 2.1.7 Visual Studio Code

Visual Studio Code (VSCode) adalah editor kode sumber yang ringan dan *powerfull* yang dikembangkan oleh Microsoft. Alat ini tersedia untuk sistem operasi Windows, macOS, dan Linux. VSCode memiliki fitur-fitur mumpuni yang dapat membantu dalam penulisan kode sumber, seperti kemampuan *debugging* yang baik, integrasi dengan sistem kontrol versi seperti Git, terminal terintegrasi, *intellisense* untuk *autocompletion* kode, dan penyorotan sintaks. Dengan VSCode pengembang juga dapat menciptakan lingkungan kerja yang nyaman, karena tampilan dan konfigurasi dapat disesuaikan dengan preferensi pribadi, hal ini didukung dengan VSCode yang memiliki kemampuan untuk kustomisasi ekstensi dan tema [15].

### 2.1.8 Git

Git adalah sistem kontrol versi (*Version Control System*) yang digunakan untuk melacak perubahan dalam *file* dan berkolaborasi dalam pengembangan proyek perangkat lunak. Dengan Git memungkinkan pengembang untuk berkolaborasi secara efisien. Git dapat mencatat setiap perubahan yang dilakukan pada kode sumber, dokumen, atau *file* lain-lain [16].

### 2.1.9 GitLab

GitLab adalah *platform DevOps all-in-one* yang menawarkan berbagai fitur yang dapat mempermudah pengembangan perangkat lunak, seperti manajemen kode sumber, integrasi dan pengiriman berkelanjutan (CI/CD), keamanan, dan berbagai fitur lainnya. GitLab memungkinkan tim pengembang untuk mengelola dan melacak perubahan pada kode sumber dengan menggunakan sistem kontrol versi. Dengan begitu setiap perubahan akan tercatat, pesan perubahan, detail perubahan, serta informasi tentang siapa yang membuat perubahan dan kapan terjadi perubahan tersebut [17].

### **2.1.10 Android**

Android adalah sistem operasi berbasis Linux yang dirancang khusus untuk perangkat seluler seperti ponsel dan *tab*. Sistem ini menyediakan *platform* terbuka yang memungkinkan pengembang untuk membuat dan mendistribusikan aplikasi mereka melalui Google Play Store. Android menawarkan antarmuka pengguna yang dapat disesuaikan dan berbagai fitur seperti notifikasi, *multitasking*, serta dukungan untuk berbagai perangkat keras. Sistem ini awalnya dikembangkan oleh Android Inc, sebuah perusahaan teknologi yang kemudian diakuisisi oleh Google Inc pada tahun 2005. Sejak saat itu, Google telah memimpin pengembangan Android, menjadikannya salah satu sistem operasi *mobile* yang paling banyak digunakan di dunia, dengan ekosistem yang luas dan beragam [18].

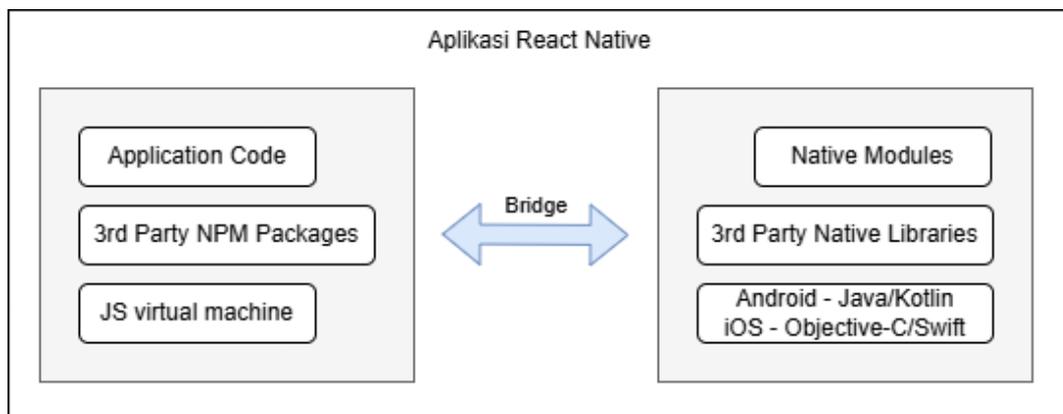
### **2.1.11 Node JS**

Node JS adalah *platform* yang memungkinkan menjalankan JavaScript di sisi *server*. *Platform* ini dirancang untuk mempermudah dalam membangun berbagai jenis sistem. Node JS menggunakan teknologi asinkron sehingga sistem berjalan dengan baik bahkan saat menerima banyak permintaan. Node JS juga dilengkapi dengan *Node Package Manager* (NPM), yaitu manajer paket yang menyediakan ribuan pustaka dan modul. Sehingga, memudahkan pengembang untuk menambahkan berbagai fitur dan mengelola dependensi dalam proyek pengembangan aplikasi [19].

### **2.1.12 React Native**

React Native adalah kerangka kerja sumber terbuka untuk mengembangkan aplikasi *mobile* yang dapat berjalan di berbagai *platform* seperti Android dan iOS. Sehingga, memungkinkan untuk membangun aplikasi iOS dan Android dengan basis kode yang sama. Kerangka kerja ini menggunakan pustaka React untuk membangun antarmuka pengguna. React menggunakan JavaScript XML (JSX), yaitu sintaksis yang memungkinkan penulisan elemen antarmuka pengguna dalam bentuk XML yang disematkan di dalam kode JavaScript [20].

Arsitektur aplikasi React Native terdiri dari *javascript virtual machine*, *React Native bridge*, dan *native modules*. *Javascript virtual machine* adalah lingkungan dimana kode aplikasi React Native ditulis dan dieksekusi. Semua logika aplikasi, seperti pengelolaan data, alur kerja, dan interaksi antarmuka pengguna, dijalankan di *javascript virtual machine*. *React native bridge* adalah jembatan komunikasi antara JavaScript dengan *platform native*, karena JavaScript tidak dapat secara langsung berinteraksi dengan elemen *native*. *Bridge* ini akan menerjemahkan instruksi JavaScript ke dalam perintah yang dapat dipahami oleh *platform native*. Selain itu, jika ada data atau hasil yang perlu dikembalikan dari *native modules* ke kode JavaScript, maka *bridge* juga akan mengembalikan informasi tersebut ke JavaScript. *Native modules* adalah bagian dari aplikasi yang dibuat menggunakan bahasa pemrograman *native* dari *platform* tersebut, seperti Java untuk Android atau Objective-C/Swift untuk iOS. Modul ini digunakan untuk memanfaatkan fitur-fitur spesifik *platform*, seperti kamera, lokasi, atau notifikasi, yang tidak bisa diakses langsung oleh JavaScript [21].



Gambar 2.2 Arsitektur Aplikasi *React Native* [21]

Komponen adalah elemen penyusun utama dalam aplikasi React Native. Komponen pada React Native adalah bagian dari aplikasi yang menggambarkan tampilan atau elemen dalam antarmuka pengguna, seperti tombol, gambar, atau formulir, yang didefinisikan dengan cara deklaratif. Deklaratif berarti bahwa komponen mendefinisikan apa yang ingin ditampilkan pada antarmuka pengguna, bukan bagaimana cara menampilkan atau membangun elemen tersebut. Dalam React Native, komponen bersifat modular, yang memungkinkan penggunaan ulang kode secara efisien. Nantinya, komponen ini dikompilasi menjadi elemen *native*

yang spesifik untuk *platform* tujuan. Selama proses pengembangan, dalam pembuatan komponen juga dapat memanfaatkan API *native* untuk berinteraksi dengan fitur perangkat keras atau sistem operasi, seperti kamera, GPS, atau notifikasi. Sehingga, aplikasi terlihat dan berfungsi seperti aplikasi *native* di *platform* iOS dan Android [22].

Tabel 2.1 Elemen antarmuka pengguna pada berbagai *platform* [22]

Web (React JS)	React Native	Android	iOS
<input>	<TextInput>	<i>EditText</i>	<i>UITextField</i>
<p>	<Text>	<i>TextView</i>	<i>UITextView</i>
<div>	<View>	<i>Andoid.view</i>	<i>UIView</i>

### 2.1.13 React Native Expo

Expo menawarkan beragam alat dan layanan yang memungkinkan pengembang untuk membangun, menguji, dan meluncurkan aplikasi di berbagai *platform* menggunakan satu basis kode yang sama. Dengan Expo, pengembangan aplikasi menjadi lebih efisien karena pengembang dapat memulai proyek tanpa perlu konfigurasi yang rumit, serta menghindari konfigurasi alat seperti Android Studio atau Xcode.

Expo juga menyediakan berbagai komponen siap pakai, seperti kamera, notifikasi, dan *geolocation*, serta mendukung integrasi dengan berbagai *Application Programming Interface* (API) yang biasa digunakan dalam aplikasi *mobile*. Salah satu fitur unggulan Expo adalah *Expo Go*, yang memungkinkan pengembang untuk melihat perubahan secara langsung di perangkat atau *emulator* tanpa harus melakukan *build* atau *setup* tambahan [23].

### 2.1.14 *Application Programming Interface* (API)

*Application Programming Interface* (API) adalah sistem yang memungkinkan satu sistem berkomunikasi dan bertukar data dengan sistem lain. API berfungsi sebagai perantara yang menerima permintaan dari pengguna, meneruskannya ke *server*, dan mengembalikan respon dari *server* kepada pengguna. Di dalam API terdapat

sekumpulan perintah atau fungsi yang dapat digunakan untuk mengakses layanan tertentu tanpa harus mengetahui detail teknis dari sistem tersebut [24][25]. Pada perancangan proyek ini akan menggunakan API yang telah dirancang oleh tim *backend*. Setiap fungsi dan data yang diperlukan akan diakses *frontend* melalui *endpoint* yang telah dibuat oleh tim *backend*.

#### **2.1.15 Google Play Store**

Google Play Store adalah platform daring tempat pengguna mencari dan mengunduh berbagai aplikasi, game, dan konten lainnya. Saat ini Google Play Store telah mempublikasi lebih dari 2 juta aplikasi dan game. Serta, memiliki miliaran pengguna di seluruh dunia. Dengan *platform* ini, pengguna atau konsumen dapat memperoleh aplikasi yang diinginkan dengan aman. Karena Google Play memberikan perlindungan bawaan dan mewajibkan developer untuk mengikuti standar keamanan yang tinggi [26].

#### **2.1.16 Taiga**

Taiga adalah perangkat lunak manajemen proyek yang dapat mempermudah dalam menggunakan metodologi *agile*, seperti *Scrum* dan *Kanban*. Taiga pertama kali diluncurkan pada tahun 2015. Dengan Taiga, tim pengembangan proyek dapat mengelola proyek dengan lebih efisien karena perangkat lunak ini dilengkapi dengan fitur-fitur manajemen proyek yang komprehensif dan mudah digunakan. Untuk mempermudah pengembangan proyek dengan metode *scrum*, Taiga menyediakan fitur diantaranya [27]:

- a. *Scrum Backlog*, memungkinkan untuk memetakan pekerjaan yang perlu dilakukan dan mengatur tugas-tugas berdasarkan prioritas.
- b. *Scrum Sprints*, memungkinkan untuk merencanakan dan mengelola *sprint*.
- c. *Scrum Filters*, mempermudah dalam menemukan dan fokus pada tugas yang relevan sesuai dengan fase *sprint* atau kriteria lainnya.
- d. *Sprint Taskboard*, menampilkan seluruh tugas dalam *sprint* dan memungkinkan pengguna untuk memindahkan tugas melalui berbagai status seperti "*To Do*", "*In Progress*", dan "*Done*".

## **2.2 Penelitian Terkait**

### **2.2.1 Cawa Lampung : Kamus Bahasa Indonesia Lampung Dialek A Berbasis Android**

Penelitian ini dilakukan oleh Febi Eka Febriansyah, Ardiansyah, dan Afan Darmaji, pada tahun 2020. Tujuan penelitian ini adalah mengembangkan perangkat lunak terjemahan Indonesia Lampung dialek A berbasis Android. Metode pengembangan yang digunakan adalah *Extreme Programming*. Hasil penelitian ini adalah aplikasi Android bernama Cawa Lampung, yang berfungsi untuk menerjemahkan bahasa Indonesia ke bahasa Lampung dialek A. Aplikasi ini dapat menerjemahkan kata maupun kalimat dari bahasa Indonesia ke bahasa Lampung dialek A atau sebaliknya. Kemudian berdasarkan hasil pengujian menggunakan *Black Box*, fungsionalitas pada aplikasi ini berjalan dengan baik. Tetapi, pada saat menerjemahkan kata di atas 150 kata, aplikasi memerlukan waktu yang lama untuk menampilkan hasil terjemahan. Dari segi fitur, aplikasi yang dihasilkan masih terbatas, hanya menyediakan fitur untuk menerjemahkan kata. [28].

### **2.2.2 Rancang Bangun Aplikasi Kamus Bahasa dan Aksara Lampung Dialek A dan Dialek O Berbasis Android**

Penelitian ini dilakukan oleh Muhammad Fauzan Azima dan Siti Nur Laila pada tahun 2020. Tujuan penelitian ini adalah mengembangkan dan merancang aplikasi berbasis Android yang bertujuan untuk mempermudah guru, siswa, serta masyarakat Lampung dalam mempelajari bahasa dan aksara Lampung melalui perangkat mobile. Penelitian ini menggunakan metode pengembangan *Agile Development Method* karena dinilai dapat beradaptasi dengan cepat terhadap perubahan-perubahan di dalam sistem. Kemudian, metode pengujian yang digunakan adalah *Black Box*. Hasil dari penelitian ini adalah sebuah aplikasi kamus bahasa dan aksara Lampung berbasis Android. Aplikasi yang dibangun memiliki fitur penerjemahan kata berdasarkan dialek, hanya saja hasil yang ditampilkan pada saat menerjemahkan kata belum lengkap. Hasil terjemahan belum memuat informasi seperti contoh penggunaan kata dalam kalimat dan lokasi penuturannya, serta pengguna tidak banyak terlibat dalam proses penambahan kata [29].

### **2.2.3 Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan *Prototype* (Studi Kasus : Toko Uda Fajri)**

Penelitian ini dilakukan oleh Faisal Fajar Nursaid, Adam Hendra Brata, dan Agi Putra Kharisma, pada tahun 2020. Adapun tujuan dari penelitian ini adalah merancang dan membangun sistem untuk mempermudah pedagang untuk tidak lagi mencatat persediaan barang secara manual dengan kertas. Pada penelitian ini perancangan aplikasi Android menggunakan alat React Native karena dinilai dapat memberi kemudahan dan kesederhanaan dalam proses pengembangan maupun pembaruan aplikasi. Untuk membantu dalam memberikan gambaran deskripsi visual sistem penelitian ini menggunakan *Unified Modeling Language* (UML). Kemudian berdasarkan hasil pengujian dengan *Black Box* aplikasi berjalan dengan baik karena 15 kebutuhan fungsional yang diuji memiliki hasil yang valid [14].

### **2.2.4 Aplikasi Manajemen Surat Masuk dan Surat Keluar Berbasis *Mobile* Di Perum Bulog Subdivre**

Penelitian ini dilakukan oleh Rahmat Hidayatullah, Vivi Asmawati, dan Deasy Wahyuni, pada tahun 2020. Adapun tujuan dari penelitian ini adalah membangun aplikasi manajemen surat berbasis *mobile* di Perum BULOG Subdivre Dumai, untuk mempermudah proses birokrasi penindaklanjutan dan pendisposisian surat masuk serta persetujuan surat keluar. Penelitian ini menggunakan alat React Native *Expo* karena alat ini menyediakan beragam pustaka, komponen, dan layanan yang mempermudah proses pengembangan. Dengan fitur tersebut, beberapa konfigurasi dasar tidak perlu dilakukan secara manual sejak awal. Hasil dari penelitian ini adalah aplikasi yang memudahkan pimpinan dalam mengakses informasi surat masuk dan surat keluar di mana saja dan kapan saja, sekaligus mempercepat pengalokasian tindak lanjut serta disposisi surat masuk kepada bidang terkait, termasuk proses persetujuan surat keluar oleh pimpinan [30].

### **2.2.5 Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode *Scrum* pada Aplikasi Presensi Karyawan Glints Academy**

Penelitian ini dilakukan oleh Zidni Iman Sholihati dan Imam Tahyudin, pada tahun 2022. Tujuan penelitian ini adalah mengembangkan aplikasi sisi *client* dan *server* yang dapat berkomunikasi dengan baik sehingga dapat digunakan oleh karyawan. Metode pengembangan yang digunakan pada penelitian ini adalah *scrum* karena berkarakter adaptif, tangkas, dan fleksibel. Selain itu, metode *scrum* juga dianggap berdampak positif dalam manajemen perangkat lunak dan juga berdampak bagi manajemen sumber daya manusia. Walaupun pada saat proses pengembangan terjadi perubahan, *scrum* mampu mengembangkan fitur di tengah proses pengembangan aplikasi tanpa mempengaruhi pengembangan fitur utama. Sehingga, aplikasi tiga-tingkat pada aplikasi presensi berhasil dikembangkan [31].

### **2.2.6 Penerapan Metode *SCRUM* dalam Pengembangan Sistem Informasi Layanan Kawasan**

Penelitian ini dilakukan oleh Warkim, Muhamad Hanif Muslim, Farhan Harvianto, dan Setiawan Utama, pada tahun 2020. Penelitian ini bertujuan untuk mempermudah proses pengajuan dan pemantauan layanan di lingkungan LIPI, baik bagi pegawai maupun pengelola kawasan. Adapun metode pengembangan yang digunakan adalah *Scrum*, karena metode ini dinilai mudah dikontrol dan fleksibel. Serta, memuat strategi pengembangan menyeluruh, dimana seluruh tim bekerja sebagai satu unit untuk mencapai *goal* yang sama. Berdasarkan hasil pengujian metode *Black Box*, sistem berjalan dengan baik sesuai dengan fungsinya. Penelitian ini menghasilkan perangkat lunak yang mampu memudahkan pegawai dalam mengajukan permohonan terkait dengan kepegawaian, serta mendukung pengelola di kawasan maupun pusat dalam menjalankan tugas secara efektif dan efisien [10].

### **2.2.7 *Comparative Analysis Of Scrum And Kanban Methodologies In Agile Software Development***

Penelitian ini dilakukan oleh Megha Suryanarayana Ravi dan Dr. Anthony M. Townsend. Tujuan penelitian ini adalah guna mengetahui perbedaan dan kesamaan

metodologi *scrum* dan *kanban*, sehingga dapat membuat keputusan yang tepat dalam mengimplementasikannya. Hasil dari penelitian didapati bahwa *scrum* secara eksplisit mengatur peran, acara, dan artefak untuk memastikan tim bekerja secara terorganisir dan efisien. Dengan *scrum* juga meminimalisir terjadinya penyimpangan tugas-tugas atau tujuan dalam proses pengerjaan proyek, karena *scrum* dirancang untuk menangani *backlog item* yang terpecah menjadi bagian-bagian kecil (*task*). Serta, setiap *sprint* pada *scrum* memiliki tenggat waktu yang jelas, sehingga pengerjaan *task* menjadi lebih terukur [32].

### **2.2.8 A Comparative Analysis of Native vs React Native Mobile App Development**

Penelitian ini dilakukan oleh Naveen Chikkanayakanahalli Ramachandrappa pada tahun 2024. Tujuan penelitian ini adalah untuk membandingkan kerangka kerja native dengan kerangka kerja *cross platform*. Hasil dari penelitian ini diperoleh bahwa dengan *react native* dapat mempersingkat waktu pengembangan, karena kemampuan *hot reload* yang dimiliki *React Native* memungkinkan pengembang untuk melihat perubahan secara instan. Dibandingkan dengan aplikasi *native*, aplikasi dengan *React Native* memerlukan penggunaan sumber daya yang lebih sedikit tinggi, karena adanya *bridge* sebagai perantara kode JavaScript dan komponen *native* [33].

### **2.2.9 Software Testing with the approach of Black Box Testing on the Academic Information System**

Penelitian ini dilakukan oleh Supriyono pada tahun 2020. Tujuan penelitian ini adalah untuk menguji SIAKAD SDIT Robbani guna mengetahui apakah sistem ini dapat berjalan sesuai dengan yang diharapkan atau tidak. Metode pengujian yang digunakan adalah *Black Box*. Hasil pengujian menunjukkan sistem penilaian belum dapat memenuhi harapan karena tidak ada batasan nilai sehingga nilai acak akan tetap tersimpan oleh sistem [34].

### ***2.2.10 Performing User Acceptance Test with System Usability Scale for Graduation Application***

Penelitian ini dilakukan oleh Nik Azlina Nik Ahmad dan Puteri Norliana Nor'Ain Megat Sazali, pada tahun 2021. Penelitian ini menguji sistem aplikasi kelulusan pada *platform website* dan seluler dengan menggunakan *User Acceptance Test* (UAT) dan *System Usability Scale* (SUS). Terdapat delapan skenario pengujian dan terbagi menjadi 25 tugas untuk pengujian dengan UAT. Hasilnya seluruh tugas pengujian berhasil dilakukan atau hasilnya 100% berhasil tanpa masalah dan kegagalan [35].





### 3.2 Alat Penelitian

Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut:

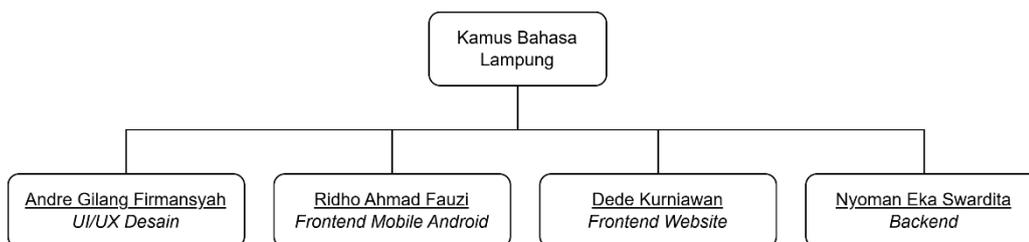
Tabel 3.2 Alat Penelitian

No	Nama	Spesifikasi	Deskripsi
1.	Laptop	Prosesor AMD Ryzen 7, RAM 16 GB, ROM 512 GB, Windows 11 64bit	Perangkat keras yang digunakan untuk pengembangan aplikasi dan penulisan laporan penelitian.
2.	Mouse	-	Perangkat keras yang digunakan untuk menggerakkan kursor
3.	Figma	Versi 124.2.9	Perangkat lunak untuk melihat desain
4.	Taiga	-	Perangkat lunak manajemen proyek yang mendukung metode <i>agile</i> untuk perencanaan, pelacakan, dan pengelolaan proyek.
5.	Visual Studio Code	Versi 1.93	Perangkat lunak editor kode yang digunakan untuk pengembangan aplikasi.
6.	Terminal	-	Antarmuka pengguna berbasis teks yang memungkinkan pengguna untuk menjalankan perintah dan mengelola <i>file</i> , pada sistem operasi.
7.	JavaScript	-	Bahasa pemrograman untuk pengembangan <i>multiplatform</i>

Tabel 3.2 Alat Penelitian (Lanjutan)

No	Nama	Spesifikasi	Deskripsi
8.	React Native Expo	Versi 51.0.25	<i>Framework</i> untuk pengembangan aplikasi <i>mobile</i> yang memungkinkan pembuatan aplikasi <i>Android</i> menggunakan JavaScript dan React, dengan dukungan fitur <i>build</i> yang mudah melalui <i>platform</i> Expo.

### 3.3 Struktur Tim Kamus Bahasa Lampung

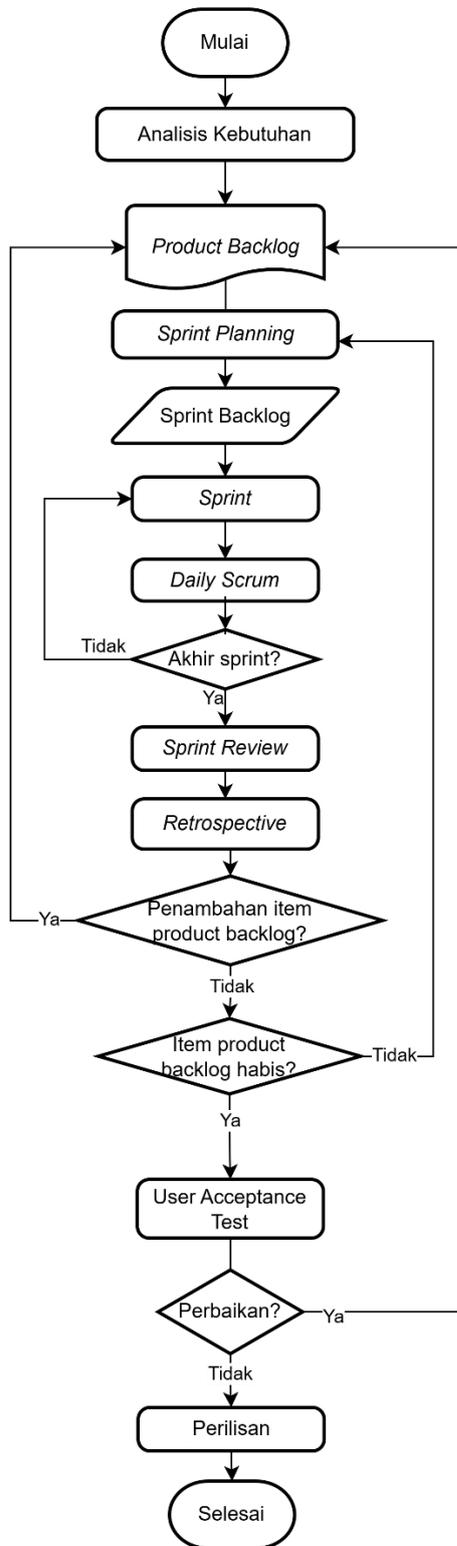


Gambar 3.1 Struktur Tim Kamus Bahasa Lampung

Perancangan proyek kamus bahasa lampung pada penelitian ini terbagi menjadi beberapa peran. Komposisi peran pada penelitian ini adalah terdapat dua mahasiswa *frontend* (*website dan mobile android*), satu mahasiswa *backend*, dan satu mahasiswa UI/UX.

### 3.4 Tahapan Penelitian

Adapun tahapan penelitian yang digunakan adalah sebagai berikut:



Gambar 3.2 Tahapan Penelitian

### **3.4.1 Analisis Kebutuhan**

Tahap ini mencakup pengumpulan data melalui studi literatur dan observasi. Data yang diperoleh akan didiskusikan bersama tim perancangan untuk merumuskan kebutuhan pengguna. Selanjutnya, kebutuhan pengguna tersebut akan digunakan sebagai dasar dalam menyusun kebutuhan fungsional dan nonfungsional.

Pengumpulan data melalui studi literatur akan dilakukan dengan mengumpulkan data dan informasi dari penelitian-penelitian sebelumnya. Data dan informasi tersebut akan diperoleh dari artikel serta dokumentasi yang relevan dengan perancangan sistem dalam penelitian ini. Tujuan dari studi literatur adalah untuk mendapatkan informasi terkait kondisi bahasa Lampung yang sedang terancam dan solusi-solusi yang telah diterapkan sebelumnya. Selanjutnya, observasi akan dilakukan dengan menggunakan metode SWAT untuk membandingkan aplikasi-aplikasi serupa pada platform *website* maupun *mobile*. Tujuan dari observasi ini adalah untuk memperoleh fitur-fitur yang dapat diimplementasikan dalam perancangan sistem ini.

#### **3.4.1.1 Analisis Perbandingan Aplikasi Serupa**

Untuk memperoleh daftar kebutuhan pengguna dan memiliki nilai tambah dibandingkan aplikasi sejenis, dilakukan analisis terhadap aplikasi serupa. Analisis ini bertujuan untuk mengidentifikasi keunggulan, kelemahan, peluang, dan ancaman (SWOT) dari aplikasi sejenis yang sudah ada. Hasil analisis ini menjadi dasar dalam merumuskan kebutuhan dan fitur yang akan dikembangkan pada sistem yang dirancang. Berikut ini adalah hasil analisis terhadap beberapa aplikasi kompetitor yang relevan:

##### **a. Kamus Lampung-Indonesia KBPL**

Aplikasi ini memiliki keunggulan dalam menyediakan data yang cukup lengkap dan kemampuan pencarian berdasarkan satu kata. Namun, data yang tersedia bersifat statis dan tidak diperbarui secara berkala, serta memiliki tampilan aplikasi yang kurang menarik. Peluang utama yang dapat dimanfaatkan adalah meningkatkan desain antarmuka dan memberikan akses bagi masyarakat untuk berkontribusi dalam memperkaya data. Di sisi lain, ancaman bagi aplikasi ini berasal dari aplikasi

serupa yang lebih populer serta penurunan relevansi data akibat kurangnya pembaruan secara rutin.

b. Kamus Bahasa Lampung (Aplikasi Android)

Aplikasi ini memiliki keunggulan berupa fitur pembelajaran aksara, budaya, dan sastra Lampung, serta kemampuan untuk menerjemahkan teks hingga kalimat. Namun, kekurangannya terletak pada alur penambahan kata yang belum jelas dan hasil terjemahan yang masih bersifat per kata, serta tidak dapat melaporkan kata pada saat ditemukan kesalahan. Peluang untuk pengembangan aplikasi ini ada pada peningkatan fitur pembelajaran interaktif dan pemanfaatan teknologi AI untuk meningkatkan akurasi terjemahan. Di sisi lain, ancaman yang dihadapi berasal dari persaingan dengan aplikasi lain yang menawarkan teknologi lebih canggih dan pasar yang lebih terbatas.

c. *Translate* Bahasa Lampung (Aplikasi Android)

Aplikasi ini menawarkan keunggulan melalui fitur OCR yang memungkinkan pengguna mengenali teks Latin untuk diterjemahkan, serta mendukung berbagai dialek Lampung. Namun, kelemahannya terletak pada tampilan teks yang kurang menarik, hasil terjemahan kalimat yang tidak sesuai dengan kaidah, dan keterbatasan perbendaharaan kata. Peluang yang ada meliputi perbaikan tampilan, penambahan fitur aksara, dan perluasan kosakata. Di sisi lain, ancaman yang dihadapi adalah kesulitan dalam menjaga kualitas data tanpa adanya kurasi dan minimnya dukungan aksara.

d. Kamus Lampung (Aplikasi Android)

Aplikasi ini memiliki keunggulan dalam menyajikan daftar kata yang lengkap serta menyediakan fitur untuk menyimpan kata favorit. Namun, kekurangannya terletak pada adanya iklan yang mengganggu, kurangnya kategori dialek, dan tampilan aplikasi yang kurang menarik. Peluang untuk pengembangan aplikasi ini adalah dengan memperbaiki tampilan dan menambahkan kategori dialek guna memperluas fungsionalitas. Ancaman yang dihadapi adalah kurangnya daya tarik aplikasi ini di tengah persaingan dengan aplikasi serupa.

Adapun rangkuman dari hasil analisis SWOT pada aplikasi serupa adalah sebagai berikut.

Tabel 3.3 Perbandingan dengan Aplikasi Serupa

Fitur	Kamus Lampung Indonesia - KBPL	Kamus Bahasa Lampung – Ell Studios	Translate Bahasa Lampung – Tapis Dev	Kamus Bahasa Lampung – Ant Bee Dev	Lamban Bahasa – Penelitian ini
Mengajukan kata	✗	✓	✗	✗	✓
Meninjau Kata	✗	✗	✗	✗	✓
Meninjau laporan	✗	✗	✗	✗	✓
Menerjemahkan kata	✓	✓	✓	✓	✓
Menerjemahkan kata berdasarkan dialek	✗	✓	✓	✗	✓
Penulisan Aksara (Kata dan Kalimat)	✓	✓	✗	✗	✓
Contoh penggunaan kata pada kalimat	✗	✗	✗	✗	✓
Informasi lokasi penuturan kata	✗	✗	✗	✗	✓
Laporkan kata	✗	✗	✗	✗	✓
Menampilkan daftar kata terurut abjad	✗	✓	✗	✗	✓
Papan peringkat kontributor	✗	✗	✗	✗	✓
Blog budaya Lampung	✗	✗	✗	✗	✓

### 3.4.1.2 Kebutuhan Pengguna, Kebutuhan Fungsional, dan Kebutuhan Non Fungsional

Dilakukan diskusi bersama dengan tim perancangan untuk membahas temuan-temuan dari hasil perbandingan dengan aplikasi kompetitor dan studi literatur. Kemudian dari hasil diskusi, diperoleh daftar kebutuhan pengguna yang akan diterapkan dalam perancangan sistem ini. Selanjutnya, kebutuhan pengguna tersebut diuraikan lebih lanjut menjadi kebutuhan fungsional dan nonfungsional. Adapun daftar kebutuhan pengguna hasil dari diskusi bersama dengan tim perancangan adalah sebagai berikut:

1. Pengguna ingin membuat akun sebagai kontributor
2. Pengguna ingin menerjemahkan kata dari dan ke bahasa Lampung.
3. Pengguna ingin melihat daftar kata yang tersedia
4. Pengguna ingin berkontribusi terhadap perbendaharaan kata kamus bahasa Lampung.
5. Pengguna ingin memberikan laporan atas kesalahan yang ditemukan pada kata.
6. Pengguna ingin mengetahui siapa kontributor teratas.
7. Pengguna ingin mengetahui statistik kamus bahasa Lampung.
8. Pengguna ingin membaca *blog* terkait budaya Lampung.
9. Pengguna ingin mengelola data profil pribadi.

Adapun daftar kebutuhan fungsional hasil dari diskusi bersama dengan tim perancangan adalah sebagai berikut:

Tabel 3.4 Kebutuhan Fungsional

ID	Penjelasan
KF-01	Sistem dapat menyediakan pendaftaran bagi pengguna dengan peran kontributor
KF-02	Sistem dapat menyediakan layanan bagi pengguna untuk masuk dan keluar sistem
KF-03	Sistem dapat menyediakan layanan bagi pengguna untuk mengubah kata sandi pada saat terlupa
KF-04	Sistem dapat menerjemahkan kata antara bahasa Lampung dan bahasa Indonesia, baik dari bahasa Lampung ke bahasa Indonesia maupun sebaliknya.

Tabel 3.4 Kebutuhan Fungsional (Lanjutan)

ID	Penjelasan
KF-05	Sistem dapat menampilkan aksara untuk kata dan kalimat dalam bahasa Lampung.
KF-06	Sistem memiliki fitur untuk filter berdasarkan dialek pada saat mencari kata.
KF-07	Sistem memiliki fitur pelaporan yang memungkinkan pengguna untuk melaporkan kata tertentu.
KF-08	Sistem dapat menampilkan daftar kata yang tersedia
KF-09	Sistem dapat menampilkan daftar peringkat kontributor teratas
KF-10	Sistem dapat menampilkan statistik data aplikasi kamus, seperti jumlah kata, penambahan kata tiap minggu, jumlah kontributor, dan jumlah peninjau
KF-11	Sistem dapat menampilkan daftar <i>blog</i> yang dapat dibaca
KF-12	Sistem dapat menyediakan layanan bagi pengguna untuk mengajukan kata
KF-13	Sistem dapat menyediakan layanan bagi pengguna untuk menyimpan kata sebagai draf sebelum diajukan
KF-14	Sistem dapat menyediakan layanan bagi pengguna untuk mengelola draf, yaitu mengajukan draf untuk ditinjau dan menghapus draf
KF-15	Sistem dapat menyediakan layanan bagi pengguna untuk mengelola kata yang direvisi, yaitu mengajukan kata yang direvisi untuk ditinjau ulang
KF-16	Sistem dapat menampilkan daftar riwayat atau riwayat kontribusi berdasarkan status (ditinjau, diterima, dan ditolak).
KF-17	Sistem dapat menyediakan layanan bagi pengguna untuk meninjau kata dan memberikan keputusan yang sesuai (publikasi, tolak, dan revisi)
KF-18	Sistem dapat menyediakan layanan bagi pengguna untuk meninjau kata yang dilaporkan dan memberikan keputusan yang sesuai (edit dan hapus kata)
KF-19	Sistem dapat menampilkan daftar riwayat tinjau kata berdasarkan status (diterima, dan ditolak).
KF-20	Sistem dapat menyediakan layanan bagi pengguna untuk mengubah foto dan data profil
KF-21	Sistem dapat menyediakan layanan bagi pengguna untuk memperbarui kata sandi
KF-22	Sistem dapat menampilkan daftar pemberitahuan

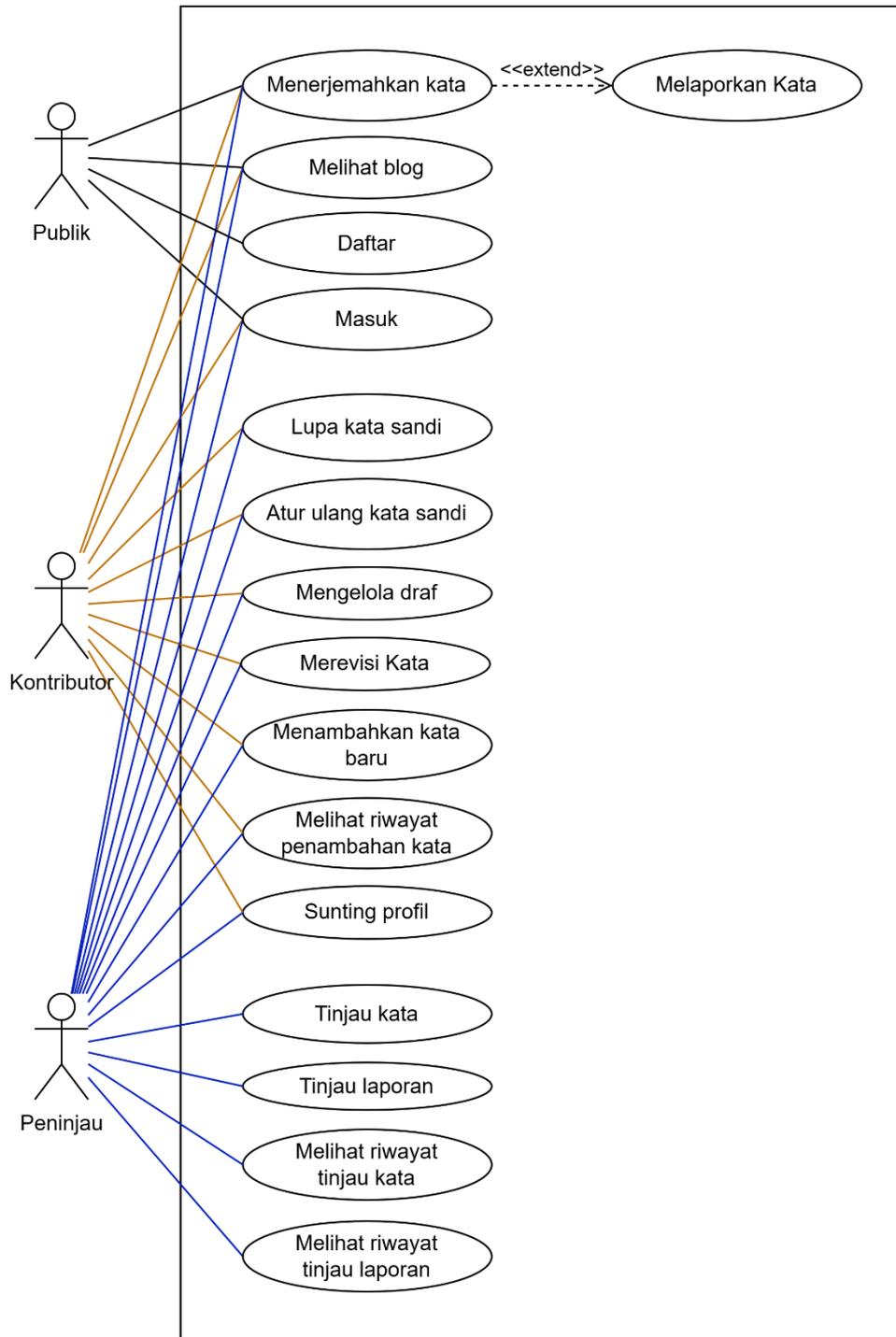
Adapun daftar kebutuhan *non-fungsional* hasil dari diskusi bersama dengan tim perancangan adalah sebagai berikut:

Tabel 3.5 Kebutuhan *Non-fungsional*

ID	Parameter	Penjelasan
KnF-01	<i>Availability</i>	1. Aplikasi harus selalu dapat diakses sepanjang waktu, yaitu 24 jam sehari dan 7 hari seminggu
KnF-02	<i>Reliability</i>	1. Sistem harus mampu menangani kesalahan <i>input</i> atau kegagalan eksternal dengan memberikan pesan yang informatif kepada pengguna tanpa menyebabkan <i>crash</i> .
KnF-03	<i>Ergonomy</i>	1. Antarmuka pengguna harus dirancang agar mudah dipahami dan digunakan secara intuitif. 2. Desain antarmuka harus ramah aksesibilitas, sehingga mendukung pengguna dengan kebutuhan khusus atau keterbatasan fisik.
KnF-04	<i>Portability</i>	1. Sistem harus kompatibel dengan sistem operasi Android dengan versi minimum 8.0 ( <i>Oreo</i> ) atau versi yang lebih baru
KnF-05	<i>Safety</i>	1. Sistem harus menerapkan langkah-langkah keamanan untuk menjaga data pengguna agar terhindar dari kehilangan atau kerusakan.
KnF-06	<i>Usability</i>	1. Pengguna harus mampu menyelesaikan tugas utama dalam waktu kurang dari 5 menit melalui langkah-langkah yang mudah dipahami dan efisien.

### 3.4.1.3 Use Case

Adapun *use case* aplikasi kamus bahasa lampung yang dirancang adalah sebagai berikut:



Gambar 3.3 Use Case

Berikut merupakan tabel definisi aktor pada *use case*:

Tabel 3.6 Definisi Aktor *Use Case*

Aktor	Deskripsi
Publik	Aktor dengan peran ini dapat mengakses fitur penerjemahan atau pencarian kata, melaporkan kata, dan registrasi
Kontributor	Aktor dengan peran ini dapat mengakses menu dasbor. Peran ini dapat melakukan hal-hal yang dilakukan oleh aktor publik. Selain itu, aktor ini juga dapat mengakses fitur menambahkan kata, melihat riwayat kontribusi, lupa kata sandi, ubah kata sandi, mengelola draf, dan sunting profil.
Peninjau	Aktor dengan peran ini dapat mengakses menu dasbor. Peran ini dapat melakukan hal-hal yang dilakukan oleh aktor publik dan kontributor. Selain itu, aktor ini juga dapat mengakses fitur tinjau kata, tinjau laporan, dan melihat riwayat tinjauan.

Berikut merupakan tabel definisi *use case*:

Tabel 3.7 Definisi *Use Case*

<i>Use Case</i>	Deskripsi
Menerjemahkan kata	Pengguna dapat menerjemahkan kata dengan memilih dialek dan memasukkan kata yang ingin dicari.
Melihat hasil terjemahan	Pengguna dapat melihat terjemahan kata dari hasil pencarian. Pengguna dapat mengetahui informasi arti dari kata, contoh kalimat, kata dan kalimat dalam aksara, dialek kata, dan lokasi penuturan kata.
Melaporkan kata	Pengguna dapat melaporkan kata apabila ditemukan kesalahan data pada suatu kata.
Lupa kata sandi	Pengguna dapat mengubah kata sandi pada saat terlupa
Atur kata sandi	Pengguna dapat mengubah kata sandi
Mengelola draf	Pengguna dapat mengelola daftar draf kata yang dimiliki. Pengguna dapat mengajukan draf atau menghapus draf.

Tabel 3.7 Definisi *Use Case* (Lanjutan)

Menambahkan kata	Pengguna dapat mengajukan kata untuk masuk ke dalam <i>database</i> aplikasi dan apabila masih ragu pengguna dapat menyimpan terlebih dahulu kata tersebut sebagai draf.
Melihat riwayat kontribusi	Pengguna dapat melihat daftar riwayat kontribusi pengajuan kata
Sunting profil	Pengguna dapat mengubah foto dan data profil
Tinjau kata	Pengguna dapat meninjau kata dan memberikan keputusan pada kata yang ditinjau. Pengguna dapat menerima, menolak, dan merevisi kata.
Tinjau laporan	Pengguna dapat meninjau laporan dan memberikan keputusan pada kata yang ditinjau. Pengguna dapat mengubah dan menghapus kata.
Melihat riwayat tinjauan	Pengguna dapat melihat daftar riwayat meninjau kata
Registrasi	Pengguna yang belum terdaftar pada sistem dapat melakukan registrasi untuk dapat mengakses fitur untuk berkontribusi
Masuk	Pengguna yang telah terdaftar dapat masuk ke dalam sistem

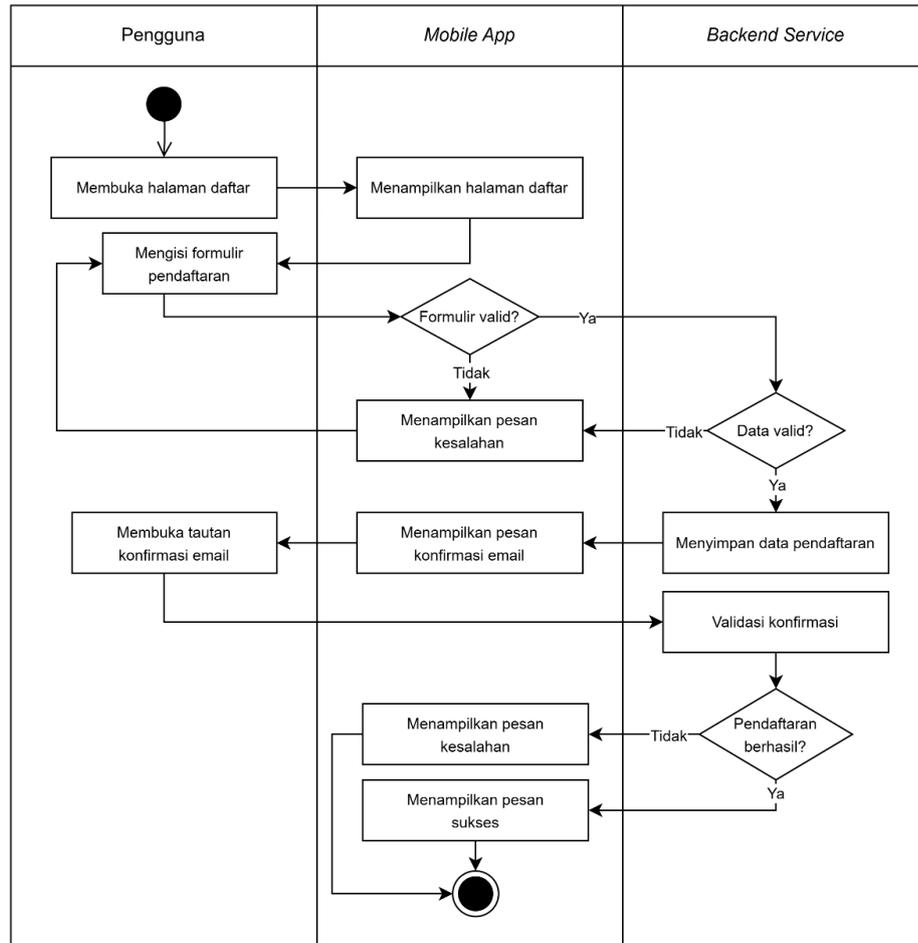
#### 3.4.1.4 *Activity Diagram*

*Activity Diagram* adalah diagram yang menggambarkan alur aktivitas atau proses dalam sistem, mulai dari awal hingga akhir, termasuk keputusan atau percabangan yang terjadi. Diagram ini membantu memahami bagaimana aktivitas saling terkait dan berurutan dalam sistem.

##### a. *Activity Diagram* daftar

Merujuk pada gambar 3.4, ilustrasi tersebut menunjukkan alur proses mendaftar sebagai kontributor. Alur *activity* ini diawali dengan pengguna yang membuka halaman daftar dan mengisi formulir yang diperlukan. Kemudian aplikasi maupun *server* memvalidasi data yang dimasukkan pengguna. Apabila data yang telah diberikan valid maka *server* akan menyimpan data pendaftaran pengguna. Namun, jika data yang diberikan tidak valid, aplikasi menampilkan pesan kesalahan.

Selanjutnya aplikasi menampilkan pesan bahwa telah dikirimkan konfirmasi pembuatan akun pada *email*. Kemudian pengguna membuka pesan pada *email* dan membuka tautan yang telah dikirimkan. Selanjutnya *server* merespon dengan memvalidasi konfirmasi yang diberikan. Apabila berhasil maka aplikasi menampilkan pesan sukses, namun apabila gagal aplikasi menampilkan pesan kesalahan.

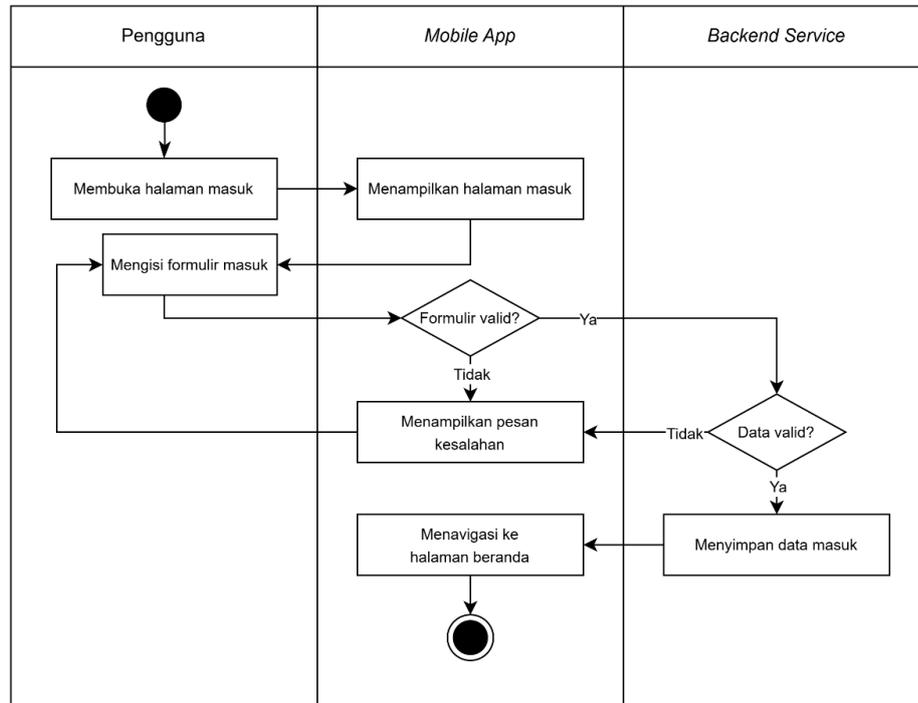


Gambar 3.4 Activity Diagram daftar

b. Activity Diagram masuk

Merujuk pada gambar 3.5, ilustrasi tersebut menunjukkan alur proses masuk ke dalam aplikasi. Proses diawali dengan pengguna yang membuka halaman masuk dan mengisi formulir yang diperlukan. Kemudian aplikasi maupun *server* memvalidasi data yang dimasukkan pengguna. Apabila data yang telah diberikan valid maka *server* akan menyimpan data pengguna ke dalam keadaan masuk ke

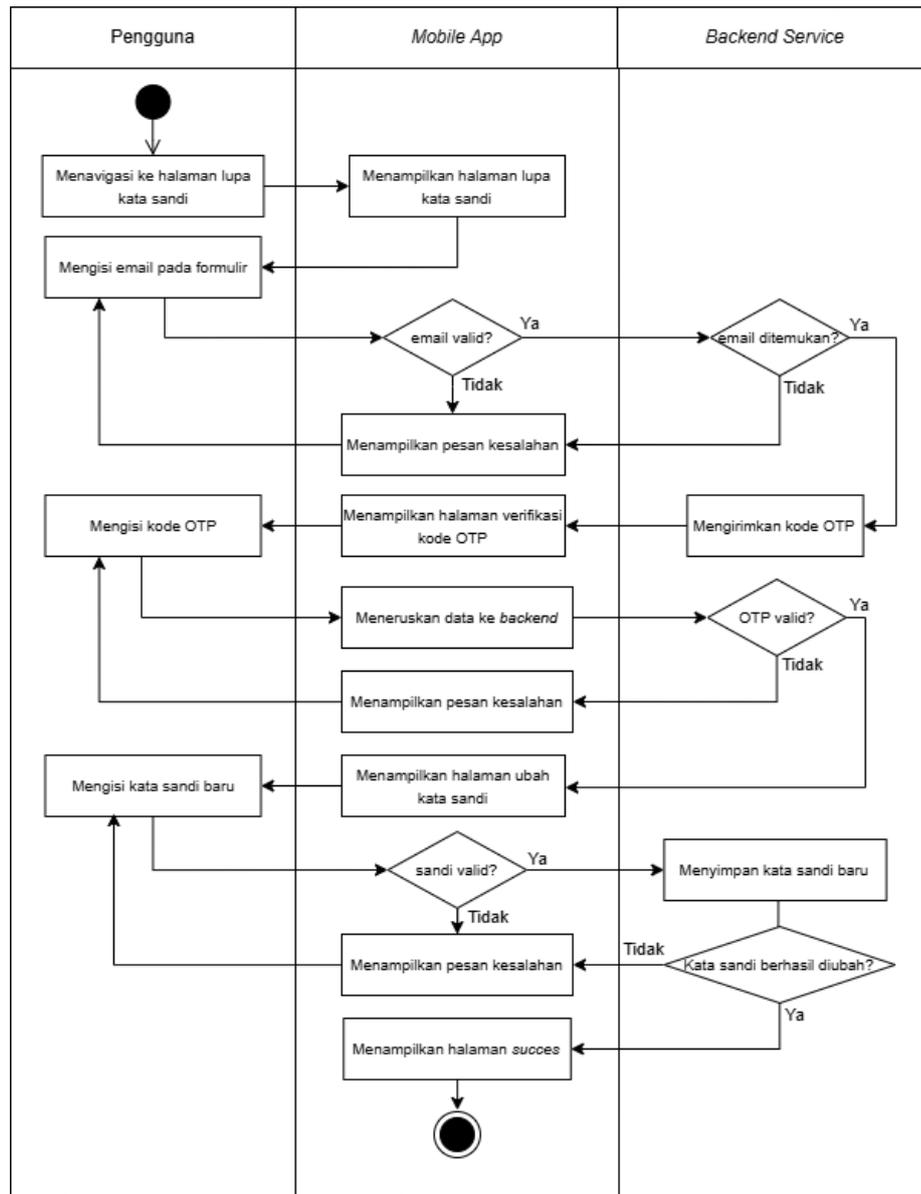
dalam sistem dan aplikasi menavigasi ke halaman beranda. Namun, jika data yang diberikan tidak valid, aplikasi menampilkan pesan kesalahan.



Gambar 3.5 Activity Diagram masuk

c. Activity Diagram lupa kata sandi

Merujuk pada gambar 3.6, ilustrasi tersebut menunjukkan alur proses mengubah kata sandi apabila terlupa. Alur *activity* ini diawali dengan pengguna mengakses halaman lupa kata sandi dan mengisi *email* yang digunakan pada saat mendaftar akun. Kemudian aplikasi maupun *server* memvalidasi *email* yang dimasukkan. Apabila *email* yang diberikan valid maka *server* akan mengirimkan kode *One Time Password (OTP)*. Selanjutnya aplikasi menampilkan halaman verifikasi OTP dan pengguna mengisi OTP sesuai dengan yang telah dikirimkan di *email*. Kemudian *server* memvalidasi OTP yang diberikan, apabila valid aplikasi akan mengarahkan ke halaman ubah kata sandi. Selanjutnya pengguna mengisi kata sandi baru dan aplikasi memvalidasi sandi yang telah diberikan, apabila valid kata sandi diteruskan ke *server* dan jika tidak valid aplikasi menampilkan pesan kesalahan. Kemudian kata sandi diperbarui oleh *server*, apabila berhasil aplikasi menampilkan halaman *sukses* dan jika tidak aplikasi menampilkan pesan kesalahan.

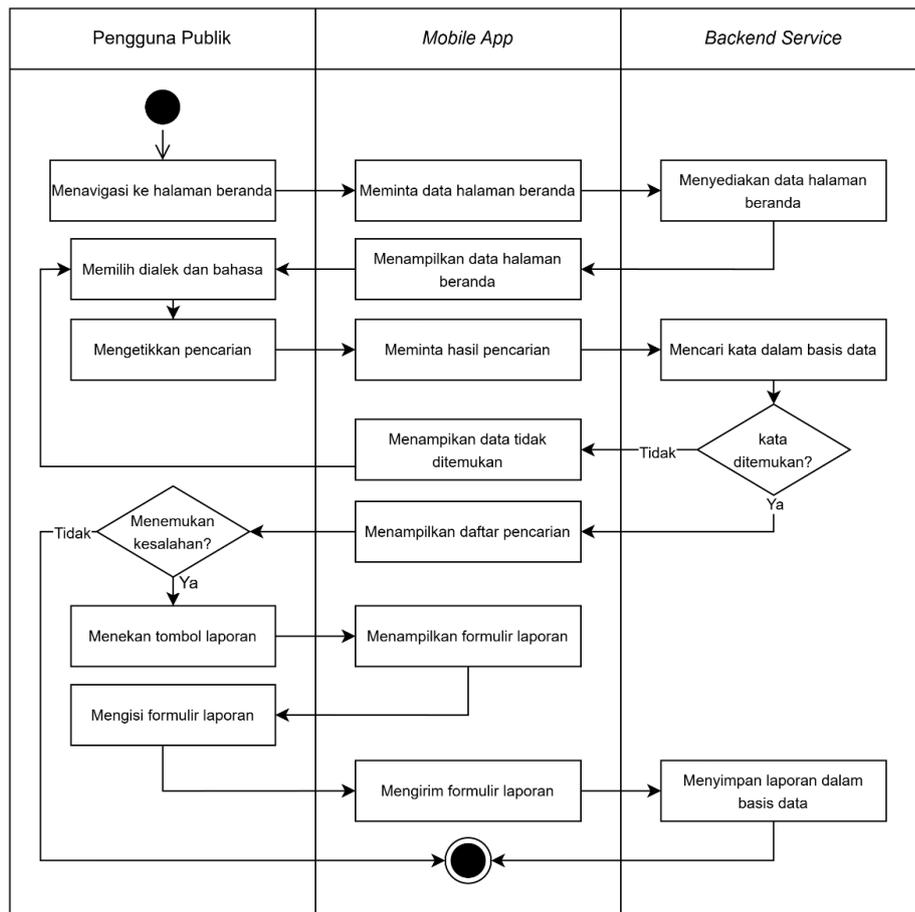


Gambar 3.6 Activity Diagram lupa kata sandi

d. Activity Diagram menerjemahkan kata dan pelaporan

Merujuk pada gambar 3.7, ilustrasi tersebut menunjukkan alur proses menerjemahkan kata dan pelaporan kata. Alur activity ini diawali dengan pengguna mengakses halaman beranda. Kemudian aplikasi meminta data untuk section pencarian kata, yaitu data dialek ke server. Selanjutnya pengguna memilih bahasa dan dialek, serta kata yang dicari. Kemudian aplikasi meneruskan data pencarian ke server dan server mencari kata yang diinginkan pengguna berdasarkan data-data yang telah dimasukkan oleh pengguna. Apabila kata yang dicari ditemukan, aplikasi akan menampilkan hasilnya, namun apabila tidak ditemukan aplikasi akan

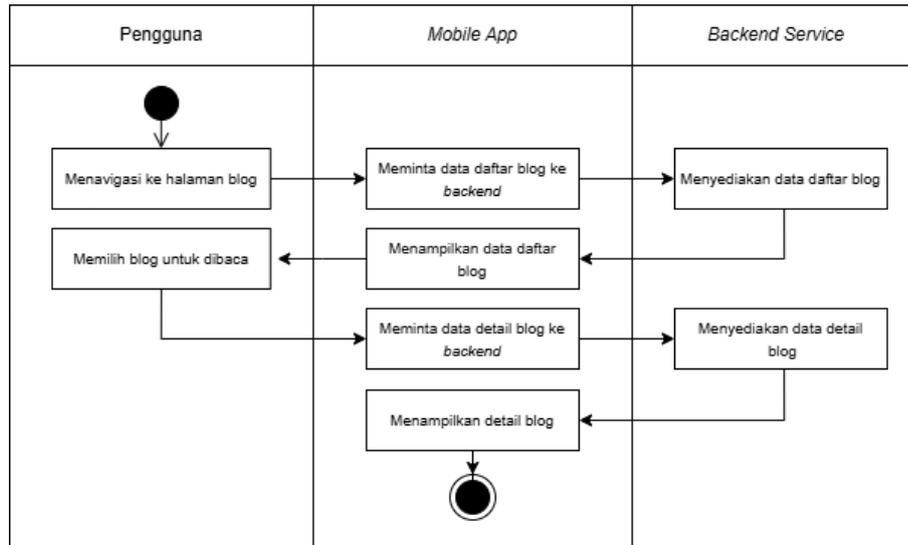
menampilkan pesan kesalahan. Selanjutnya apabila pengguna menemukan kesalahan dari kata yang ditemukan, pengguna dapat melaporkan kata tersebut. Kemudian aplikasi akan menampilkan formulir deskripsi laporan pengguna dan pengguna mengisi formulir tersebut. Selanjutnya *server* menyimpan laporan tersebut untuk ditindaklanjuti.



Gambar 3.7 Activity Diagram menerjemahkan kata dan pelaporan

e. Activity Diagram melihat blog

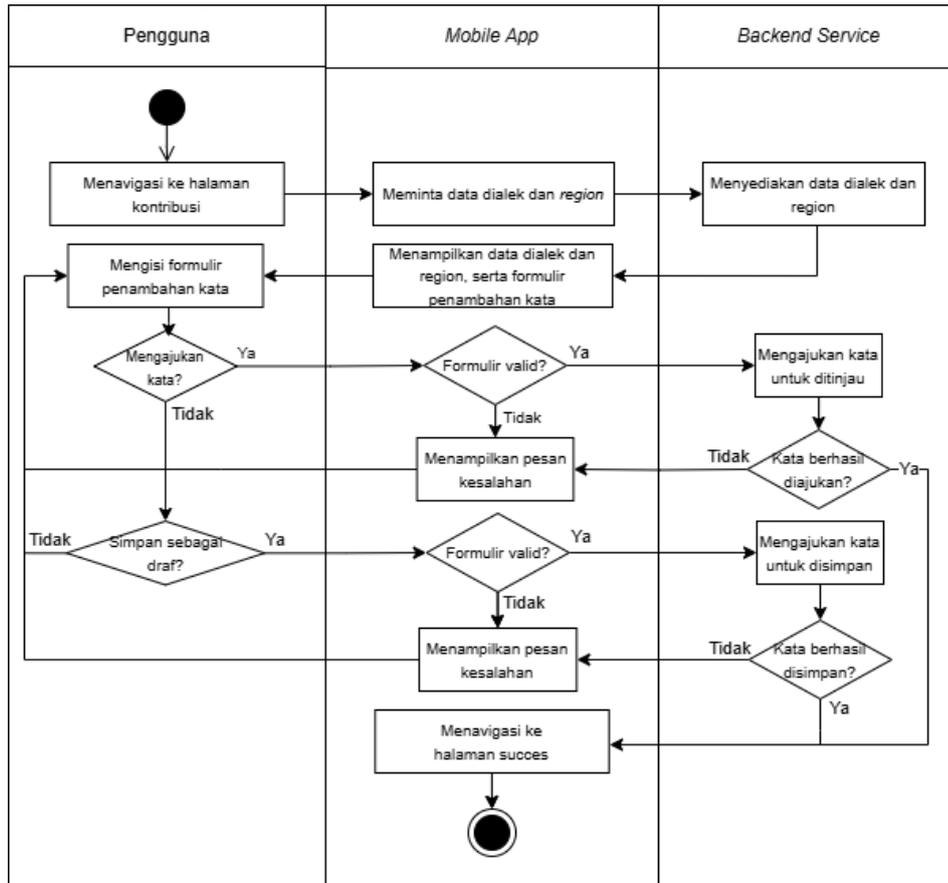
Merujuk pada gambar 3.8, ilustrasi tersebut menunjukkan alur proses menerjemahkan melihat *blog*. Alur *activity* ini diawali dengan pengguna mengakses halaman *blog*. Kemudian aplikasi akan meminta daftar data *blog* ke *server*. Setelah data diberikan oleh *server*, aplikasi akan menampilkan daftar *blog* yang diberikan. Selanjutnya pengguna memilih *blog* untuk dibaca dan aplikasi meminta detail *blog* ke *server* kembali. Kemudian aplikasi menampilkan detail *blog* yang telah diperoleh dari *server*.



Gambar 3.8 Activity Diagram melihat blog

f. Activity Diagram menambahkan kata baru

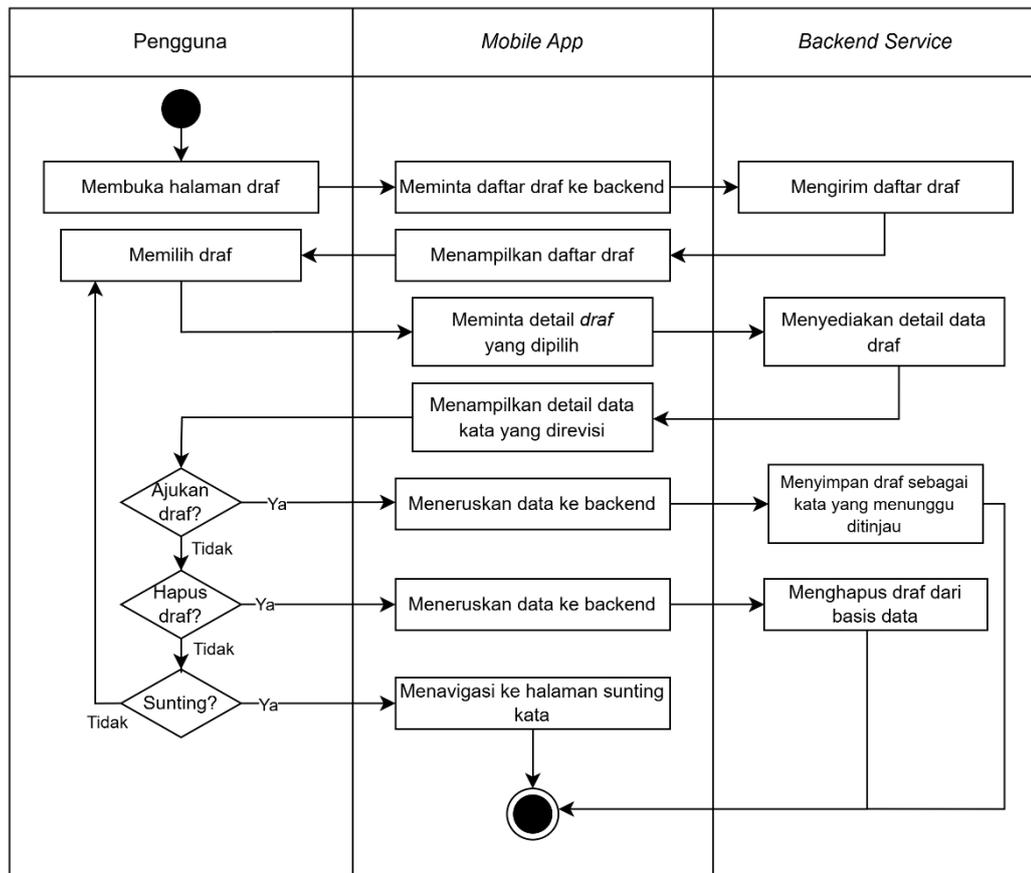
Merujuk pada gambar 3.9, ilustrasi tersebut menunjukkan alur proses menambahkan kata baru. Alur *activity* ini diawali dengan pengguna mengakses halaman kontribusi. Kemudian aplikasi meminta data dialek dan *region* ke *server* untuk ditampilkan pada *selector*. Selanjutnya pengguna dapat memilih aksi untuk mengajukan kata atau menyimpannya sebagai draf. Apabila pengguna memilih mengajukan kata, aplikasi akan memvalidasi data yang dimasukkan pengguna dan apabila valid akan diteruskan ke *server*; jika tidak valid aplikasi akan menampilkan pesan kesalahan. Kemudian *server* akan menyimpan data kata tersebut untuk diajukan. Apabila *server* berhasil mengajukan kata tersebut, aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi akan menampilkan pesan kesalahan. Kemudian apabila pengguna memilih menyimpan kata sebagai draf, aplikasi akan memvalidasi data yang dimasukkan pengguna dan apabila valid akan diteruskan ke *server*; jika tidak valid aplikasi akan menampilkan pesan kesalahan. Kemudian *server* akan menyimpan data kata tersebut untuk disimpan sebagai draf. Apabila *server* berhasil menyimpan kata tersebut, aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi akan menampilkan pesan kesalahan.



Gambar 3.9 Activity Diagram menambahkan kata baru

g. Activity Diagram mengelola draf

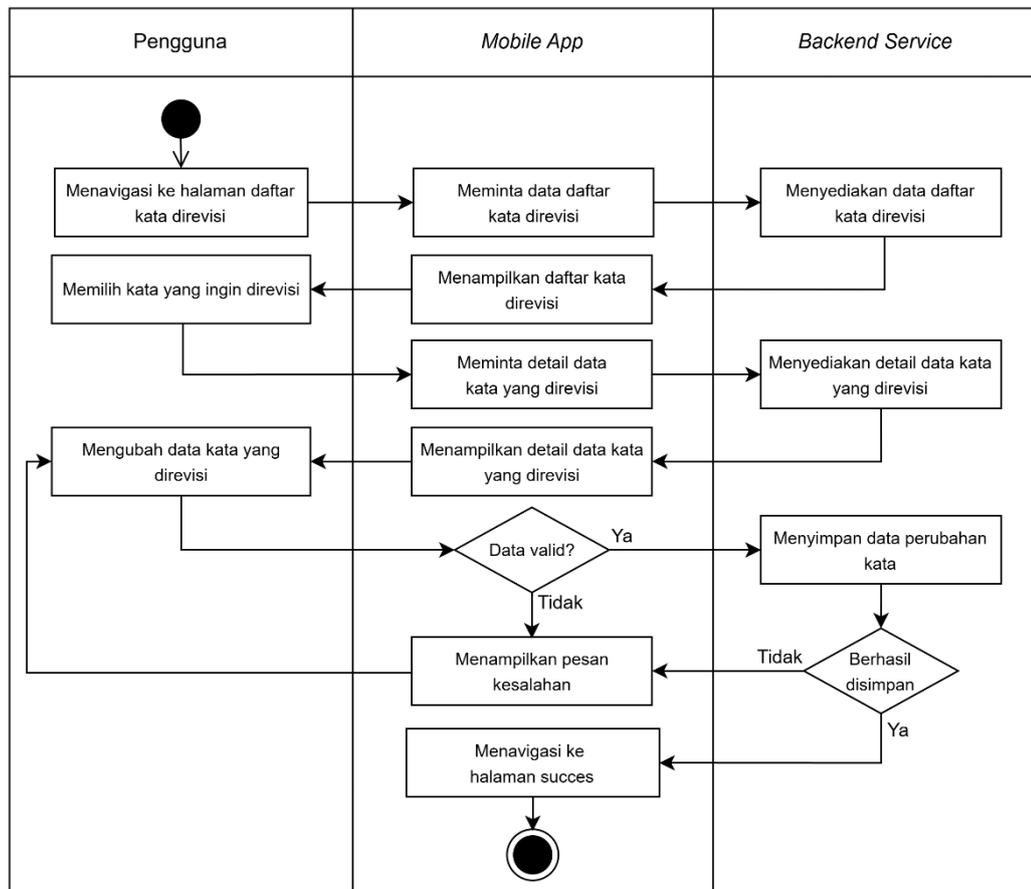
Merujuk pada gambar 3.10, ilustrasi tersebut menunjukkan alur mengelola draf. Alur *activity* ini diawali dengan pengguna mengakses halaman draf. Kemudian aplikasi meminta data daftar draf dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar draf dan pengguna memilih draf yang ingin dikelola. Kemudian aplikasi meminta detail data draf ke *server* dan menampilkannya. Selanjutnya pengguna dapat memilih aksi untuk mengajukan draf, hapus draf, dan ubah draf. Apabila pengguna memilih mengajukan draf aplikasi meneruskan data kata tersebut ke *server* dan *server* menyimpan kata tersebut sebagai kata yang menunggu untuk ditinjau. Jika pengguna memilih menghapus draf aplikasi meneruskan data kata tersebut ke *server* dan *server* menghapus kata tersebut dari *database*. Apabila pengguna memilih mengubah draf aplikasi akan menavigasi ke halaman ubah draf.



Gambar 3.10 Activity Diagram mengelola draf

h. Activity Diagram merevisi kata

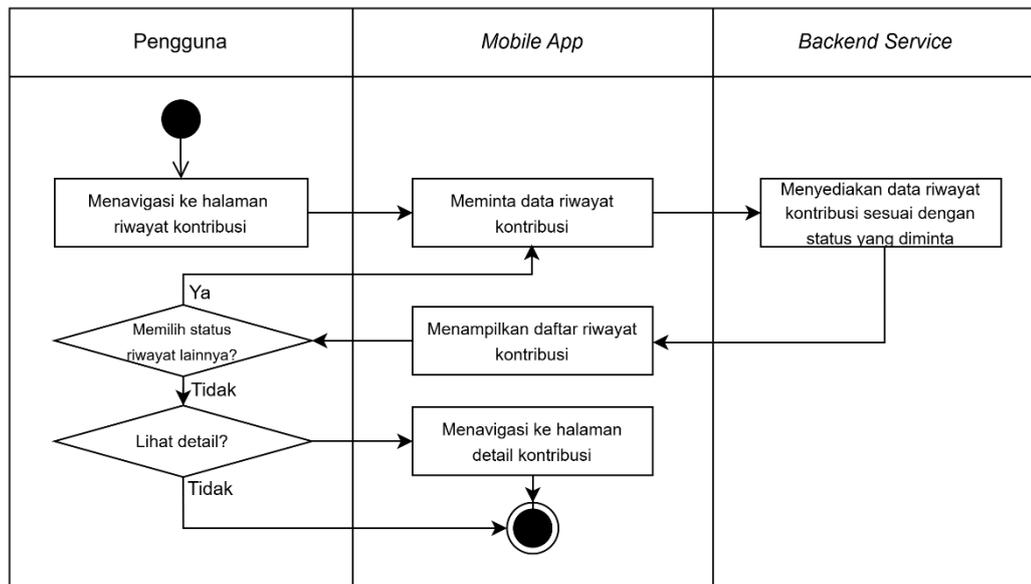
Merujuk pada gambar 3.11, ilustrasi tersebut menunjukkan alur proses merevisi kata. Alur *activity* ini diawali dengan pengguna mengakses halaman daftar kata direvisi. Kemudian aplikasi meminta daftar kata direvisi ke *server* dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar kata direvisi dan pengguna memilih kata direvisi yang ingin dikelola. Kemudian aplikasi meminta detail kata direvisi ke *server* dan menampilkannya. Selanjutnya pengguna mengubah data kata sesuai dengan pesan yang diberikan peninjau. Kemudian aplikasi memvalidasi data yang dimasukkan, apabila valid data akan diteruskan ke *server* dan jika tidak valid aplikasi akan menampilkan pesan kesalahan. Selanjutnya *server* akan menyimpan perubahan dan mengubah status menjadi “sedang ditinjau”. Apabila berhasil aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi menampilkan pesan kesalahan.



Gambar 3.11 Activity Diagram merevisi kata

i. Activity Diagram melihat riwayat penambahan kata

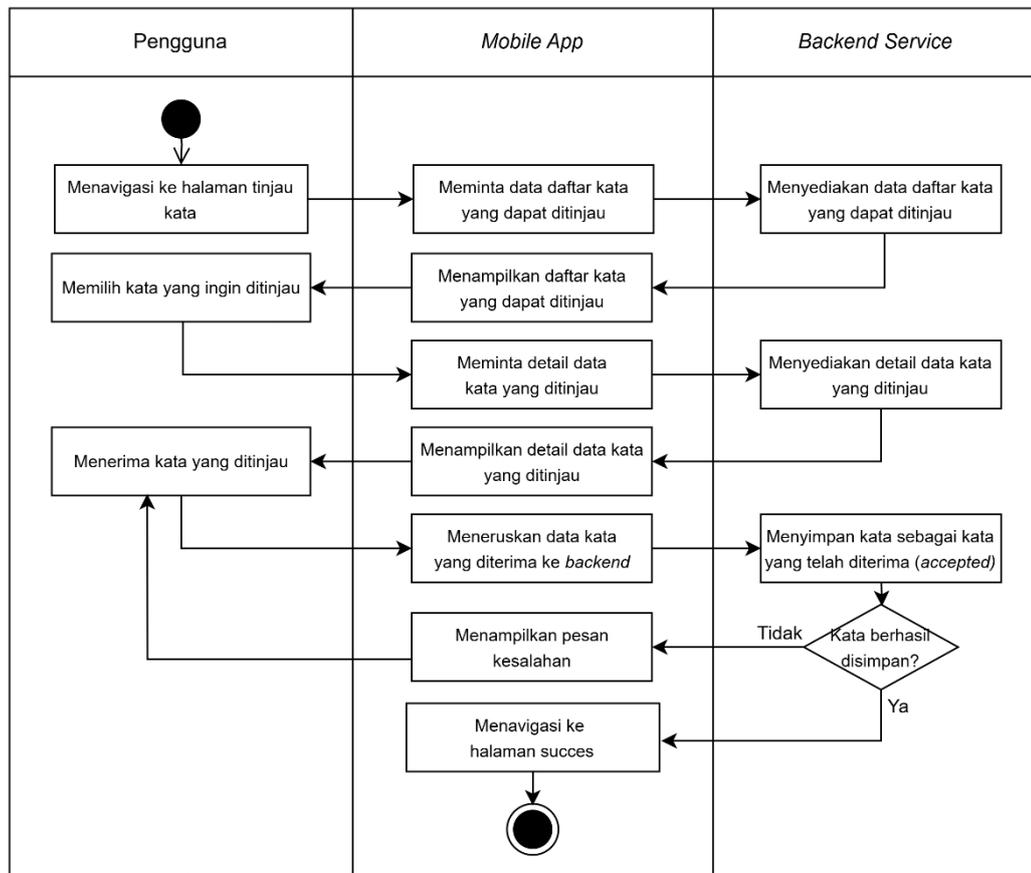
Merujuk pada gambar 3.12, ilustrasi tersebut menunjukkan alur proses melihat riwayat penambahan kata. Alur *activity* ini diawali dengan pengguna mengakses halaman riwayat kontribusi. Kemudian aplikasi meminta daftar riwayat kontribusi ke *server* dan *server* memberikan data yang diminta. Selanjutnya, jika pengguna memilih status riwayat lainnya aplikasi akan meminta kembali daftar riwayat berdasarkan status yang dipilih. Kemudian apabila pengguna melihat detail riwayat, aplikasi menavigasi ke halaman riwayat detail kontribusi.



Gambar 3.12 Activity Diagram melihat riwayat penambahan kata

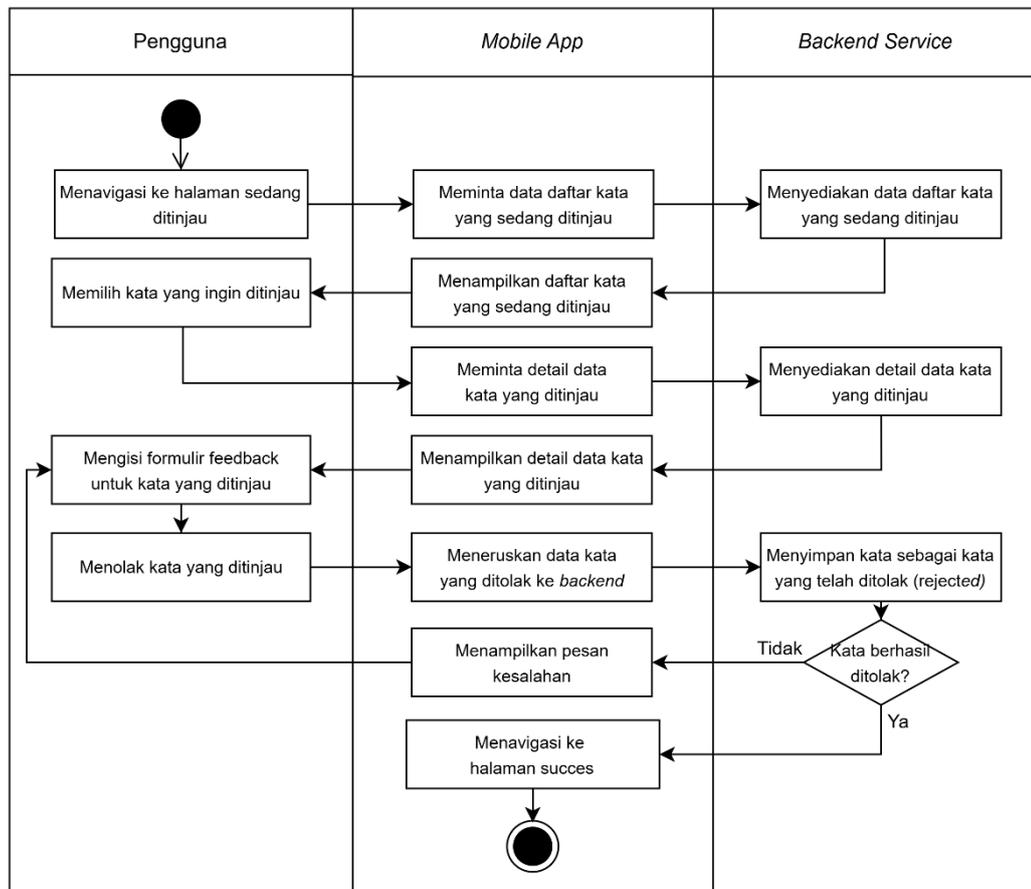
j. Activity Diagram tinjau kata

Merujuk pada gambar 3.13, ilustrasi tersebut menunjukkan alur proses meninjau kata dengan aksi menerima kata yang ditinjau. Alur *activity* ini diawali dengan pengguna mengakses halaman tinjau kata. Kemudian aplikasi meminta daftar kata yang dapat ditinjau ke *server* dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar kata yang dapat ditinjau dan pengguna memilih kata yang ingin ditinjau. Kemudian aplikasi meminta detail kata ke *server* dan menampilkannya. Selanjutnya pengguna memilih aksi menerima kata yang ditinjau. Kemudian data akan diteruskan ke *server*. Selanjutnya *server* akan menyimpan kata sebagai kata yang telah diterima. Apabila berhasil aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi menampilkan pesan kesalahan.



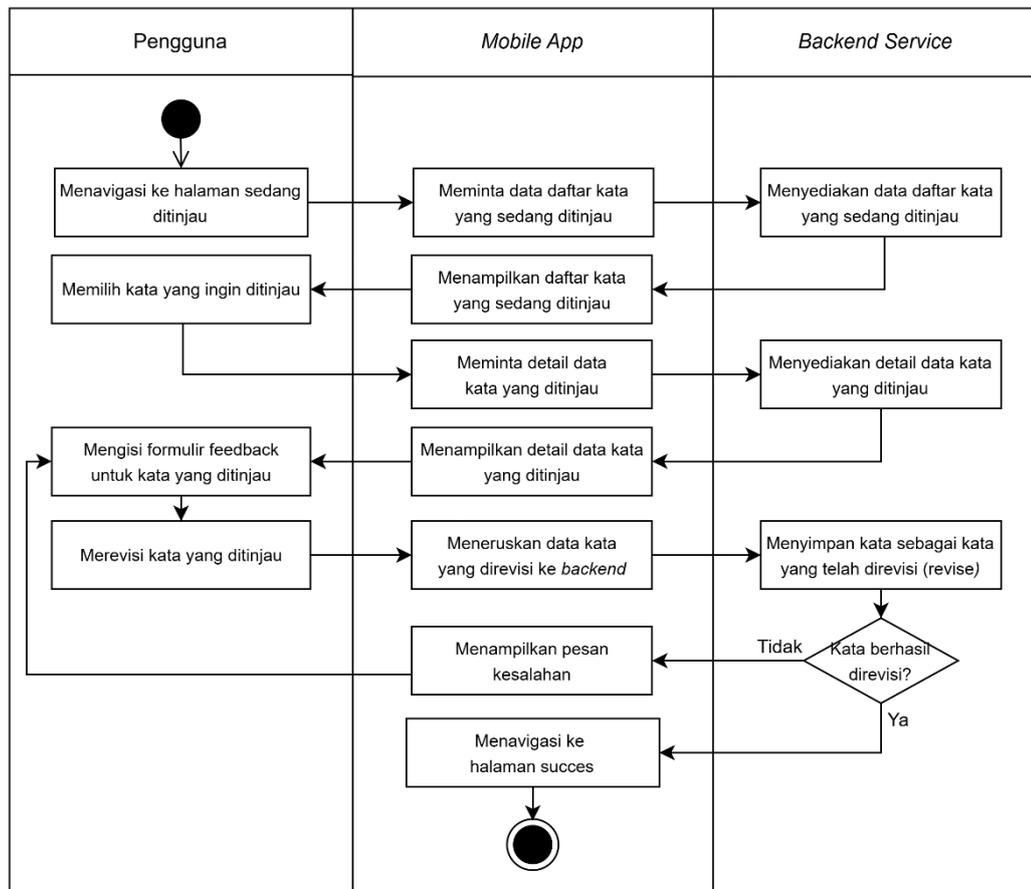
Gambar 3.13 Activity Diagram tinjau kata (menerima)

Merujuk pada gambar 3.14, ilustrasi tersebut menunjukkan alur proses meninjau kata dengan aksi menolak kata yang ditinjau. Alur *activity* ini diawali dengan pengguna mengakses halaman tinjau kata. Kemudian aplikasi meminta daftar kata yang dapat ditinjau ke *server* dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar kata yang dapat ditinjau dan pengguna memilih kata yang ingin ditinjau. Kemudian aplikasi meminta detail kata ke *server* dan menampilkannya. Selanjutnya pengguna memilih aksi menolak kata yang ditinjau. Kemudian data akan diteruskan ke *server*. Selanjutnya *server* akan menyimpan kata sebagai kata yang telah ditolak. Apabila berhasil aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi menampilkan pesan kesalahan.



Gambar 3.14 Activity Diagram tinjau kata (menolak)

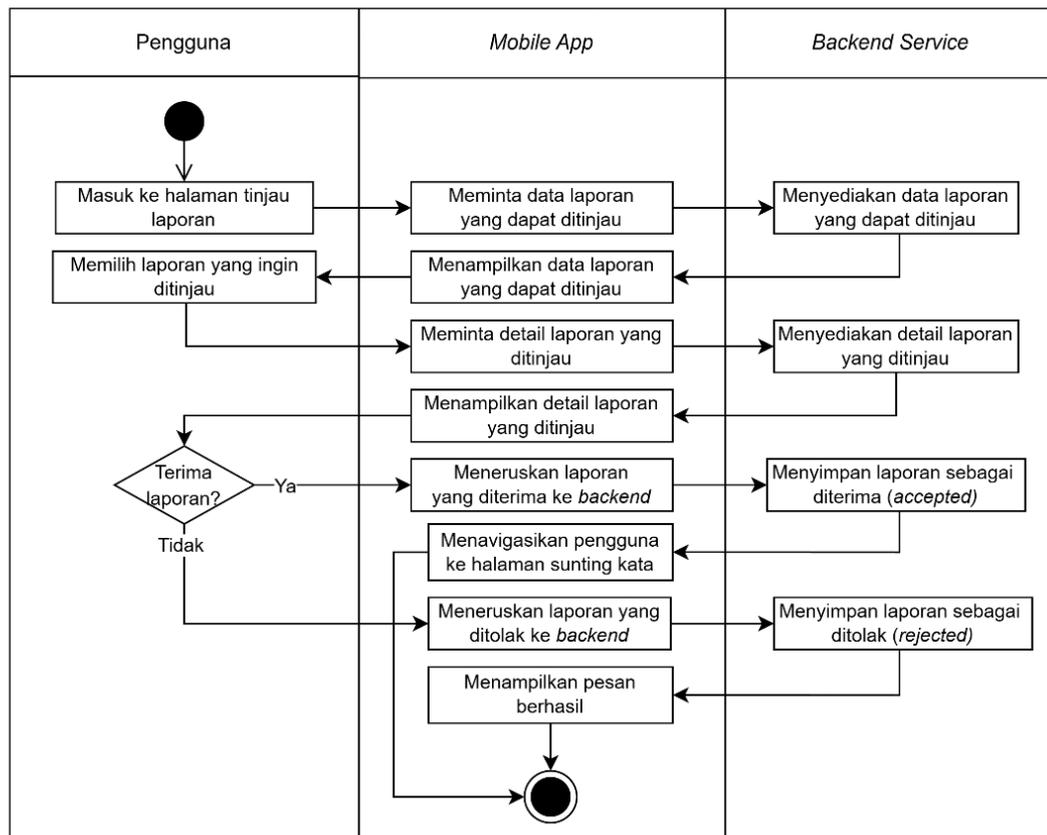
Merujuk pada gambar 3.15, ilustrasi tersebut menunjukkan alur proses meninjau kata dengan aksi merevisi kata yang ditinjau. Alur *activity* ini diawali dengan pengguna mengakses halaman tinjau kata. Kemudian aplikasi meminta daftar kata yang dapat ditinjau ke *server* dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar kata yang dapat ditinjau dan pengguna memilih kata yang ingin ditinjau. Kemudian aplikasi meminta detail kata ke *server* dan menampilkannya. Selanjutnya pengguna memilih aksi revisi kata yang ditinjau. Kemudian data akan diteruskan ke *server*. Selanjutnya *server* akan menyimpan kata sebagai kata yang direvisi. Apabila berhasil aplikasi akan menavigasi ke halaman *sukses*, namun apabila gagal aplikasi menampilkan pesan kesalahan.



Gambar 3.15 Activity Diagram tinjau kata (merevisi)

k. Activity Diagram tinjau laporan

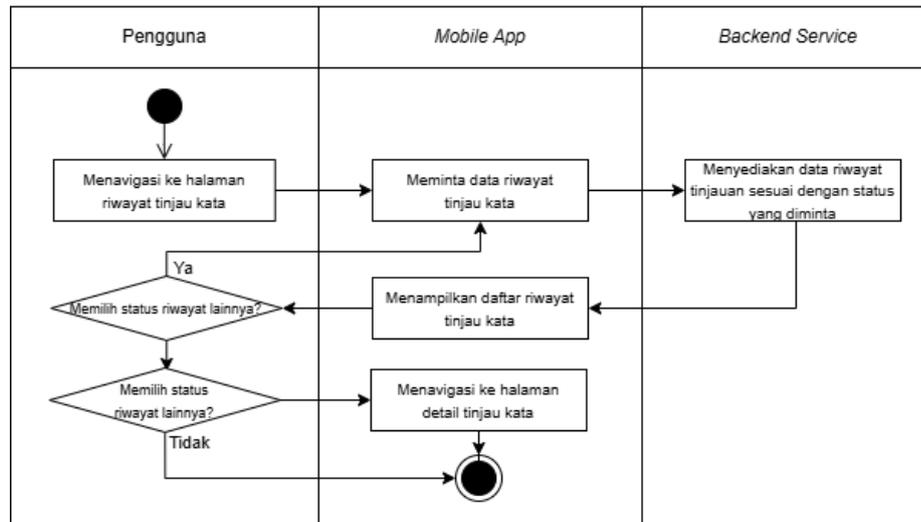
Merujuk pada gambar 3.16, ilustrasi tersebut menunjukkan alur proses meninjau laporan. Alur *activity* ini diawali dengan pengguna mengakses halaman tinjau laporan. Kemudian aplikasi meminta daftar laporan yang dapat ditinjau ke *server* dan *server* memberikan data yang diminta. Selanjutnya aplikasi menampilkan daftar laporan yang dapat ditinjau dan pengguna memilih laporan yang ingin ditinjau. Kemudian aplikasi meminta detail laporan ke *server* dan menampilkannya. Selanjutnya, apabila pengguna memilih aksi menerima laporan, aplikasi akan meneruskan laporan sebagai laporan yang diterima dan menavigasi ke halaman sunting kata. Kemudian, apabila pengguna menolak laporan, aplikasi meneruskan ke *server* dan aplikasi menampilkan pesan berhasil.



Gambar 3.16 Activity Diagram tinjau laporan

1. Activity Diagram melihat riwayat tinjau kata

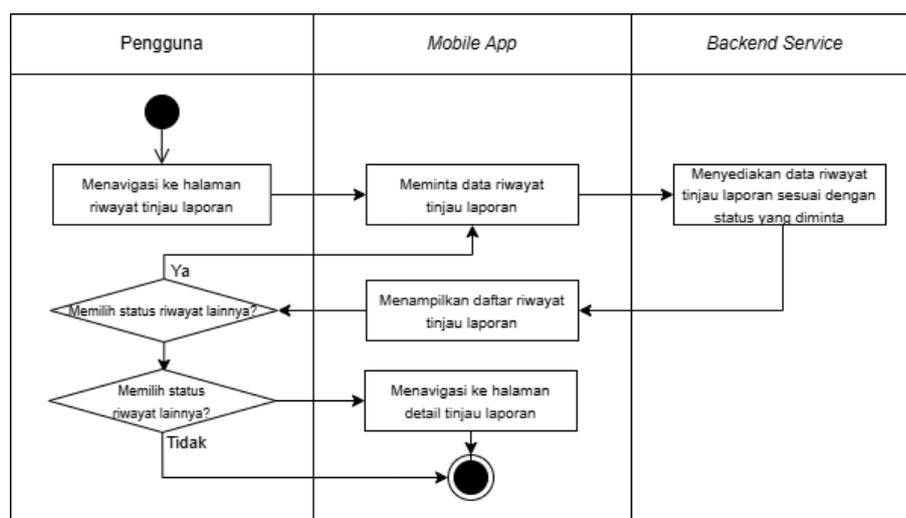
Merujuk pada gambar 3.17, ilustrasi tersebut menunjukkan alur proses melihat riwayat tinjau kata. Alur *activity* ini diawali dengan pengguna mengakses halaman riwayat tinjau kata. Kemudian aplikasi meminta daftar riwayat tinjau kata ke *server* dan *server* memberikan data yang diminta. Selanjutnya, jika pengguna memilih status riwayat lainnya aplikasi akan meminta kembali daftar riwayat berdasarkan status yang dipilih. Kemudian apabila pengguna melihat detail riwayat, aplikasi menavigasi ke halaman riwayat detail tinjau kata.



Gambar 3.17 Activity Diagram melihat riwayat tinjau kata

m. Activity Diagram melihat riwayat tinjau laporan

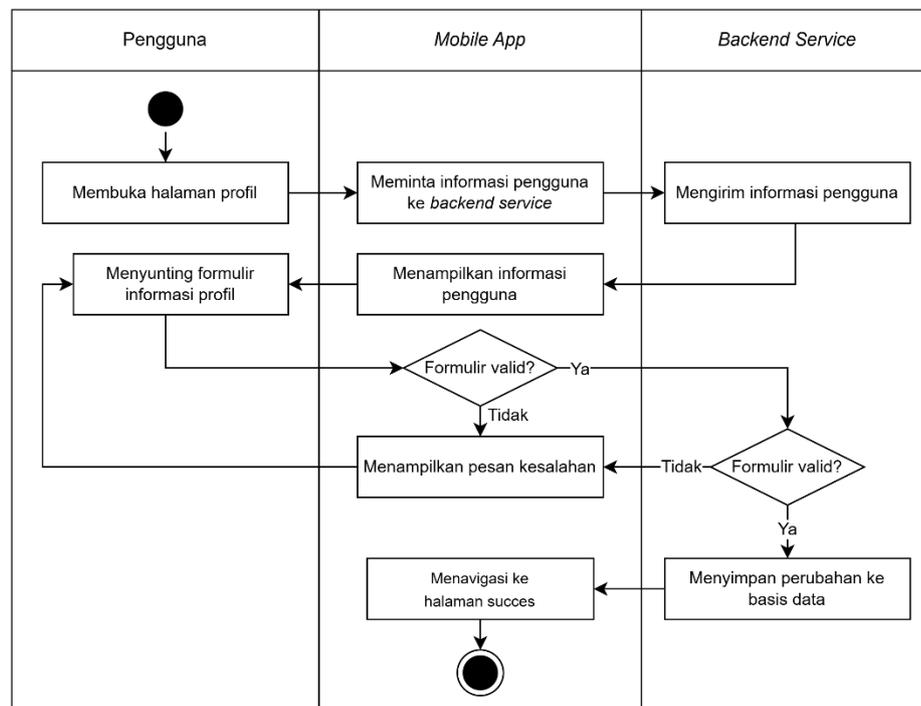
Merujuk pada gambar 3.18, ilustrasi tersebut menunjukkan alur proses melihat riwayat tinjau laporan. Alur *activity* ini diawali dengan pengguna mengakses halaman riwayat tinjau laporan. Kemudian aplikasi meminta daftar riwayat tinjau laporan ke *server* dan *server* memberikan data yang diminta. Selanjutnya, jika pengguna memilih status riwayat lainnya aplikasi akan meminta kembali daftar riwayat berdasarkan status yang dipilih. Kemudian apabila pengguna melihat detail riwayat, aplikasi menavigasi ke halaman riwayat detail tinjau laporan.



Gambar 3.18 Activity Diagram melihat riwayat tinjau laporan

n. *Activity Diagram* sunting profil

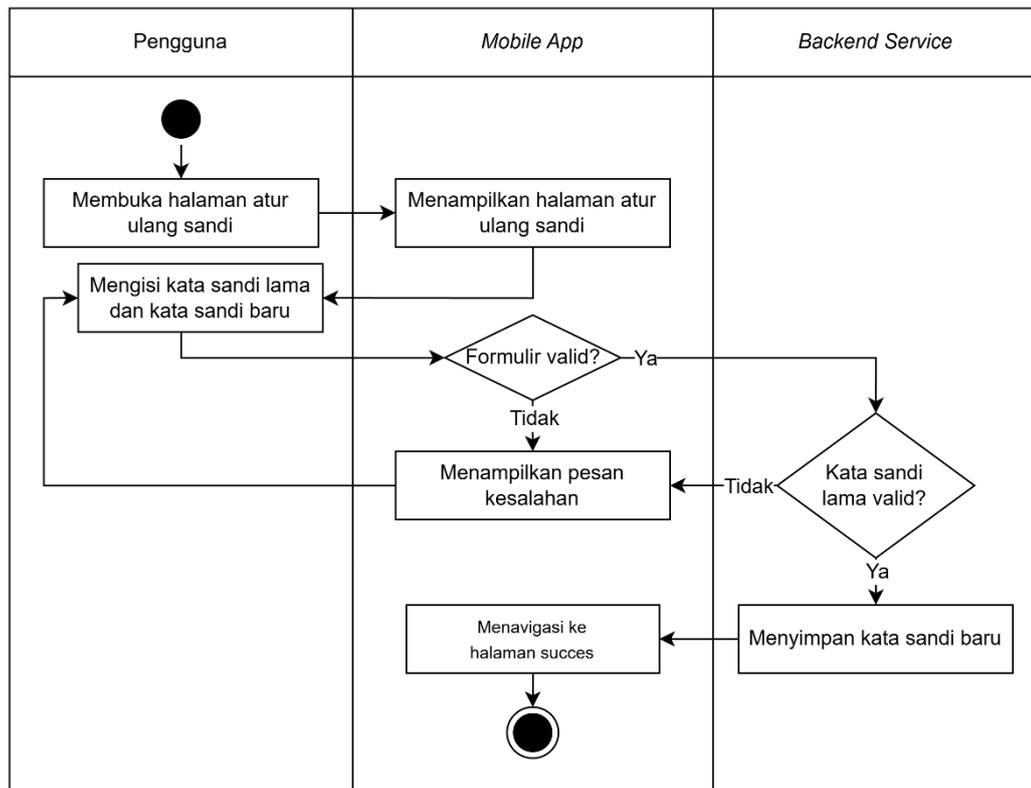
Merujuk pada gambar 3.19, ilustrasi tersebut menunjukkan alur proses menyunting profil. Alur *activity* ini diawali dengan pengguna mengakses halaman profil. Kemudian aplikasi meminta data informasi pengguna ke *server* dan *server* merespon dengan memberikan data informasi pengguna. Selanjutnya aplikasi menampilkan data informasi pengguna yang telah diperoleh. Kemudian pengguna menyunting informasi. Selanjutnya aplikasi dan *server* memvalidasi data yang dimasukkan. Apabila valid perubahan data informasi pengguna disimpan oleh *server* dan aplikasi menavigasi ke halaman *sukses*. Jika data tidak valid aplikasi akan menampilkan pesan kesalahan.



Gambar 3.19 *Activity Diagram* sunting profil

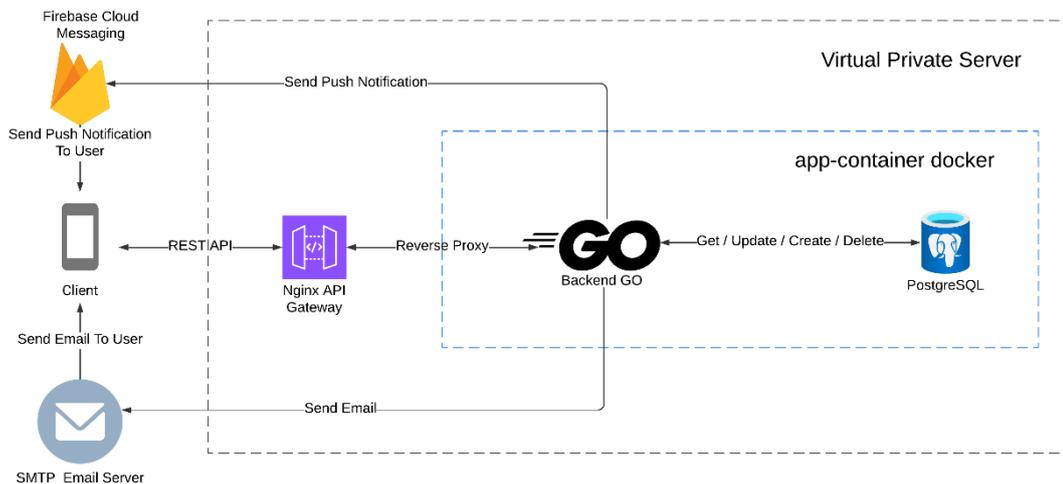
o. *Activity Diagram* atur ulang kata sandi

Merujuk pada gambar 3.20, ilustrasi tersebut menunjukkan alur proses atur ulang kata sandi. Alur *activity* ini diawali dengan pengguna mengakses halaman atur ulang kata sandi. Kemudian pengguna mengisi kata sandi lama dan baru. Selanjutnya aplikasi dan *server* memvalidasi data yang dimasukkan. Apabila valid perubahan kata sandi pengguna disimpan oleh *server* dan aplikasi menavigasi ke halaman *sukses*. Jika data tidak valid aplikasi akan menampilkan pesan kesalahan.



Gambar 3.20 Activity Diagram atur ulang kata sandi

### 3.4.1.5 Arsitektur Sistem



Gambar 3.21 Arsitektur Sistem

Arsitektur sistem pada gambar 3.21 menunjukkan bagaimana arsitektur sistem aplikasi kamus bahasa Lampung. Aplikasi *backend* dibangun menggunakan bahasa pemrograman Go dan dideploy di dalam *container* Docker pada *Virtual Private Server* (VPS). Untuk menangani lalu lintas dari *client*, sistem menggunakan *Ngix*

sebagai *API Gateway* dan *reverse proxy*, yang bertugas menerima permintaan *REST API* dari *client* dan meneruskannya ke *backend* Go. Kemudian *backend* memproses permintaan seperti mengambil, mengubah, atau menambahkan data ke dalam *database* PostgreSQL. Selain itu, sistem juga mendukung pengiriman notifikasi ke perangkat pengguna melalui *Firestore Cloud Messaging* (FCM) dan pengiriman *email* melalui *SMTP Email Server*.

### **3.4.2 *Product Backlog***

Pada tahap ini, dilakukan diskusi untuk menyusun *product backlog* dengan merujuk pada daftar kebutuhan pengguna yang telah diidentifikasi sebelumnya. Diskusi diawali dengan menetapkan *product goal* sebagai tujuan utama yang ingin dicapai oleh produk. Selanjutnya, *item-item backlog* yang mencakup fitur-fitur utama yang perlu dikembangkan, tugas-tugas teknis dan non-teknis, disusun guna mendukung proses pengembangan aplikasi.

### **3.4.3 *Sprint Planning***

Pada tahap ini tim pengembang dan *scrum master* berdiskusi untuk menentukan *backlog item* apa saja yang akan dikerjakan pada *sprint* berikutnya. Kemudian, *backlog item* yang telah dipilih diubah menjadi *task* yang lebih spesifik, terukur, dan dapat dikerjakan oleh tim selama *sprint*. Selanjutnya, *backlog item* yang dipilih dan *task* yang telah didefinisikan dimasukkan ke dalam *sprint backlog*.

### **3.4.4 *Sprint***

Pada tahap ini tim pengembang bekerja sesuai dengan *task* yang telah direncanakan dalam *sprint backlog* untuk menyelesaikan *backlog item*. Setelah *task* selesai, hasil pekerjaan diuji guna memastikan kualitas yang dihasilkan sesuai dengan ekspektasi yang ditetapkan pada *backlog item*. Pengujian dalam *sprint* dilakukan dengan menggunakan metode *Black Box*. Jika ditemukan kesalahan selama pengujian, akan dilakukan perbaikan hingga hasilnya memenuhi standar yang diharapkan. Setelah pekerjaan dinyatakan sesuai, hasil akhirnya disimpan ke dalam repositori *GitLab*.

#### **3.4.5 Daily Scrum**

Tahap ini dilakukan setiap hari untuk mengetahui kemajuan pada *sprint goal*. Tiap-tiap anggota tim *scrum*, menyampaikan kemajuan yang telah dicapai pada hari sebelumnya, hambatan yang terjadi di hari sebelumnya, dan merencanakan pekerjaan untuk dilakukan pada hari berjalan. Pada penelitian ini *daily scrum* dilaksanakan secara tidak langsung melalui grup khusus pada aplikasi WhatsApp. Pendekatan tersebut dipilih karena lebih mudah terdokumentasi dibandingkan dengan pertemuan secara langsung.

#### **3.4.6 Sprint Review**

Setelah menyelesaikan *sprint*, tim mengadakan *sprint review* untuk meninjau hasil kerja yang telah dicapai. Pada pertemuan ini, setiap anggota tim *scrum* akan mempresentasikan apa yang telah dikerjakan kepada *scrum master*. Pada tahap ini setiap anggota tim juga menyampaikan *product backlog* apa saja yang telah selesai dan yang tidak selesai.

#### **3.4.7 Retrospective**

Tahap ini dilakukan setelah mengadakan *sprint review*. Pada tahap ini akan dilakukan evaluasi proses kerja tim selama *sprint*, mengidentifikasi hal-hal yang berjalan dengan baik, dan menemukan hal-hal yang perlu ditingkatkan. Hasil dari *retrospective* ini digunakan untuk memperbaiki metode kerja pada *sprint* berikutnya, dengan tujuan peningkatan berkelanjutan.

#### **3.4.8 User Acceptance Test (UAT)**

Pada tahap ini sistem yang telah selesai dirancang diuji secara menyeluruh kepada pengguna akhir atau *end user*, guna memastikan kualitas dan kesesuaian terhadap spesifikasi yang telah ditetapkan. Pengujian akan diuji oleh pengguna akhir menggunakan aplikasi yang siap diluncurkan. Pengguna akhir untuk pengujian fitur pada peran peninjau adalah Kepala Program Studi Pendidikan Bahasa Lampung dan Dosen Program Studi Pendidikan Bahasa Lampung, yakni Bapak Munaris dan

Ibu Jesika Wulandari. Sedangkan, pengujian fitur untuk peran kontributor dan peran publik dilakukan oleh mahasiswa. Setelah mencoba aplikasi, pengguna akhir akan mengisi kuesioner untuk menyampaikan pengalamannya setelah melakukan pengujian. Kuesioner ini mencakup pertanyaan mengenai reaksi pengguna terhadap sistem, antarmuka pengguna, kemudahan dalam mengoperasikan sistem, dan fungsionalitas yang tersedia. Hasil kuesioner kemudian dianalisis menggunakan skala *likert* untuk mengukur pendapat atau persepsi individu terhadap sistem.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Adapun kesimpulan yang diperoleh dari penelitian yang telah dilakukan adalah sebagai berikut.

1. Aplikasi kamus bahasa Lampung berbasis komunitas telah berhasil dikembangkan menggunakan metode *scrum* dalam 9 *sprint* dengan total 19 *product backlog*.
2. Hasil pengujian dengan metode *Black Box* yang dilakukan dalam dua iterasi menunjukkan bahwa aplikasi telah berfungsi dengan baik setelah diuji dengan 63 skenario. Pada iterasi pertama, tingkat keberhasilan mencapai 84,12% atau 53 skenario yang berhasil, kemudian setelah dilakukan perbaikan, meningkat menjadi 100% atau 63 skenario yang berhasil.
3. Berdasarkan hasil pengujian menggunakan metode *User Acceptance Test* (UAT), aplikasi kamus bahasa Lampung berbasis komunitas dapat diterima dan layak digunakan oleh pengguna dengan sangat baik. Hasil akumulasi dari tiga kelompok peran diperoleh rata-rata 95,02%, di antaranya penguji berperan sebagai peninjau (2 penguji) memperoleh rata-rata 100%, penguji berperan sebagai kontributor (10 penguji) memperoleh rata-rata 91,85%, dan penguji berperan sebagai publik (10 penguji) memperoleh rata-rata 93,22%.

### 5.2 Saran

Adapun saran yang dapat diterapkan pada pengembangan selanjutnya adalah mengembangkan fitur penerjemahan kalimat, pemindaian teks aksara, serta gamifikasi dalam proses pembelajaran. Hal ini bertujuan untuk meningkatkan efektivitas pembelajaran bahasa Lampung agar lebih interaktif dan menarik.

## DAFTAR PUSTAKA

- [1] D. F. Maulana, D. A. Wijaya, D. A. Agustinova, dan Suharno, “Pendidikan Demokrasi dan Pendidikan Multikultural: Tinjauan Praktis di Sekolah Menengah Pertama,” *Edukatif: Jurnal Ilmu Pendidikan*, vol. 6, hlm. 1512–1526, 2024, doi: <https://doi.org/10.31004/edukatif.v6i2.6458>.
- [2] Badan Pusat Statistik, “Statistik Sosial Budaya 2021.” Diakses: 14 September 2024. [Daring]. Tersedia pada: <https://www.bps.go.id/id/publication/2022/06/30/6a2dabc16d556ab9d075f918/statistik-sosial-budaya-2021.html>
- [3] B. Purwaka, W. Permanawiyat, A. O. Anindyatri, dan I. Mufidah, *Gambaran kondisi vitalitas bahasa daerah di Indonesia : berdasarkan data tahun 2018-2019*. 2020. [Daring]. Tersedia pada: <http://repositori.kemdikbud.go.id/id/eprint/22796>
- [4] F. H. Tondo, “Kepunahan Bahasa-Bahasa Daerah: Foktor Penyebab dan Implikasi Etnolinguistik,” *Jurnal Masyarakat & Budaya*, vol. 11, no. 2, hlm. 277–296, 2009, doi: <https://doi.org/10.14203/jmb.v11i2.245>.
- [5] N. W. Putri, “Pergeseran Bahasa Dearah Lampung Pada Masyarakat Kota Bandar Lampung,” *PRASASTI: Journal of Linguistics*, vol. 3, no. 1, hlm. 83–97, 2018, doi: <https://doi.org/10.20961/prasasti.v3i1.16550>.
- [6] N. E. Rusminto, F. Ariyani, A. B. Setiyadi, dan G. E. Putrawan, “Local language vs. National Language: The lampung Language Maintenance in The Indonesian Context,” *Kervan - International Journal of Afro-Asiatic Studies*, vol. 25, no. 1, hlm. 287–307, 2021, doi: [10.13135/1825-263X/5787](https://doi.org/10.13135/1825-263X/5787).

- [7] Kemendikbudristek, *Statistik Kebahasaan dan Kesastraan 2023*, vol. 5. 2023. [Daring]. Tersedia pada: <http://repositori.kemdikbud.go.id/id/eprint/29024>
- [8] N. A. Rahardini dan A. A. Niswah, “Revitalisasi Bahasa Minoritas di Indonesia,” *Jurnal Etnolingual*, vol. 6, no. 2, hlm. 113–134, 2022, doi: <https://doi/10.20473/etno.v6i2.41287>.
- [9] Z. A. Aliana, S. Nursato, S. S. Arifin, S. Soetopo, dan M. Waif, “Ragam Dan Dialek Bahasa Lampung,” hlm. v–144, 1985, [Daring]. Tersedia pada: <http://repositori.kemdikbud.go.id/id/eprint/3474>
- [10] W. Warkim, M. H. Muslim, F. Harvianto, dan S. Utama, “Penerapan Metode SCRUM dalam Pengembangan Sistem Informasi Layanan Kawasan,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, hlm. 365–378, 2020, doi: [10.28932/jutisi.v6i2.2711](https://doi.org/10.28932/jutisi.v6i2.2711).
- [11] Scrum.org, “The 2020 Scrum Guide.” Diakses: 17 September 2024. [Daring]. Tersedia pada: <https://scrumguides.org/index.html>
- [12] M. T. Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 1, hlm. 126–129, 2018, doi: [10.30591/jpit.v3i1.765](https://doi.org/10.30591/jpit.v3i1.765).
- [13] A. Verma, A. Khatana, dan S. Chaudhary, “A Comparative Study of Black Box Testing and White Box Testing,” *International Journal of Computer Sciences and Engineering*, vol. 5, no. 12, hlm. 301–304, 2017, doi: [10.26438/ijcse/v5i12.301304](https://doi.org/10.26438/ijcse/v5i12.301304).
- [14] F. F. Nursaid, A. Hendra Brata, dan A. P. Kharisma, “Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 1, hlm. 46–55, 2020, [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6859>

- [15] Visual Studio Code, “Visual Studio Code - Code Editing.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://code.visualstudio.com/>
- [16] Git, “About Us.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://git-scm.com/site>
- [17] GitLab, “About GitLab Behind the scenes of The DevSecOps Platform.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://about.gitlab.com/company/>
- [18] M. Ichwan dan F. Hakiky, “Pengukuran Kinerja Goodreads Application Programming Interface (API) Pada Aplikasi Mobile Android (Studi Kasus Untuk Pencarian Data Buku),” *Jurnal Informatika*, vol. 2, no. 2, hlm. 13–21, 2011, [Daring]. Tersedia pada: <https://lib.itenas.ac.id/kti/?p=2220>
- [19] NodeJs, “About NodeJS.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://nodejs.org/en/about>
- [20] React Native, “React Native Learn once, write anywhere.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://reactnative.dev/>
- [21] M. Kuitunen, “Cross-Platform Mobile Application Development with React Native,” *Tamperey University*, no. February, hlm. 1–26, 2019, doi: <https://urn.fi/URN:NBN:fi:tty-201902111234>.
- [22] A. E. Fentaw, “Cross Platform Mobile Application Development: a Comparison Study of React Native Vs Flutter,” *Jurnal University of Jyväskylä*, vol. 27791, hlm. 37–38, 2020, [Daring]. Tersedia pada: [https://jyx.jyu.fi/jyx/Record/jyx\\_123456789\\_70969](https://jyx.jyu.fi/jyx/Record/jyx_123456789_70969)
- [23] Expo, “Expo and EAS are an Ecosystem of Tools That Help you Develop, Review, and Deploy.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://expo.dev/>
- [24] F. N. Izza dan R. A. Rajagede, “Pengembangan Permainan Tradisional Dam-Daman dalam Bentuk Application Programming Interface,” *Automata*, vol. 3, no. 2, hlm. 1–6, 2022, [Daring]. Tersedia pada: <https://journal.uui.ac.id/AUTOMATA/article/download/24329/14058>

- [25] H. G. Simanullang, A. P. Silalahi, dan D. R. Manalu, “Sistem Informasi Pendaftaran Mahasiswa Baru Menggunakan Framework Codeigniter dan Application Programming Interface,” *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, vol. 12, no. 1, hlm. 67–73, 2021, doi: 10.31937/si.v12i1.1803.
- [26] Google Play, “Cara kerja Google Play.” Diakses: 14 Januari 2025. [Daring]. Tersedia pada: [https://play.google/intl/id\\_id/howplayworks/](https://play.google/intl/id_id/howplayworks/)
- [27] Taiga, “Open Source and Lean principles gave birth to Taiga.” Diakses: 16 September 2024. [Daring]. Tersedia pada: <https://taiga.io/about-us/>
- [28] F. E. Febriansyah, A. Ardiansyah, dan A. Darmaji, “Cawa Lampung : Kamus Bahasa Indonesia-Lampung Dialek a Berbasis Android,” *Klik - Kumpulan Jurnal Ilmu Komputer*, vol. 7, no. 3, hlm. 331, 2020, doi: 10.20527/klik.v7i3.352.
- [29] S. N. Azima, Muhammad Fauzan Laila, “Rancang Bangun Aplikasi Kamus Bahasa dan Aksara Lampung Dialek A dan Dialek O Berbasis Android,” *Jurnal Teknika*, vol. 14, no. x, hlm. 1–9, 2020, doi: <https://doi.org/10.5281/zenodo.13346173>.
- [30] R. Hidayatullah, V. Asmawati, dan D. Wahyuni, “Aplikasi Manajemen Surat Masuk dan Surat Keluar Berbasis Mobile di Perum Bulog Subdivre Dumai,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 2, hlm. 66–80, 2020, doi: 10.14421/jiska.2020.52-02.
- [31] Z. I. Sholihati dan I. Tahyudin, “Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, hlm. 169–176, 2022, doi: 10.29207/resti.v6i1.3793.
- [32] M. S. Ravi, “Comparative Analysis of Scrum and Kanban Methodologies in Agile Software Development,” vol. 1350, no. 515, 2023, doi: <https://doi.org/10.31274/cc-20240624-389>.

- [33] N. C. Ramachandrappa, “A Comparative Analysis of Native vs React Native Mobile App Development,” vol. 72, no. 9, hlm. 57–62, 2024, doi: <https://doi.org/10.14445/22312803/IJCTT-V72I9P110> Volume.
- [34] Supriyono, “Software Testing with the approach of Black Box Testing on the Academic Information System,” *International Journal of Information System & Technology*, vol. 3, no. 36, hlm. 227–235, 2020.
- [35] N. A. Nik Ahmad dan P. N. N. A. Megat Sazali, “Performing User Acceptance Test with System Usability Scale for Graduation Application,” *Proceedings - 2021 International Conference on Software Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021*, no. August, hlm. 86–91, 2021, doi: 10.1109/ICSECS52883.2021.00023.
- [36] I. Afrianto, A. Heryandi, A. Finandhita, dan S. Atin, “User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program,” *Internation Journal of Information System & Technology*, vol. 5, no. 3, hlm. 270–280, 2021, [Daring]. Tersedia pada: <https://doi.org/10.30645/ijistech.v5i3>
- [37] N. Hartono dan A. A. Muin, “Penggunaan User Acceptance Testing ( UAT ) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang,” *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 1, hlm. 84–100, 2025, doi: <https://doi.org/10.62951/switch.v3i1.330>.