

**RANCANG BANGUN SISTEM NAVIGASI PADA
APLIKASI ANDROID DENGAN *ROUTE GUIDANCE*
UNTUK TUNANETRA BERBASIS *INDOOR*
*POSITIONING***

TUGAS AKHIR

Diajukan untuk melengkapi tugas-tugas dan
memenuhi syarat-syarat guna pelaksanaan penelitian Tugas Akhir

Oleh:

**YUDA ADITYA
1608107010030**



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
AGUSTUS, 2023**

PENGESAHAN TUGAS AKHIR

**RANCANG BANGUN SISTEM NAVIGASI PADA APLIKASI ANDROID
DENGAN *ROUTE GUIDANCE* UNTUK TUNANETRA BERBASIS *INDOOR
POSITIONING***

***DESIGN AND IMPLEMENTATION OF ROUTE GUIDANCE APPLICATION
ON ANDROID BASED NAVIGATION SYSTEM FOR VISUALLY IMPAIRED
INDIVIDUALS***

Oleh:

Nama : Yuda Aditya
NPM : 1608107010030
Jurusan : Informatika

Menyetuji:

Pembimbing I

Pembimbing II

Kurnia Saputra, S.T., M.Sc.
NIP. 198003262014041001

Dalila Husna Yunardi, B.Sc., M.Sc.
NIP. 199006172015042001

Mengetahui:

Ketua Jurusan Informatika FMIPA
Universitas Syiah Kuala,

Viska Mutiawani, B.IT, M.IT
NIP. 198008312009122003

ABSTRAK

Aplikasi navigasi yang dikhawasukan untuk individu dengan gangguan penglihatan, menggunakan teknologi Bluetooth Low Energy (BLE), API VOSK untuk pengenalan ucapan, dan filter Kalman untuk meningkatkan pelacakan posisi. Aplikasi yang diusulkan bertujuan untuk memberikan sistem navigasi yang dapat digunakan untuk individu dengan gangguan penglihatan bergerak di lingkungan dalam dan luar dengan mudah dan mandiri. Penelitian ini memanfaatkan teknologi BLE untuk menghubungkan perangkat seluler pengguna dengan beacon yang ditempatkan strategis di sekitar lingkungan Gedung FMIPA Universitas Syiah Kuala. BLE mengirimkan informasi yang spesifik terkait lokasi kepada perangkat pengguna, memungkinkan penentuan posisi dalam ruangan dengan presisi serta navigasi yang akurat. API VOSK memungkinkan pengenalan ucapan secara real-time, memungkinkan pengguna memberikan perintah suara dan menerima umpan balik audio mengenai lokasi saat ini dan petunjuk navigasi. Kalman filter digunakan untuk meningkatkan akurasi sistem pelacakan posisi. Terdapat 3 pengujian utama pada penelitian ini meliputi pengujian akurasi rute navigasi menggunakan Kalman Filter dan Speech Command Recognition, pengujian fungsionalitas aplikasi dan pengujian usabilitas aplikasi. Berdasarkan pengujian rute navigasi menggunakan Kalman filter MSE (Mean Squared Error) pemilihan rute dipengaruhi oleh kekuatan sinyal BLE, lokasi pengguna, tembok, dan saran rute yang diberikan. Semakin rendah MSE maka semakin akurat rute yang diberikan. Pengujian fungsionalitas aplikasi dilakukan dengan menggunakan Black Box Testing mendapatkan hasil bahwa aplikasi yang dibangun berhasil berjalan dengan baik. Hasil yang didapatkan dari pengujian usabilitas menggunakan Usability Matrix for User Experience (UMUX) mendapatkan skor 78,33% sehingga aplikasi dapat diterima oleh pengguna.

Kata kunci : *Bluetooth Low Energy, Indoor Positioning System, Fingerprinting, VOSK API, Kalman Filter, Black Box, UMUX.*

ABSTRACT

The proposed navigation application is designed specifically for individuals with visual impairments, utilizing Bluetooth Low Energy (BLE) technology, VOSK API for speech recognition, and Kalman filter for improved position tracking. The aim of this application is to provide a reliable and intuitive navigation system for individuals with visual impairments to navigate indoor and outdoor environments confidently and independently. This research leverages BLE technology to connect the user's mobile device with strategically placed beacons in the environment of the FMIPA Building, Syiah Kuala University. BLE provides specific location information to the user's device, enabling precise indoor positioning and accurate navigation. The integration of the VOSK API allows real-time speech recognition, allowing users to give voice commands and receive audio feedback regarding their current location and navigation instructions. This voice-based interaction eliminates the need for manual input and enhances safety and efficiency. The Kalman filter is utilized to improve the accuracy of the position tracking system. The research includes three main tests: accuracy testing of navigation routes using the Kalman filter and speech command recognition, functional testing of the application, and usability testing of the application. Based on the navigation route testing using the Kalman filter's Mean Squared Error (MSE), the route selection is influenced by BLE signal strength, user location, obstacles, and provided route suggestions. A lower MSE indicates more accurate route guidance. Functional testing of the application using Black Box Testing confirms that the developed application performs well. The usability testing results using the Usability Matrix for User Experience (UMUX) obtained a score of 78.33%, indicating the application's acceptance by users.

Keywords : *Bluetooth Low Energy, Indoor Positioning System, Fingerprinting, VOSK API, Kalman Filter, Black Box, UMUX.*

KATA PENGANTAR

Segala puji dan syukur kepada Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya kepada kita semua dan juga atas izin-Nya penulis dapat menyelesaikan penulisan Tugas Akhir ini. Tak lupa Shalawat dan Salam penulis sanjung sajikan kepada Nabi Besar Muhammad SAW, karena beliau telah membawa kita semua dari alam jahiliah ke alam yang penuh ilmu pengetahuan.

Tugas Akhir yang berjudul “**RANCANG BANGUN SISTEM NAVIGASI PADA APLIKASI ANDROID DENGAN ROUTE GUIDANCE UNTUK TUNANETRA BERBASIS INDOOR POSITIONING**” ini telah dapat diselesaikan atas bantuan banyak pihak. Oleh karena itu, melalui tulisan ini penulis ingin mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada:

1. Orang tua serta keluarga penulis yang senantiasa selalu mendukung aktivitas dan kegiatan penulis lakukan baik secara moral maupun material serta menjadi motivasi terbesar bagi penulis untuk menyelesaikan Tugas Akhir ini.
2. Bapak Kurnia Saputra, S.T., M.Sc., selaku Dosen Pembimbing I dan Ibu Dalila Husna Yunardi, B.Sc., M.Sc., selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Dr. Muhammad Subianto, S.Si., M.Si., selaku Ketua Jurusan Informatika.
4. Bapak Kurnia Saputra, S.T., M.Sc., selaku Dosen Wali penulis.
5. Seluruh Dosen di Jurusan Informatika Fakultas MIPA atas ilmu dan didikannya selama perkuliahan.
6. Andika Pratama, Budi Gunawan, Fauzy Nisa, dan Muammar Zikri Aksana selaku teman yang telah banyak memberikan dukungan, masukan serta ilmu yang cukup besar dan bermanfaat dalam penulisan Tugas Akhir ini.
7. Seluruh teman-teman seperjuangan Jurusan Informatika Unsyiah 2016 lainnya.

Penulis juga menyadari segala ketidak sempurnaan yang terdapat didalamnya baik dari segi materi, cara, ataupun bahasa yang disajikan. Seiring dengan ini penulis mengharapkan kritik dan saran dari pembaca yang sifatnya dapat berguna untuk kesempurnaan Tugas Akhir ini. Harapan penulis semoga tulisan ini dapat bermanfaat bagi banyak pihak dan untuk perkembangan ilmu pengetahuan.

Banda Aceh, Agustus 2023

Penulis

DAFTAR ISI

PENGESAHAN TUGAS AKHIR	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
BAB II TINJAUAN KEPUSTAKAAN	5
2.1. <i>Indoor Positioning System</i>	5
2.2. <i>Route Guidance System / Wayfinding and Navigation System</i>	6
2.3. <i>Bluetooth Low Energy (BLE)</i>	6
2.3.1. <i>Beacon</i>	7
2.4. <i>Fingerprinting</i>	7
2.4.1. <i>Received Signal Strength Indicator (RSSI)</i>	8
2.4.2. <i>Kalman Filter</i>	9
2.5. <i>Speech Recognition</i>	11
2.5.1. <i>Text-to-Speech(TTS)</i>	11
2.6. <i>VOSK API</i>	11
2.7. <i>Android</i>	12
2.8. <i>Android Studio</i>	12
2.9. <i>SCRUM</i>	13
2.10. <i>Black Box Testing</i>	15
2.11. <i>Usability Testing</i>	16
2.11.1. <i>Usability Metric for User Experience (UMUX)</i>	16
2.12. Penelitian Terkait	18
BAB III METODOLOGI PENELITIAN	19
3.1. Waktu dan Lokasi Penelitian	19
3.2. Alat dan Bahan	19
3.3. <i>Roadmap</i> Penelitian	20
3.4. Metode Penelitian	23
3.4.1. Identifikasi Masalah	24
3.4.2. Studi Literatur	24

3.4.3. Analisa Kebutuhan	24
3.4.4. Perancangan Sistem	25
3.4.5. Pengumpulan Data	25
3.4.6. Pembuatan Sistem Aplikasi	28
3.4.7. Pengujian Akurasi	28
3.4.8. Pengujian Fungsionalitas dengan Metode <i>Black Box</i>	29
3.4.9. <i>Usability Testing</i>	30
BAB IV HASIL DAN PEMBAHASAN	31
4.1. Perancangan Sistem	31
4.2. Pembuatan Sistem	37
4.2.1. Persiapan data	37
4.2.2. Pembuatan Aplikasi	40
4.3. Pengujian Sistem	45
4.3.1. Akurasi rute	45
4.3.2. Pengujian Usabilitas UMUX	46
4.3.3. Pengujian Fungsionalitas Menggunakan BlackBox	47
BAB V KESIMPULAN DAN SARAN	48
5.1. Kesimpulan	48
5.2. Saran	49
DAFTAR KEPUSTAKAAN	52

DAFTAR TABEL

Tabel 2.1	Kekuatan sinyal RSSI (Sideeq dan Qasem, 2016)	9
Tabel 2.2	Daftar Pertanyaan Metode UMUX menurut (Lewis, 2018). . .	16
Tabel 3.1	Daftar Pertanyaan Metode UMUX.	30
Tabel 4.1	Data Ruangan & Perintah	37
Tabel 4.2	Skema Pengujian Akurasi Pengambilan Rute di Gedung A	
	FMIPA Optimal dalam MSE (<i>Mean Squared Error</i>)	45
Tabel 4.3	Hasil Pengujian UMUX Aplikasi Navigasi	46
Tabel 4.4	Pengujian Blackbox Aplikasi Navigasi	47

DAFTAR GAMBAR

Gambar 2.1	Perangkat Bluetooth Low Energy.	6
Gambar 2.2	Ilustrasi Metode <i>Fingerprinting</i> (Yudha dkk., 2018)	8
Gambar 2.3	Ilustrasi Kalman Filter (Ihsan dkk., 2018)	9
Gambar 2.4	Ilustrasi Metode Pengembangan Menggunakan Scrum (Schwaber dan Sutherland, 2011).	15
Gambar 3.1	<i>Roadmap</i> Penelitian Fase 1	21
Gambar 3.2	<i>Roadmap</i> Penelitian Fase 2	22
Gambar 3.3	Diagram Alir Penelitian	23
Gambar 3.4	Denah lantai 1 Gedung A FMIPA	26
Gambar 3.5	Denah lantai 2 Gedung A FMIPA	26
Gambar 3.6	Denah lantai 3 Gedung A FMIPA	27
Gambar 4.1	<i>Use Case Diagram</i>	31
Gambar 4.2	<i>Deployment Diagram</i>	32
Gambar 4.3	<i>Class Diagram</i>	33
Gambar 4.4	Alur Kerja Sistem	34
Gambar 4.5	<i>Flow Diagram</i> Navigasi Indoor	35
Gambar 4.6	Tampilan Halaman Aplikasi Navigasi Indoor untuk Tunanetra	36
Gambar 4.7	Denah lantai 1 Gedung A FMIPA dengan BLE	39
Gambar 4.8	Denah lantai 2 Gedung A FMIPA dengan BLE	40
Gambar 4.9	Denah lantai 3 Gedung A FMIPA dengan BLE	40
Gambar 4.10	Potongan Code untuk Implementasi Talkback	41
Gambar 4.11	Potongan Code untuk perhitungan Kalman Filter	42
Gambar 4.12	Potongan Code untuk perhitungan Akurasi Rute	43
Gambar 4.13	Tampilan Halaman Aplikasi Navigasi Indoor untuk Tunanetra	44
Gambar 4.14	Skor UMUX ke SUS	46

BAB I

PENDAHULUAN

1.1. Latar Belakang

Manusia adalah makhluk sosial yang selalu berinteraksi dan berkomunikasi dengan manusia lainnya, berbagai cara penyampaian informasi dan komunikasi dilakukan untuk saling terhubung dengan lingkungan sekitar. Cara berkomunikasi yang paling sering dilakukan oleh manusia adalah berbicara atau menggunakan media suara, dan beberapa cara lain seperti tulisan, isyarat serta media visual seperti gambar. Dengan adanya suara juga dapat membantu manusia yang memiliki keterbatasan penglihatan, sebagai pengganti indra penglihatan mereka dengan mengandalkan serta mempertajam indra pendengaran mereka. Menurut *World Health Organization* disebutkan dalam laporan *World Report on Vision*, secara global setidaknya sekitar 2,2 Miliar orang diantaranya memiliki gangguan penglihatan atau kebutaan, dimana setidaknya 1 miliar di antaranya memiliki gangguan penglihatan yang dapat dicegah atau belum ditangani (WHO, 2020).

Perkembangan teknologi yang sangat pesat telah memudahkan orang dengan keterbatasan penglihatan dalam berbagai aspek kehidupan, seperti dengan adanya teknologi berbasis suara dapat membantu orang-orang dengan keterbatasan penglihatan dalam berbagai aspek kehidupan seperti, berkomunikasi serta bekerja sebagai pemandu jalan atau arah. Selain berkomunikasi, pemandu arah juga menjadi salah satu kebutuhan penting bagi orang-orang dengan keterbatasan penglihatan bahkan tidak hanya mereka. Untuk mengurangi kesulitan pengguna tunanetra dan untuk memastikan posisi pengguna serta menemukan destinasi tujuannya dibutuhkan teknologi berupa *Global Positioning Systems* (GPS). Namun, saat ini sistem yang menggunakan sensor GPS bekerja dengan baik sebagai alat pencari rute di luar ruangan, sehingga tidak memadai untuk memandu orang di dalam ruangan (Ko dan Kim, 2017). Oleh karena itu tujuan dari penelitian ini adalah untuk membangun sistem pencarian rute yang efektif dengan menggunakan *Indoor Positioning System*.

Indoor Positioning System (IPS) merupakan teknologi untuk melacak suatu objek atau individu di area tertutup atau gedung merupakan bagian dari Location Based Service (LBS) (Brena dkk., 2017). LBS adalah layanan yang menyediakan informasi bagi pengguna berdasarkan lokasi pengguna (Rawat dan Ghafoor, 2018). Teknologi IPS ini nantinya akan digunakan sebagai dasar dari *Wayfinding System* atau *Route Guidance System* dan akan dipadukan dengan *Speech Command Recognition*.

Speech Recognition atau *Automatic Speech Recognition (ASR)* merupakan salah satu bentuk dari teknologi yang disebut *Artificial Intelligence (AI)*. ASR sendiri merupakan teknologi pada perangkat komputer agar dapat mengenali dan memahami kata-kata yang diucapkan oleh manusia. Pada sistem *Speech Recognition* memiliki 2 sistem utama yaitu *Speech-to-Text (STT)* dan *Text-to-Speech (TTS)*. *Speech Recognition* juga dikenal sebagai sistem *Speech-to-Text* dimana suara diterima dan diterjemahkan sebagai teks agar dapat dikenali (Mustikarini dkk., 2019). Sedangkan *Text-to-Speech* merupakan teks yang diterjemahkan oleh perangkat menjadi suara sebagai keluaran dari perangkat sehingga suara dapat didengar oleh pengguna (Mustikarini dkk., 2019). Seiring berjalannya waktu penerapan *Speech Recognition* yang terkenal adalah Google Assistant oleh Google, Siri oleh Apple dan Alexa dari Amazon. Teknologi ini akan diimplementasikan pada *smartphone* Android yang dimiliki tunanetra untuk memandu rute pada gedung A Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Syiah Kuala (USK).

Android sendiri telah didukung fitur *mobile GPS, Geolocation, Bluetooth, Voice Recorder, Speaker*, dsb. Fitur tersebut dapat mendukung teknologi yang akan dibangun dalam penelitian ini. Pada penelitian ini juga menggunakan IPS dengan menggunakan Bluetooth Low Energy (BLE) dan Speech Command Recognition akan diimplementasikan pada aplikasi android untuk penentu lokasi dan pemandu rute. BLE memiliki kelebihan-kelebihan dibandingkan protokol *Internet of Things (IoT)* lainnya di antaranya konfigurasi yang mudah, metode pengiriman data yang mudah serta jangkauan koneksi yang luas (Puspitasari, 2020).

Pada proses implementasi, Beberapa ruangan pada Gedung A Fakultas Matematika dan Ilmu Pengetahuan Alam USK akan dipasang alat transmisi BLE. BLE bertugas untuk memancarkan gelombang radio untuk mengirimkan informasi secara berkala ke *smartphone* Android yang disebut *Beacon* (Lin dan Lin, 2018). Informasi yang diperoleh berupa kekuatan sinyal atau *signal strength*. Kemudian, *smartphone* pengguna akan menangkap indeks kekuatan sinyal atau *Received Signal Strength Indicator (RSSI)* yang dipancarkan dari masing-masing *Beacon* (Li dkk., 2018). Dengan memanfaatkan RSSI dari *Beacon* untuk menentukan lokasi dari pengguna serta sistem akan menggunakan masukkan berupa suara untuk menentukan ke mana tujuan pengguna dan keluaran berupa rute menuju tujuan pengguna yang dipandu menggunakan suara dari *smartphone*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka masalah yang akan dikaji pada penelitian ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi *Speech Command Recognition* untuk *Route Guidance* berbasis *Indoor Positioning* untuk tunanetra pada area gedung A FMIPA USK?
2. Bagaimana aplikasi dapat membantu serta memandu tunanetra agar sampai ke tujuan pada area gedung A FMIPA USK dengan menggunakan teknologi *Bluetooth Low Energy* (BLE) menggunakan metode *Received Signal Strength Indicator* (RSSI)?
3. Bagaimana proses pengimplementasian Model *Speech Command Recognition* ke dalam aplikasi *Route Guidance* berbasis *Indoor Positioning*?
4. Bagaimana menghitung akurasi penetapan rute pada aplikasi dengan mengambil posisi pertama sampai ke tujuan pada area gedung A FMIPA USK?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disebutkan sebelumnya, maka dapat dipaparkan tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun aplikasi *Speech Command Recognition* untuk *Route Guidance* berbasis *Indoor Positioning* untuk tunanetra pada area gedung A FMIPA USK.
2. Membangun aplikasi yang dapat memandu tunanetra agar sampai ke tujuan pada area gedung A FMIPA USK.
3. Mengimplementasi Model *Speech Command Recognition* ke dalam aplikasi *Route Guidance* berbasis *Indoor Positioning*.
4. Menghitung akurasi penetapan rute pada aplikasi dengan mengambil posisi pertama sampai ke tujuan pada area gedung A FMIPA USK.

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan kemudahan bagi pengguna terkhusus kepada tunanetra untuk menuju ke ruangan dipandu dengan navigasi suara pada area Gedung A FMIPA Universitas Syiah Kuala.

2. Memberikan aplikasi *Route Guidance* dengan teknologi *Indoor Positioning* serta *Speech Command Recognition* dengan keakuratan yang baik sehingga pengguna dapat sampai ke tujuan.

BAB II

TINJAUAN KEPUSTAKAAN

Untuk mendukung penelitian ini, maka dalam bab ini akan dikemukakan beberapa rumusan teori pendukung, yang dikutip dari berbagai referensi baik dalam bentuk buku, artikel, maupun tulisan karya ilmiah lainnya termasuk hasil penelitian sebelumnya yang ada kaitannya dengan penelitian yang dilakukan.

2.1. *Indoor Positioning System*

Indoor Positioning System (IPS, atau "*Indoor Location System*") merupakan teknologi yang menyediakan informasi bagi pengguna berdasarkan lokasi pengguna di dalam sebuah gedung (Brena dkk., 2017). IPS merupakan proses mendapatkan perangkat atau lokasi pengguna di dalam ruangan atau lingkungan (Zafari dkk., 2019). Banyak penelitian yang dilakukan dalam mengimplementasi IPS dengan berbagai pemanfaatan seperti WLAN. Namun pemanfaatan WLAN memiliki konsumsi daya yang boros pada smartphone sehingga pada akhirnya implementasi IPS diterapkan pada *Bluetooth low Energy* (BLE) yang memiliki konsumsi daya yang lebih rendah. Metode yang dapat dilakukan dalam pengimplementasian IPS adalah metode *trilateration* (berdasarkan jarak), *triangulation* (berdasarkan sudut) dan *fingerprinting* (Puspitasari, 2020).

Pada luar ruangan untuk mendeteksi suatu lokasi sudah diterapkan dengan adanya teknologi GPS, dimana GPS banyak memberikan dampak yang luar biasa pada kehidupan sehari-hari. Namun kegunaan GPS atau sistem lokasi berbasis satelit hanya terbatas pada lingkungan di luar ruangan. Dengan demikian dibutuhkan suatu metode dan teknologi khusus untuk sistem lokasi pada dalam ruangan yang disebut sebagai *Indoor Positioning System* (IPS) (Brena dkk., 2017). Dengan menggunakan sensor dan teknologi komunikasi sistem penentuan posisi pada IPS dapat menemukan objek di lingkungan dalam ruangan. Peluang pasar pada penggunaan IPS cukup besar sehingga dapat menarik minat ilmiah komunikasi untuk menerapkan teknologi tersebut (Brena dkk., 2017).

Pada teknologi IPS memiliki beberapa pendekatan yaitu, menentukan posisi dengan berbasis WiFi (WPS), menentukan posisi berbasis *Bluetooth Low Energy* (BLE), menentukan posisi berbasis Identifikasi Frekuensi Radio (RFID) dan menentukan posisi dengan menggunakan teknologi *Ultra-Wide Band* (UWB) atau *Visible Light Communication* (VLC) (Cantón Paterna dkk., 2017). IPS juga dapat dipadukan dengan kemampuan *route guidance* atau pemandu rute serta didampingi dengan sistem pemandu dengan suara.

2.2. Route Guidance System / Wayfinding and Navigation System

Menurut (Karimi, 2015), *Wayfinding and navigation system* memiliki 2 makna, yaitu merupakan perangkat khusus yang menyediakan solusi pencarian jalan dan navigasi tanpa memerlukan tautan komunikasi atau koneksi ke sistem lain dan tanpa penyedia pihak ketiga. Makna ke dua adalah referensi layanan melalui *smartphone* yang menyediakan solusi pencarian arah dan navigasi hanya melalui tautan komunikasi kabel-nirkabel ke sistem jarak jauh yang didukung oleh penyedia pihak ketiga. Dari perspektif komputasi, *Wayfinding and navigation system* adalah platform terpusat, terdistribusi, dan didistribusikan melalui klien (pada *smartphone*) dan server yang dikelola oleh penyedia layanan.

2.3. Bluetooth Low Energy (BLE)

Teknologi *Bluetooth* dikembangkan oleh Ericsson pada tahun 1994 dengan kegunaan sebagai standar komunikasi nirkabel untuk bertukar data dalam jarak dekat (Kaluža dkk., 2017). Teknologi *Bluetooth* memiliki fitur utama yaitu memiliki biaya yang rendah, konsumsi daya rendah, jangkauan kecil, memiliki ketahanan dan penggunaan secara global. Kecepatan transfer data yang diberikan oleh *Bluetooth* sebesar 1 Mbit/s dan menggunakan pita frekuensi tanpa izin 2,4 hingga 2,485 GHz (Kaluža dkk., 2017).

Pertengahan tahun 2010 spesial otoritas kompeten yang bernama "Bluetooth Special Interest Group" (SIG) mengumumkan spesifikasi *Bluetooth* 4.0, dimana meliputi : *Bluetooth* Klasik, *Bluetooth* dengan kecepatan tinggi, dan protokol *Bluetooth* hemat energi. Karakteristik dari *Bluetooth Low Energy* (BLE) adalah memiliki ukuran yang kecil, menggunakan biaya rendah, konsumsi penggunaan daya yang rendah dengan kemungkinan penggunaan sampai beberapa tahun bekerja dengan menggunakan baterai AAA dan memiliki kompatibilitas dengan perangkat seluler, tablet, dan komputer (Kaluža dkk., 2017).



Gambar 2.1. Perangkat Bluetooth Low Energy.

2.3.1. Beacon

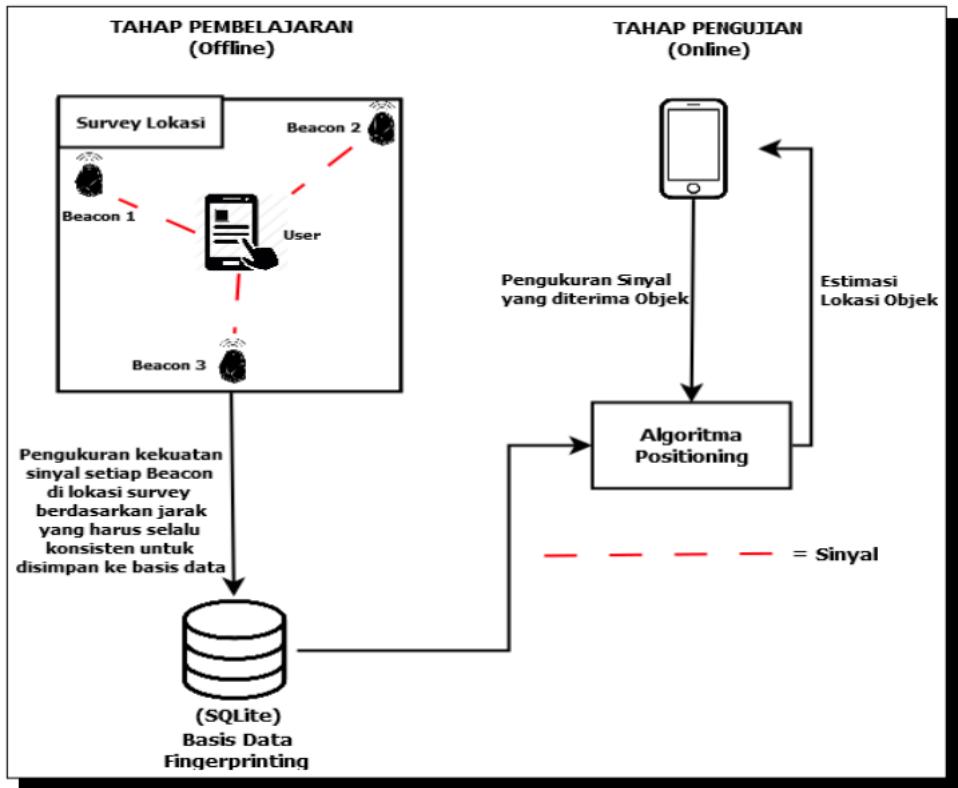
BLE memancarkan sinyal dari alat transmiter yang beroperasi menggunakan baterai. Alat transmiter tersebut disebut dengan *Beacon*. *Beacon* merupakan alat pendekripsi lokasi dengan harga yang terjangkau, ukurannya yang kecil, memiliki daya tahan baterai yang cukup lama, dan tidak membutuhkan energi listrik tambahan (Puspitasari, 2020). Setiap perangkat *smartphone* dan tablet yang mendekripsi sinyal dari *Beacon*, dapat menghitung jarak dan memperkirakan keberadaan lokasi setiap perangkat sekaligus. *Beacon Bluetooth* mengubah pengalaman menggunakan *smartphone* untuk bepergian, berbelanja, bekerja, dan bermain (Kaluža dkk., 2017).

2.4. Fingerprinting

Fingerprinting adalah teknik untuk menentukan lokasi dengan pemanfaatan *Radio Signal Strength* (RSS) dari suatu *Access Point* (AP). Menurut (Yudha dkk., 2018), metode ini memperhitungkan atenuasi karena kekuatan sinyal sering berubah-ubah. Setiap titik referensi dikumpulkan yang mengintegrasikan adanya penghalang antara transmiter dan *receiver*.

Transmiter untuk teknologi IPS yang ditunjukkan untuk WLAN disebut dengan AP, sedangkan transmiter untuk teknologi IPS yang ditunjukkan untuk *Bluetooth Low Energy* disebut dengan *Beacon* Puspitasari (2020). Metode fingerprinting berbasis IPS ini terdiri dari 2 tahap. Tahap pertama adalah tahap pembelajaran (*offline*), dimana lokasi fingerprints itu sendiri diperoleh dengan cara mengumpulkan RSSI dalam satuan desibel (dB) yang dipancarkan dari masing-masing AP. Kemudian gelombang radio yang dipancarkan AP yang diletakkan pada posisi yang telah ditentukan sebelumnya ditangkap oleh smartphone yang terintegrasi dengan WLAN ataupun *Bluetooth*.

Selama tahap pembelajaran lokasi yang tidak diketahui data pembelajarannya kemudian dirujuk sebagai titik referensi estimasi lokasi. Tahap kedua adalah tahap pengujian (*online*), dimana keakuratan yang dikumpulkan dalam data pembelajaran. Selama tahap pengujian sistem harus memberikan lokasi setiap objek berdasarkan data RSSI yang diamati. Namun, nilai RSSI bisa dipengaruhi oleh keadaan lingkungan sekitar yang dapat mengganggu keakuratannya (Subhan dkk., 2011). Menurut (Yudha dkk., 2018), metode Fingerprinting berbasis IPS ini melibatkan 2 tahap yang dapat dilihat pada Gambar 2.2.



Gambar 2.2. Ilustrasi Metode *Fingerprinting* (Yudha dkk., 2018)

2.4.1. *Received Signal Strength Indicator (RSSI)*

Received Signal Strength Indicator atau RSSI adalah metode pengukuran jarak yang memperoleh sinyal dari transmisi seperti *Bluetooth* untuk menentukan *distance-loss* model, dan kemudian memperkirakan posisi pengguna melalui beberapa algoritma (Li dkk., 2018). Menurut (Puspitasari, 2020) *Positioning System* menghasilkan data yang penting untuk menghitung lokasi pengguna. *Time of Arrival* (TOA), *Time Difference of Arrival* (TDOA), *Angle of Arrival* (AOA). RSSI memperkirakan jarak *node* yang belum diketahui ke referensi *node* dari beberapa kumpulan unit perhitungan dengan menggunakan atenuasi kekuatan sinyal (*signal strength*) yang dipancarkan oleh transmisi (Puspitasari, 2020).

Nilai RSSI didefinisikan dengan bilangan negatif. Semakin tinggi bilangan negativnya, maka kekuatan sinyal tersebut tergolong lemah. Namun apabila nilainya mendekati 0, maka sinyal tersebut tergolong kuat. RSSI dapat digolongkan menjadi 5 kategori kekuatan sinyal seperti Tabel 2.1

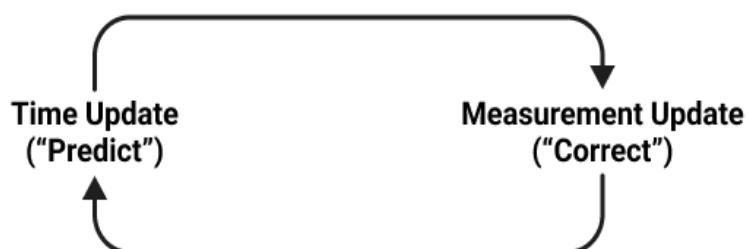
Tabel 2.1. Kekuatan sinyal RSSI (Sideeq dan Qasem, 2016)

No.	Signal Noise Ratio (SNR)	RSSI
1.	Kurang dari -40 dB	Luar Biasa
2.	-40 dB hingga -55 dB	Sangat Baik
3.	-55 dB hingga -70 dB	Baik
4.	-70 dB hingga -80 dB	Cukup Baik
5.	Lebih dari -80 dB	Buruk

2.4.2. Kalman Filter

Kalman Filter adalah seperangkat persamaan matematika yang menyediakan cara komputasi yang efisien untuk memperkirakan keadaan suatu proses, dengan cara meminimalkan rata-rata kesalahan kuadrat. Filter ini sangat berguna dalam beberapa aspek: mendukung estimasi-estimasi keadaan lampau, sekarang dan masa depan, dan dapat melakukannya bahkan ketika sifat pasti dari sistem yang dimodelkan tidak diketahui Welch (2020). Kalman Filter sendiri merupakan algoritma yang menggunakan serangkaian data yang diamati dari waktu ke waktu, yang mengandung *noise* dan ketidakakuratan lainnya, untuk memperkirakan variabel yang tidak diketahui dengan lebih akurat. Ini diusulkan oleh R. E. Kalman pada tahun 1960, dan menjadi pendekatan standar untuk estimasi yang optimal. Karena memiliki keunggulan waktu secara *realtime*, cepat, efisien, dan anti-interferensi yang kuat, Kalman filter telah banyak diterapkan di bidang perhitungan orbit, pelacakan target dan navigasi Li dkk. (2015).

Menurut (Ihsan dkk., 2018) berikut proses Kalman Filter dapat dilihat pada gambar dibawah ini:



Gambar 2.3. Ilustrasi Kalman Filter (Ihsan dkk., 2018)

Pada Kalman Filter dilakukan dengan dua tahapan yaitu tahap prediksi (*time update*) dan tahap pembaruan (*measurement update*) terhadap data-data pengukuran untuk memperbaiki hasil estimasi. Tahap prediksi adalah suatu tahapan untuk

mengubah suatu keadaan menjadi keadaan berikutnya dengan mengabaikan error yang ada (Ihsan dkk., 2018).

Berikut rumus menurut (Ihsan dkk., 2018):

- *Time Update (Predict)*

Predict State

$$x = x \quad (2.1)$$

Predict error covariance

$$p = p + q \quad (2.2)$$

- *Measurements update (Correct)*

Update the estimate via k

$$x = x + k * (\text{measurement}^\vee x) \quad (2.3)$$

Kalman gain

$$k = p / (p + r) \quad (2.4)$$

Update the error covariance

$$p = (1^\vee k) * p \quad (2.5)$$

Keterangan

x: Nilai yang di filter

p: Error estimasi

q: Noise yang diakibatkan dari proses

k: Kalman Gain

r: Noise dari sensor

Kalman Filter menerima inputan *gain* untuk meminimumkan kovariansi *error*. Kalman gain berupa inputan *high-gain* dan *low-gain*, dimana *high-gain* dapat mendekati posisi sebenarnya karena menghasilkan lintasan perkiraan lebih responsif sementara *low-gain* dapat menghaluskan *noise* namun mengurangi responsifnya. Oleh karena itu, semakin tinggi inputan nilai gain dari Kalman Filter maka semakin mendekati posisi sebenarnya (Ihsan dkk., 2018).

2.5. *Speech Recognition*

Speech Recognition merupakan proses yang dilakukan komputer untuk identifikasi suara yang diucapkan oleh seseorang tanpa memerlukan identitas orang terkait. *Speech Recognition* atau sering disebut juga *Automatic Speech Recognition* (ASR) biasa digunakan untuk mengubah ucapan menjadi teks, perintah suara untuk menjalankan perintah-perintah tertentu pada perangkat pengguna (Mustikarini dkk., 2019). Selain itu, ASR juga digunakan untuk Autentikasi biometri, yang dapat mengkonfirmasi pengguna dari suara mereka. ASR juga berguna untuk *general transcription*, contohnya untuk membuat *captions* secara otomatis untuk audio atau video (mentranskripsikan film atau video atau *live discussions*) (Mustikarini dkk., 2019).

2.5.1. *Text-to-Speech(TTS)*

Speech Synthesis atau *text-to-speech* merupakan kebalikan dari ASR dalam memetakan teks ke bentuk gelombang akustik. TTS adalah sistem yang mengonversi teks ke dalam bentuk ucapan. TTS dapat dipadukan dengan OCR (optical character recognition) atau komponen lainnya untuk menunjang suatu sistem yang menjalankan fungsi membaca untuk membantu pekerjaan manusia. TTS digunakan dalam agen percakapan yang berdialog dengan orang-orang, berperan dalam perangkat yang membacakan dengan keras untuk tunanetra atau dalam permainan, dan dapat digunakan untuk berbicara bagi penderita gangguan neurologis (Jonathan dan Suyanto, 2020)

2.6. *VOSK API*

Vosk api merupakan toolkit untuk speech recognition, dimana memiliki beberapa kelebihan, yaitu (Cephei, 2019):

1. Memiliki 19 lebih bahasa dan dialek yang didukung Vosk.
2. Vosk api merupakan toolkit speech recognition yang bisa digunakan secara offline, yang dapat digunakan pada Raspberry Pi, Android, iOS.
3. Kemudahan untuk menginstalasi vosk dengan menggunakan pip3 install vosk.
4. Model yang portabel pada masing-masing bahasa sebesar 50Mb, namun ada beberapa model server yang lebih besar pula.
5. Menyediakan streaming API untuk pengalaman pengguna terbaik.
6. Memiliki beberapa paket bahasa pemrograman yang berbeda-beda, seperti java, csharp, javascript dll.

7. Untuk akurasi terbaik dapat memungkinkan konfigurasi ulang kosakata dengan cepat.
8. Mendukung identifikasi pembicara selain dengan speech recognition yang sederhana.

2.7. Android

Android merupakan suatu *software* (perangkat lunak) yang digunakan pada *mobile device*(perangkat berjalan)yang meliputi sistem operasi, *middleware*, dan aplikasi inti. Android *Standart Development Kit* (SDK) menyediakan alat dan *Application Programming Interface*(API) yang diperlukan untuk memulai pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java, yaitu kode Java yang terkompilasi dengan data dan *file resources* yang dibutuhkan aplikasi dan digabungkan oleh *app tools* menjadi paket Android. *File* tersebut ditandai dengan ekstensi .apk. *File* inilah yang didistribusikan sebagai aplikasi dan dipasang pada perangkat *mobile* (Nasution dkk., 2018).

Menurut (Shaheen dkk., 2017). Ada 4 jenis komponen aplikasi. Setiap jenis memiliki tujuan yang berbeda dan memiliki siklus proses yang berbeda yang menentukan bagaimana komponen di *create* dan di *destroy*. Berikut adalah ke 4 jenis komponen tersebut:

1. *Activities*, merupakan sebuah *activity* merepresentasikan tampilan aplikasi kepada pengguna (*user interface*).
2. *Service*, merupakan komponen yang berjalan pada *background* untuk menjalankan operasi atau proses yang tidak memiliki *user interface*.
3. *Content Providers*, merupakan komponen yang menangani data antar aplikasi.
4. *Broadcast Receivers*, merupakan komponen yang bertanggung jawab atas menerima serta menyampaikan informasi atau notifikasi.

2.8. Android Studio

Android Studio merupakan *Integrated Development Environtment* (IDE) untuk pengembangan platform Android. Android Studio diumumkan pada tanggal 16 Mei 2013 di Google I/O Conference. Android Studio dapat digunakan secara gratis di bawah pengawasan Apache License 2.0. Android Studio merupakan kolaborasi antara JetBrains dan Google. Android Studio mirip dengan Eclipse yang disertai dengan ADT *Plugin* (Android Development Tools)(Craig dan Gerber, 2015).

Fitur-fitur Android Studio menurut (Puspitasari, 2020) adalah sebagian berikut:

- Proyek berbasis pada Gradle *Build*.
- *Refactory* dan perbaikan bug yang cepat.
- *Tools* baru yang bernama "Lint" diklaim dapat memonitor kecepatan, kegunaan, serta kompatibilitas aplikasi dengan cepat.
- Mendukung Proguard and *App-signing* untuk keamanan.
- Memiliki GUI aplikasi Android lebih mudah.
- Didukung oleh Google *Cloud Platform*, sehingga lebih mudah mengintegrasikan Google *Cloud Messaging and Application Engine* untuk setiap aplikasi yang dikembangkan.

2.9. SCRUM

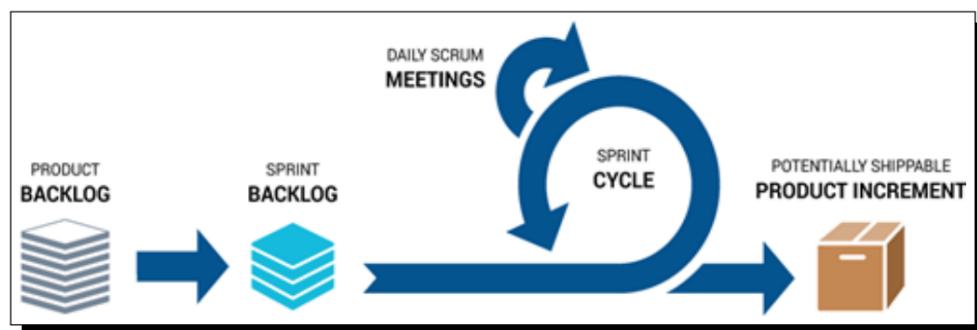
Scrum adalah *development framework* dimana tim lintas-fungsi mengembangkan produk atau proyek secara berulang dan bertahap. *Scrum* menyusun siklus pengembangan yang disebut *Sprint*. Iterasi ini masing-masing tidak lebih dari empat minggu dan berlangsung satu demi satu tanpa jeda (Deemer dkk., 2012). *Sprint* memiliki durasi tetap atau *Sprint* berakhir pada tanggal tertentu baik selesai atau belum, dan tidak pernah diperpanjang. Oleh karena itu *Sprint* dikatakan *timeboxed* (Schwaber dan Sutherland, 2011). Tahapan-tahapan metode *Scrum* menurut (Schwaber dan Sutherland, 2011) adalah sebagai berikut :

1. Dimulai dengan mengumpulkan *user requirements*, namun tidak harus semua *requirements* diharapkan harus keluar dari pemikiran pengguna di tahap awal proses pengembangan. Pengguna dapat mengubah pikiran mereka di setiap waktu selama proses pengembangan, pengguna dapat menambah fitur-fitur baru, menghapus atau memperbarui beberapa fitur yang telah ada sebelumnya.
2. Tahapan selanjutnya adalah memprioritaskan *requirements* dan *Product Backlog*. Sebuah perencanaan yang tepat dalam *Sprint* harus dilakukan sesuai dengan jumlah *Sprint* yang dibutuhkan untuk mengembangkan perangkat lunak, yang terdiri dari durasi *Sprint* tersebut dan *requirements* apa saja yang terdapat di *Product Backlog* yang harus diimplementasikan di setiap *Sprint* (dikenal dengan *Sprint Backlog*).
3. *Sprint* diawali dengan *Sprint Planning* dimana *Product Owner*, satu orang uang telah diberikan wewenang dan bertanggung jawab untuk memaksimalkan nilai produk di pasar, bertemu dengan tim *Scrum* (tim dengan jumlah 2-9

orang), kemudian bekerja sama untuk memperkirakan *requirements* dari *Product Backlog* apa saja yang dikerjakan selama satu *Sprint*.

4. *Sprint Planning* difasilitasi dengan *Scrum Master*. *Scrum Master* adalah seorang pemimpin yang melayani (*Servant Leader*). *Sprint Planning* memiliki batasan waktu selama 8 jam di dalam sebuah *Sprint* yang berdurasi selama 30 hari. Keluaran dari *Sprint Planning* adalah daftar pekerjaan dari hasil kesepakatan antara *Product Owner* dan tim *Scrum* dimana pekerjaan itu yang dikerjakan oleh tim *Scrum* nantinya selama satu *Sprint* beserta *Sprint Goal* yang dinamakan dengan *Sprint Backlog*.
5. Setelah *Sprint Planning* berakhir, tim *Scrum* akan mengambil *Sprint Backlog* untuk diri mereka masing-masing dan mengerjakan *Sprint Backlog* setiap hari hingga akhir *Sprint* tanpa campur tangan dari pihak mana pun. *Daily Scrum* akan dikerjakan oleh tim *Scrum* yang tidak lebih dari 15 menit untuk menentukan apa saja yang akan mereka kerjakan selama 24 jam ke depan berdasarkan perkembangan 24 jam terakhir, serta menyampaikan permasalahan yang menghambat mereka untuk bisa mencapai *Sprint Goal*. Tim *Scrum* akan melakukan perbaikan-perbaikan item dari *Product Backlog* pada *Sprint* yang akan datang selama proses pengembangan berlangsung, dengan tujuan membuat *Sprint Planning* menjadi lebih efektif.
6. Di akhir *Sprint* saat acara *Sprint Review*, *Product Owner* akan mempresentasikan hasil pekerjaan tim *Scrum* selama satu *Sprint* dan juga menjelaskan apa saja pencapaian tim *Scrum* menuju *Sprint Goal* di dalam *Sprint* tersebut kepada para pemegang kepentingan (*stakeholder*) agar mendapatkan *feedback*. *Feedback* ini akan dimasukkan ke dalam *Product Backlog* agar meningkatkan nilai dari sebuah produk. *Sprint Review* memiliki batasan waktu tidak lebih dari 4 jam untuk *sprint* yang memiliki durasi selama 30 hari.
7. Setelah *Sprint Review*, *Scrum Master* memfasilitasi acara yang bernama *Sprint Retrospectives* agar tim *Scrum*, *Product Owner* bekerja sama menentukan apa saja peningkatan yang akan mereka implementasikan di *Sprint* berikutnya. *Definition of Done* adalah salah satu hal yang ditekankan oleh tim *Scrum* pada saat *Sprint Retrospectives*.

8. *Sprint Retrospectives* merupakan acara yang paling penting dalam *Scrum* dikarenakan sifatnya yang menekankan *continuous learning* yang dapat meningkatkan tingkat *agility* perusahaan. *Sprint Review* memiliki batasan waktu tidak lebih dari 3 jam untuk *Sprint* yang memiliki durasi selama 30 hari. Setelah *Sprint Retrospectives* berakhir, maka *Sprint* berikutnya akan langsung dilakukan tanpa jeda antar *Sprint*. Pada setiap *Sprint*, *Product Owner* akan memastikan agar produk mencapai nilai setinggi mungkin saat pengembangan produk diakhiri. *Product Owner*, *Scrum Master* dan tim *Scrum* memegang komitmen, keberanian, saling menghargai satu sama lain, keterbukaan dan fokus. Ilustrasi tahapan-tahapan metode *Scrum* dapat dilihat pada 2.4 di bawah ini.



Gambar 2.4. Ilustrasi Metode Pengembangan Menggunakan Scrum (Schwaber dan Sutherland, 2011).

2.10. *Black Box Testing*

Salah satu jenis pengujian fungsional adalah *blackbox testing*. *Blackbox* testing merupakan suatu pengujian yang tidak menggunakan pengetahuan tentang struktur interior aplikasi. Saat melakukan pengujian, penguji akan berinteraksi langsung dengan *user interface*, lalu penguji akan memberikan masukan dan memeriksa hasil dari aplikasi yang digunakan tanpa mengetahui bagaimana proses dari hasil tersebut (Xu dkk., 2016). Pada *blackbox testing* mendapatkan pengujian dari deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain (Ammann dan Offutt, 2016).

Blackbox testing memiliki beberapa keuntungan, diantaranya penguji tidak perlu mengetahui suatu bahasa pemrograman, pengujian dilakukan dari sudut pandang pengguna agar dapat mengetahui suatu inkonsisten pada spesifikasi kebutuhan serta yang terakhir programmer dan penguji dapat saling bergantung (Jaya, 2018). *Blackbox testing* juga memiliki beberapa kekurangan yaitu suatu

pengujian sulit didesain tanpa spesifikasi yang jelas, memiliki kemungkinan untuk pengulangan *testing* dan tidak ada pengujian pada bagian *back-end* (Jaya, 2018).

2.11. *Usability Testing*

Usability Testing adalah kegiatan pengujian untuk mengumpulkan data mengenai sebuah produk dalam tahap pengembangan. Tujuan utama dari *usability testing* adalah untuk mengumpulkan data kuantitatif dengan mengukur waktu untuk mengidentifikasi dan memperbaiki kekurangan yang ada dalam produk serta bahan pendukung yang menyertainya sebelum produk dirilis (Barnum, 2020). Dengan *usability testing*, didapatkan apa yang sebenarnya dilakukan pengguna, apa yang berhasil untuk mereka, dan apa yang tidak dipikirkan akan mereka lakukan atau bahkan apa yang mereka pikir akan mereka lakukan jika menggunakan produk (Barnum, 2020).

Pengujian ini diharapkan akan mendapatkan kekuatan dan kelemahan dari setiap aspek yang ada pada aplikasi itu sendiri. Maka dari itu, perlu adanya dokumentasi pengalaman aktual para calon pengguna aplikasi atau produk saat dievaluasi (Wesfix, 2017). Tujuan lain dilakukannya pengujian ini adalah untuk mengumpulkan data kualitatif yang berhubungan dengan produk yang diuji. Data kualitatif tersebut terdiri dari komentar yang dibuat oleh partisipan, jawaban dari kuesioner pertanyaan dan tanggapan dari partisipan saat proses wawancara. *Usability testing* telah terbukti dapat mengurangi waktu pada tahap pengembangan, mengurangi jumlah bugs, dan menghasilkan produk yang lebih berkualitas untuk meningkatkan nilai jual (Wesfix, 2017).

2.11.1. *Usability Metric for User Experience (UMUX)*

UMUX merupakan salah satu instrumen kuesioner standar selain *System Usability Scale* (SUS). Meskipun SUS merupakan kuesioner yang relatif singkat, ada beberapa situasi dimana instrumen yang lebih pendek lebih disukai (misalnya, ketika ada kebutuhan untuk mengukur lebih banyak atribut daripada hanya kegunaan yang dirasakan yang mengarah ke sesuatu yang tetap serta terbatas untuk atribut apa pun). UMUX dirancang di Intel untuk mendapatkan pengukuran yang dirasakan konsisten dengan SUS, tetapi hanya menggunakan empat (bukan 10) item (Lewis, 2018).

Tabel 2.2. Daftar Pertanyaan Metode UMUX menurut (Lewis, 2018).

1	Kemampuan sistem ini memenuhi persyaratan saya.
2	Menggunakan sistem ini adalah pengalaman yang membuat frustrasi.
3	Sistem ini mudah digunakan.
4	Saya harus menghabiskan terlalu banyak waktu untuk memperbaiki hal-hal dengan sistem ini.

Menurut (Lewis, 2018) UMUX memiliki skala dari 1 (sangat tidak setuju) hingga 7 (sangat setuju) dengan skor item dimanipulasi untuk mendapatkan skor keseluruhan berkisar dari 0 hingga 100. Berikut tingkat yang dapat diterima sebagai kelayakan skor UMUX:

- Reliability (Koefisien alfa lebih besar dari 0.80).
- Concurrent validity (korelasi dengan SUS lebih besar dari 0.55, korelasi CSUQ sama dengan 0.65).
- Sensitivitas terhadap tingkat yang berbeda dari berbagai variabel independen (misalnya, membedakan antara sistem yang dinilai secara independen dari tingkat kegunaan yang relatif baik dan buruk, mendekripsi perbedaan dalam kegunaan yang dirasakan sebagai fungsi dari pengalaman).

Menurut (Finstad, 2010) berikut cara menghitung skor UMUX :

1. Item ganjil diberi skor [skor pengguna - 1]. Item genap diberi skor [7 - skor pengguna].
2. Jumlahkan perbedaan ini dan bagi jumlahnya dengan 24 (skor tertinggi).
3. Kalikan hasil dengan 100.
4. Cari nilai rata-rata dari seluruh pengguna.

2.12. Penelitian Terkait

Sampai saat ini terdapat banyak penelitian yang mengembangkan metode Indoor Positioning dengan menggunakan wireless devices. Selain mengembangkan metode, banyak juga yang menggunakan pendekatan dengan pengguna yang berbeda. Berikut adalah penelitian terkait dengan penelitian yang dilakukan.

1. Pada penelitian yang dilakukan oleh (Bai dkk., 2020), menggunakan BLE dengan RSSI, *Fingerprinting* untuk memantau pola hidup sehari-hari seperti orang tua, disabilitas, dan dementia. Hasil menunjukkan bahwa sistem mampu melacak lokasi pengguna dengan akurat.
2. Penelitian yang dilakukan oleh (Xiao dkk., 2018), menggunakan kamera untuk menangkap objek statis dengan *monocular photography* seperti pintu dan jendela untuk menentukan lokasi pengguna dengan menggunakan *deep learning* dan *computer vision*. Pengujian dilakukan di tempat umum dengan bidang pandang rumit seperti museum.
3. Penelitian yang dilakukan oleh (Yadav dkk., 2019), menggunakan BLE dan Kalman Filter dengan algoritma Trusted K neares Bayesian Estimation (TKBE) untuk memantau pejalan kaki dengan menduga pergekannya.
4. Penelitian yang dilakukan oleh (Qin dkk., 2021) menggunakan WiFi dengan RSSI untuk *Indoor* dengan menggunakan algoritma CCPoS dan dibangun dengan *acoustic modeling* TDNNF.

BAB III

METODOLOGI PENELITIAN

3.1. Waktu dan Lokasi Penelitian

Penelitian ini akan bertempat pada Gedung A Fakultas Matematika dan Ilmu Pengetahuan Alam. Waktu yang dibutuhkan agar penelitian ini dapat diimplementasikan adalah 8 bulan terhitung dari Oktober 2022 hingga Juni 2023.

3.2. Alat dan Bahan

Alat dan Bahan yang akan digunakan pada penelitian ini terdiri dari beberapa perangkat keras (*Hardware*) dan perangkat lunak (*Software*) yang dijabarkan sebagai berikut:

1. Perangkat Keras

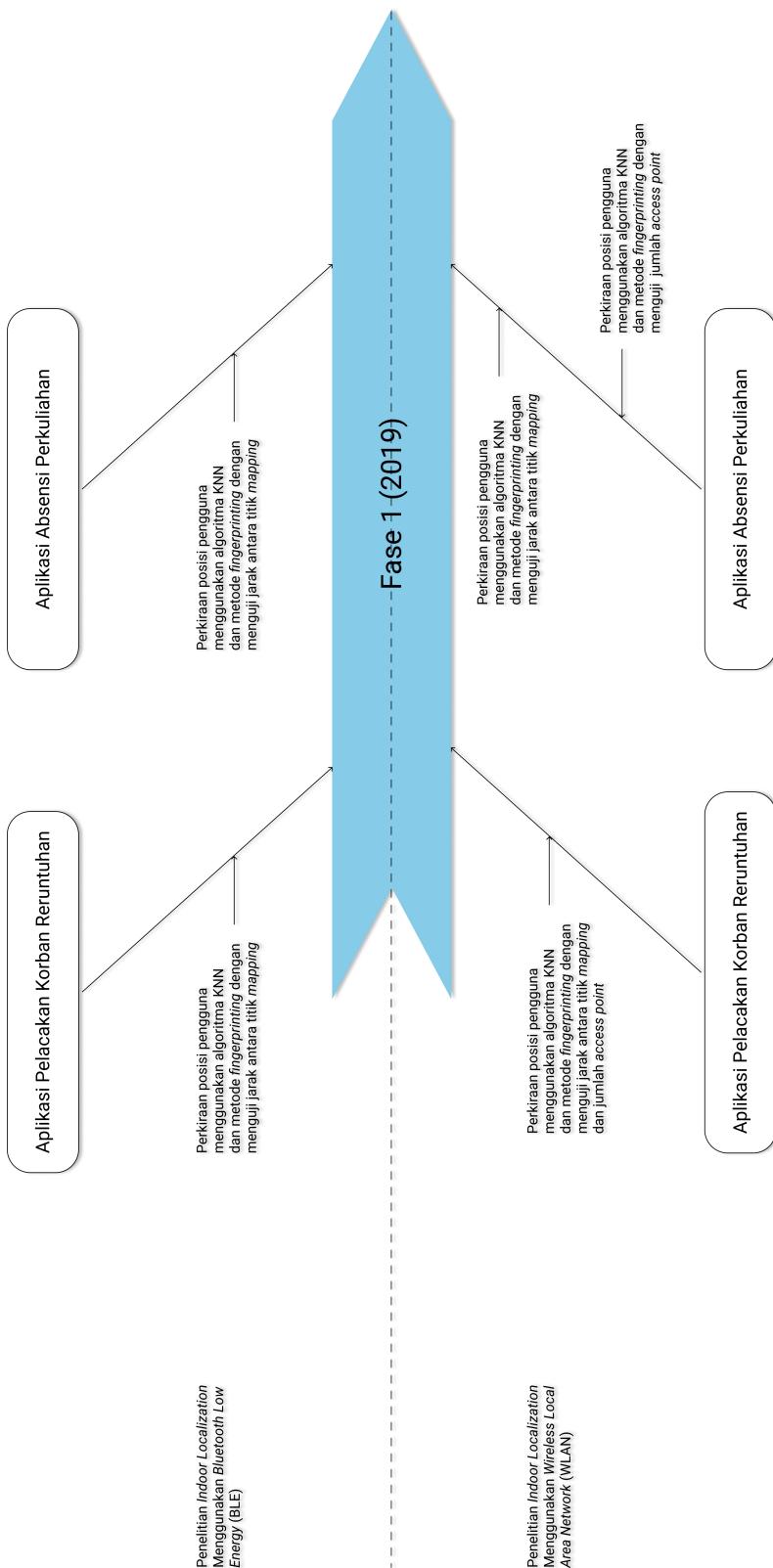
- Laptop Acer Aspire E5-475g dengan RAM 16GB, Intel Core i5-7200U 2.5GHz, Nvidia GeForce 940MX 2GB, *Harddisk (HDD)* 1500Gb, *Solid State Drive (SSD)* 250GB.
- *Smartphone* Xiaomi Poco F3 dengan RAM 6GB, *Internal Storage* 128GB.
- *Personal Computer* dengan AMD Ryzen 7 2700x, RAM 16GB, Nvidia GeForce RTX 2080 8GB, *Solid State Drive (SSD)* 500GB.
- *Beacon Bluetooth.*

2. Perangkat Lunak

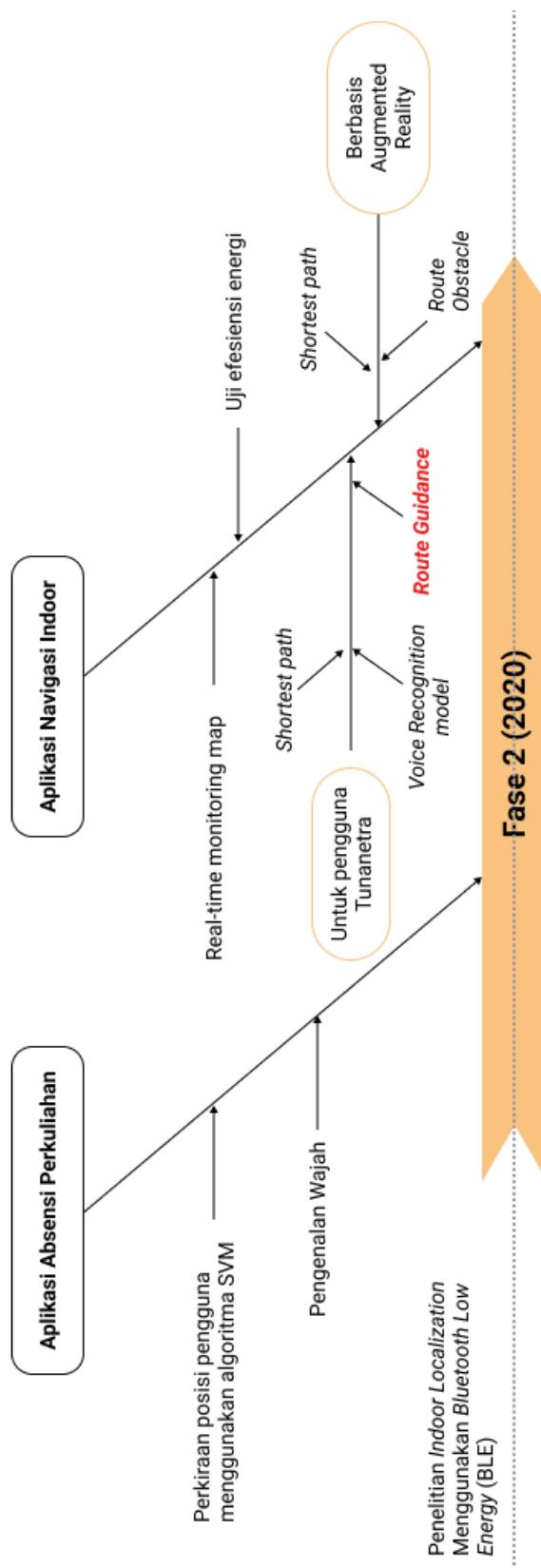
- Windows 11 Pro
- Linux Ubuntu 20.04.1 LTS
- Android Studio 2021.2.1.15
- Visual Studio Code
- Figma
- Notion
- Vosk API

3.3. *Roadmap* Penelitian

Roadmap penelitian merupakan diagram yang menggambarkan rangkaian beberapa penelitian yang saling berkesinambungan dalam rentang waktu tertentu. *Roadmap* penelitian biasa dibuat untuk memberikan batasan kepada peneliti agar menghindari pengamatan yang tidak perlu dan fokus terhadap bagian penelitiannya saja. Pada penelitian ini dibagi ke dalam 2 fase. Fase pertama pada tahun 2019 memiliki fokus penelitian pada *indoor localization* dengan menggunakan *Bluetooth Low Energy* (BLE) dan menggunakan *Wireless Local Area Network* (WLAN) dapat dilihat gambar 3.1. Pada fase kedua pada tahun 2020 lebih berfokus pada penelitian *Indoor Localization* dengan menggunakan BLE. Penelitian ini terletak pada fase 2 di tahun 2020 dengan topik utama yaitu Aplikasi Navigasi *Indoor* dengan sub topik untuk pengguna Tunanetra. Penelitian ini memiliki batasan berupa pembangunan aplikasi *mobile* untuk *Route Guidance* untuk Tunanetra, seperti yang ditunjukkan pada gambar 3.2 dengan kata yang dicetak tebal berwarna merah sebagai berikut.



Gambar 3.1. Roadmap Penelitian Fase 1

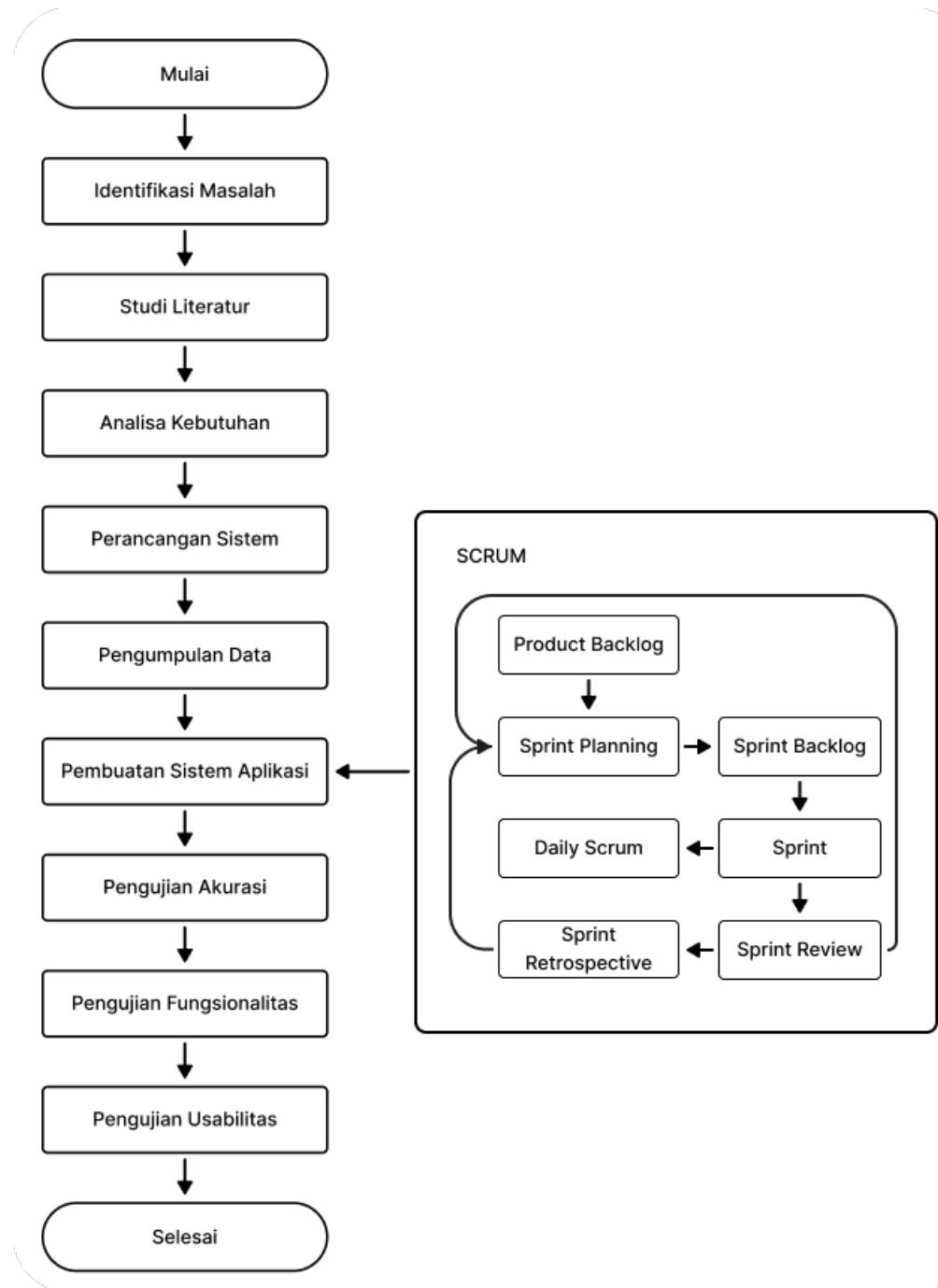


Gambar 3.2. Roadmap Penelitian Fase 2

Penelitian Indoor Localization
Menggunakan Wireless Local
Area Network (WLAN)

3.4. Metode Penelitian

Metode penelitian yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3.3.



Gambar 3.3. Diagram Alir Penelitian

3.4.1. Identifikasi Masalah

Tahapan ini merupakan tahapan yang dilakukan untuk mengidentifikasi masalah pada lingkungan yang berhubungan dengan aplikasi yang akan dibuat baik secara langsung maupun tidak langsung dan menjadi landasan mengapa aplikasi ini harus dibuat. Masalah-masalah yang berhasil di identifikasi adalah sebagai berikut:

- Belum ada teknologi *Route Guidance System/Wayfinding System* berbasis *Indoor positioning* dengan panduan rute menggunakan *Speech Command Recognition* di Universitas Syiah Kuala.
- Membantu Tunanentra menemukan ruangan di Gedung A FMIPA Universitas Syiah Kuala dipandu dengan navigasi suara.

3.4.2. Studi Literatur

Studi literatur digunakan sebagai bahan referensi selama proses penelitian. Studi literatur dilakukan dengan cara mencari situs *website* dan jurnal-jurnal terkait tentang penelitian, baik jurnal nasional maupun internasional, buku-buku yang telah diterbitkan, serta situs-situs internet yang berkaitan dengan permasalahan yang dikaji dalam penelitian. Studi literatur dapat dikembangkan untuk menyempurnakan kekurangan dari penelitian sebelumnya.

3.4.3. Analisa Kebutuhan

Pada tahapan ini dilakukan proses analisa kebutuhan yang bersumber dari masalah yang di identifikasi sebelumnya. Proses pembangunan aplikasi akan didasarkan pada kebutuhan yang ada. Berikut adalah hasil analisa dari beberapa kebutuhan dari sistem yang akan dibangun.

1. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fungsionalitas sistem. Kebutuhan fungsional dari identifikasi masalah yang telah dilakukan adalah sebagai berikut:

- Melakukan proses pemanduan kepada pengguna secara *background process* dengan bantuan *speech command recognition* dan *route guidance* dengan aplikasi berbasis Android.
- Menampilkan prediksi lokasi pengguna serta rute terbaik menuju ke lokasi tujuan pengguna.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional memastikan batasan eksternal yang harus dipenuhi oleh sistem. Batasan-batasan tersebut antara lain:

- Proses pemanggilan aplikasi dimulai dengan memanggil *hotword*.
- Proses pencatatan dan penentuan lokasi dilakukan secara berkala dalam interval yang sesingkat-singkatnya.
- Hanya dapat melakukan proses pemanduan rute apabila *Bluetooth* pada perangkat hidup dan terhubung dengan *Beacon*.
- Sistem hanya dapat mendeteksi lokasi pengguna di dalam gedung yang telah di petakan terlebih dahulu.

3.4.4. Perancangan Sistem

Tahap perancangan sistem ini meliputi perancangan alur kerja sistem yang berfungsi untuk memastikan sistem yang dibangun dapat digunakan secara baik oleh pengguna. Tahap perancangan sistem ini terbagi menjadi beberapa bagian yaitu diagram Use Case , Diagram Deployment, alur kerja sistem, *Flow Diagram* dan desain prototipe berdasarkan analisa kebutuhan yang telah dijabarkan.

3.4.5. Pengumpulan Data

1. Data Model *Voice Recognition*

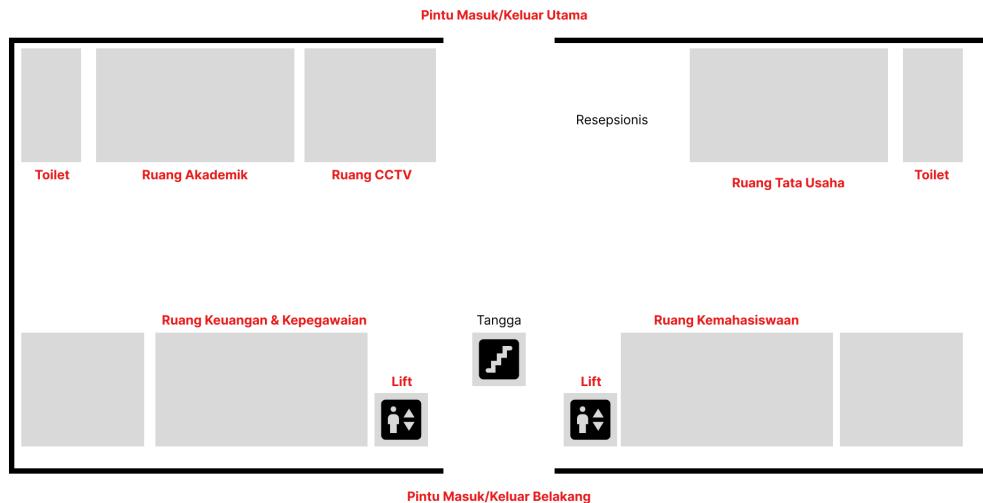
Data yang digunakan dalam penelitian ini terdiri atas data *pre-trained* model yang diperoleh dari penelitian yang berada dalam satu penelitian yang sama pada *roadmap* pada gambar 3.2. *Pre-trained* model dimanfaatkan sebagai titik awal untuk merespons perintah suara dari pengguna, kemudian perintah suara akan di terjemahkan ke dalam bentuk teks yang akan diterima oleh *smartphone* pengguna. Teks yang diterima akan digunakan sebagai input yang akan menjadi lokasi tujuan serta pemilihan rute.

2. Data Rute dan Denah Lokasi Penelitian

Rute yang akan digunakan berlokasi di gedung A FMIPA lantai 1 sampai dengan lantai 3. Data rute yang digunakan akan di petakan dengan menggunakan bantuan BLE. Data titik-titik BLE disimpan dalam *database* lokal berupa MAC Address, nilai RSSI, dan nama perangkat BLE. Dengan adanya RSSI atau kekuatan sinyal yang disimpan perangkat smartphone pengguna akan menangkap sinyal dan menyesuaikan dengan titik-titik yang telah dipetakan sehingga mendapatkan lokasi pengguna berada serta

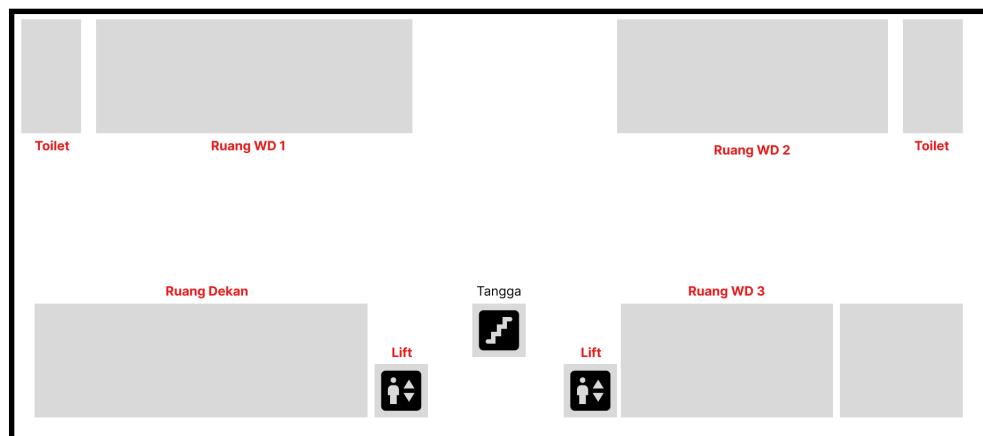
menjadikan titik lokasi pengguna sebagai pemilihan rute terdekat ke lokasi tujuan pengguna. Denah lokasi penelitian dapat dilihat pada gambar berikut.

Lantai 1



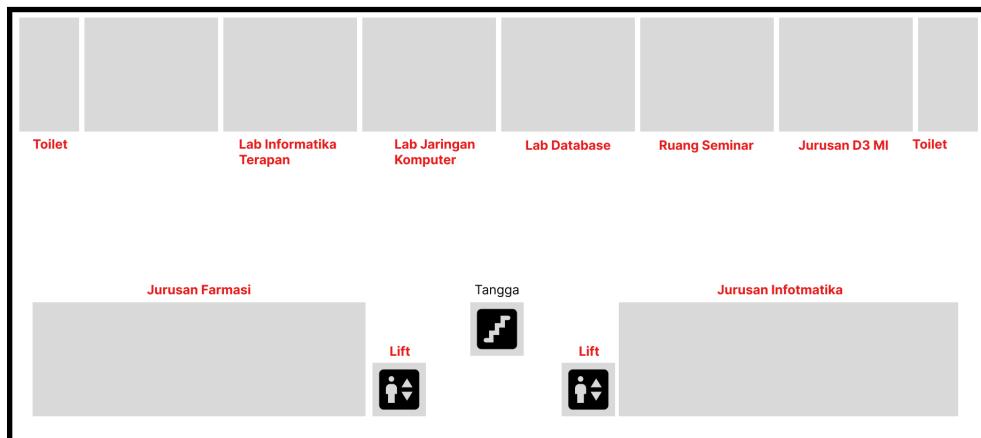
Gambar 3.4. Denah lantai 1 Gedung A FMIPA

Lantai 2



Gambar 3.5. Denah lantai 2 Gedung A FMIPA

Lantai 3



Gambar 3.6. Denah lantai 3 Gedung A FMIPA

3. Data *Text-to-Speech*

Data yang akan digunakan menggunakan *text-to-speech* yang disediakan oleh library Google speech. Sehingga proses pemanduan menuju ruangan akan dipandu dengan suara seperti "Menuju Ruangan Jurusan Informatika", "Belok ke kanan dalam 5 langkah", "Telah sampai di tujuan , Ruangan Jurusan Informatika".

4. Data akurasi lokasi pengguna serta akurasi rute

Data yang akan diperoleh dari proses koneksi BLE terdekat dengan RSSI sebagai tolak ukur akurasi lokasi pengguna, sedangkan untuk akurasi rute menggunakan data pemetaan rute dan lokasi tujuan yang telah dipetakan pada aplikasi dengan memanfaatkan RSSI dari beberapa koneksi BLE terdekat yang saling terhubung menggunakan Kalman Filter.

3.4.6. Pembuatan Sistem Aplikasi

Pada Proses pembuatan sistem aplikasi, metode pengembangan aplikasi yang digunakan adalah metode *Scrum*, dikarenakan sistem ini dikembangkan bersama dengan tim dan membutuhkan fleksibilitas terhadap perubahan yang terjadi dalam proses pengembangannya, sehingga memerlukan iterasi secara berkala agar berjalan dengan baik. Salah satu tahap yang merupakan paling awal dari metode ini adalah *product backlog*. Pada tahap ini, semua kebutuhan akan ditransformasikan menjadi rancangan pekerjaan untuk diselesaikan agar aplikasi dapat bekerja secara fungsional. Adapun *product backlog* dari keseluruhan aplikasi ini adalah sebagai berikut.

- Menampilkan halaman beranda.
- Menampilkan *Pop up* yang muncul dari *input* suara.
- Meluncurkan aplikasi dengan proses pengenalan suara.
- Menampilkan dan mengeluarkan suara daftar ruangan yang bisa di gunakan sebagai tujuan.
- Menampilkan rute terdekat ke ruangan yang di tuju oleh pengguna.
- Mengeluarkan suara navigasi ke ruangan yang di tuju.

3.4.7. Pengujian Akurasi

Pengujian akurasi posisi pengguna dan pemilihan rute akan menggunakan gabungan titik BLE sebagai titik awal dengan mengukur kekuatan RSSI, kemudian menggunakan letak dan posisi BLE yang telah dipetakan sebelumnya untuk melakukan pemilihan rute serta proses navigasi yang akan diprediksi menggunakan algoritma Kalman Filter disertai dengan fase pembaharuan prediksi titik lokasi pengguna ke tujuan.

Berikut perhitungan yang akan digunakan untuk menguji akurasi pengguna serta rute pengguna, rumus menurut (Ihsan dkk., 2018):

- *Time Update (Predict)*

Predict State

$$x = x \quad (3.1)$$

Predict error covariance

$$p = p + q \quad (3.2)$$

- *Measurements update (Correct)*

Update the estimate via k

$$x = x + k * (\text{measurement}^\circ x) \quad (3.3)$$

Kalman gain

$$k = p / (p + r) \quad (3.4)$$

Update the error covariance

$$p = (1^\circ k) * p \quad (3.5)$$

Keterangan

x: Nilai yang di filter

p: Error estimasi

q: Noise yang diakibatkan dari proses

k: Kalman Gain

r: Noise dari sensor

3.4.8. Pengujian Fungsionalitas dengan Metode *Black Box*

Metode yang digunakan untuk melakukan pengujian fungsionalitas adalah dengan *black box testing*. Pengujian ini dilakukan untuk memeriksa fungsi-fungsi pada sistem yang dibangun pada penelitian ini apakah sudah bekerja sesuai dengan kebutuhan pengguna ataupun tidak. Pengujian ini dilakukan dengan cara menjalankan setiap fungsi yang ada pada sistem dan memastikan agar sistem dapat bekerja dengan semestinya sesuai alur yang telah dirancang. Pengujian ini juga dapat melibatkan target pengguna dan pengembang.

3.4.9. Usability Testing

Usability Testing merupakan pengujian aplikasi untuk melihat apakah pengguna dapat dengan mudah dan nyaman dalam menggunakan aplikasi tersebut serta melihat dan mengevaluasi keberhasilan dari sebuah produk atau jasa. Pengujian ini akan menggunakan metode UMUX. UMUX terdiri dari 4 pertanyaan dengan menggunakan skala 1-7. Berikut daftar pertanyaan-pertanyaan dengan metode UMUX dapat dilihat pada Tabel 3.1.

Tabel 3.1. Daftar Pertanyaan Metode UMUX.

1	Kemampuan sistem ini memenuhi persyaratan saya.
2	Menggunakan sistem ini adalah pengalaman yang membuat frustrasi.
3	Sistem ini mudah digunakan.
4	Saya harus menghabiskan terlalu banyak waktu untuk memperbaiki hal-hal dengan sistem ini.

Untuk menghitung skor akhir metode UMUX (Finstad, 2010), dapat dilihat dari cara berikut.

1. Item ganjil diberi skor [skor pengguna - 1]. Item genap diberi skor [7 - skor pengguna].
2. Jumlahkan perbedaan ini dan bagi jumlahnya dengan 24 (skor tertinggi).
3. Kalikan hasil dengan 100.
4. Cari nilai rata-rata dari seluruh pengguna.

Tingkat nilai skala pengujian menentukan apakah sistem tersebut layak digunakan, bermanfaat, diterima oleh pengguna dan bertahan lama penggunaannya. Sebuah sistem dengan nilai pengujian yang tinggi membuat sistem tersebut menjadi populer dalam waktu yang lama dan penggunaannya yang luas, dikarenakan banyak individu akan merasakan manfaat dari kehadiran sistem tersebut. Sedangkan sistem dengan nilai pengujian yang rendah, sering kali diabaikan walaupun dibuat berdasarkan kebutuhan dan menghasilkan sumber daya yang banyak.

BAB IV

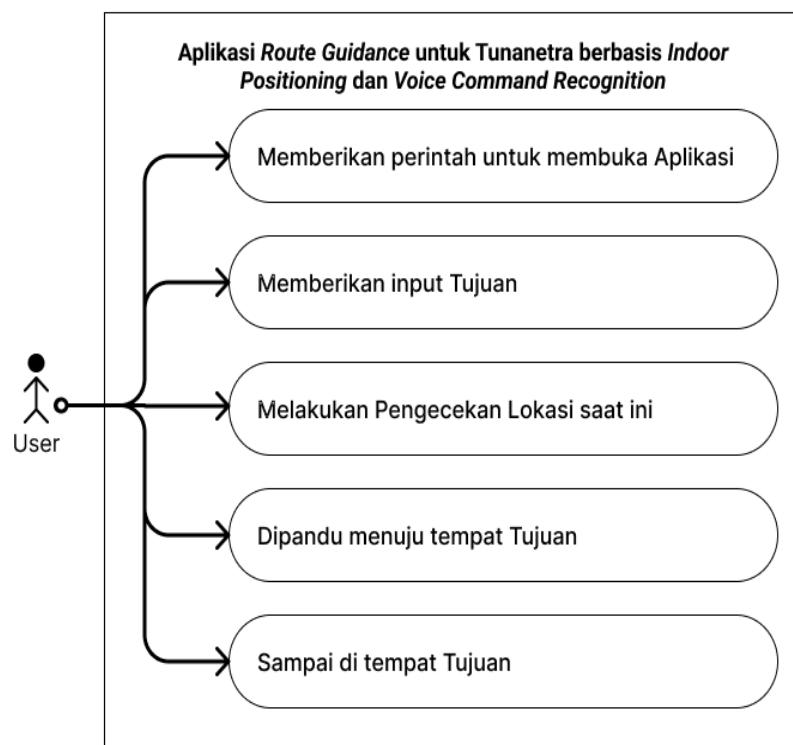
HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem

Tahap perancangan sistem ini meliputi perancangan alur kerja sistem yang berfungsi untuk memastikan sistem yang dibangun dapat digunakan secara baik oleh pengguna. Tahap perancangan sistem ini terbagi menjadi beberapa bagian yaitu diagram Use Case , Diagram Deployment, alur kerja sistem, *Flow Diagram* dan desain prototipe berdasarkan analisa kebutuhan yang telah dijabarkan. Berikut ini merupakan tahapan tersebut:

1. *Use Case Diagram*

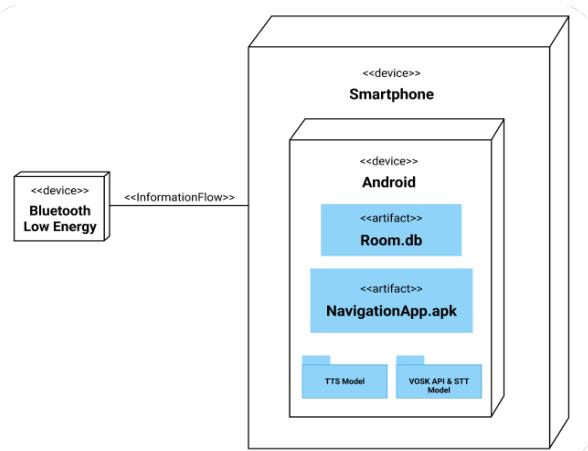
Use Case Diagram merupakan gambaran interaksi antara pengguna dan sistem. Pengguna dari sistem yang akan dibangun ini adalah Tunanetra dan pengunjung gedung FMIPA USK. *Use Case Diagram* dari sistem ini dapat dilihat pada Gambar 4.1



Gambar 4.1. *Use Case Diagram*

2. Deployment Diagram

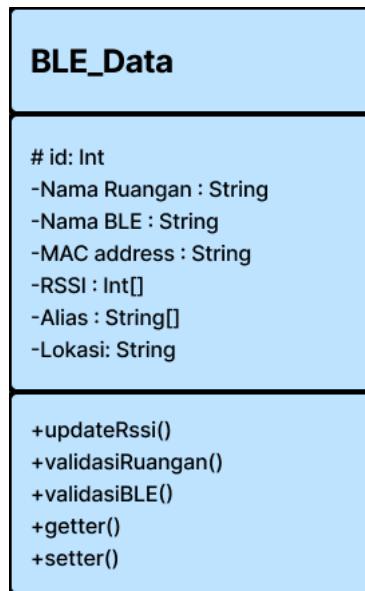
Deployment Diagram merupakan gambaran hubungan antara perangkat lunak dan perangkat keras yang digunakan dalam sebuah sistem. Seperti yang telah disebutkan sebelumnya, penelitian ini merupakan bagian dari penelitian lain yang saling terintegrasi sehingga tidak semua *node* dalam diagram menjadi fokus dalam penelitian ini. Bagian dalam diagram yang menjadi jangkauan dari penelitian ini ditandai dengan warna biru seperti yang ditunjukkan pada Gambar 4.2



Gambar 4.2. Deployment Diagram

3. *Class Diagram*

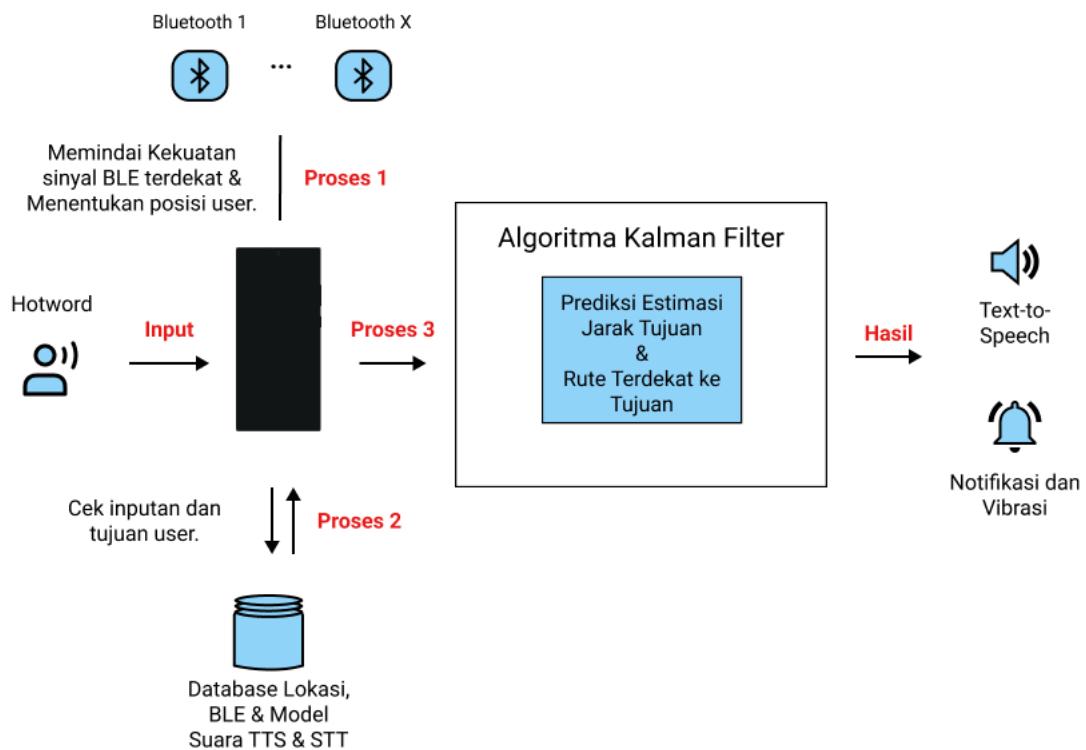
Class Diagram merupakan gambaran struktur sistem untuk memodelkan sistem mulai dari kelas sistem, atribut, metode dan hubungan antar objek. fungsi utama *class diagram* adalah untuk memvisualisasikan, menentukan, dan mendokumentasikan fitur struktural dari pemodelan sistem, dalam penelitian untuk BLE Data yang akan disimpan secara lokal dalam *database*. Berikut adalah *class diagram* untuk penelitian ini.



Gambar 4.3. *Class Diagram*

4. Alur Kerja Sistem

Alur kerja sistem merupakan langkah-langkah yang dilalui sistem hingga fungsionalitas sistem dapat dimanfaatkan oleh pengguna. Alur kerja sistem ini dapat dilihat pada Gambar 4.4

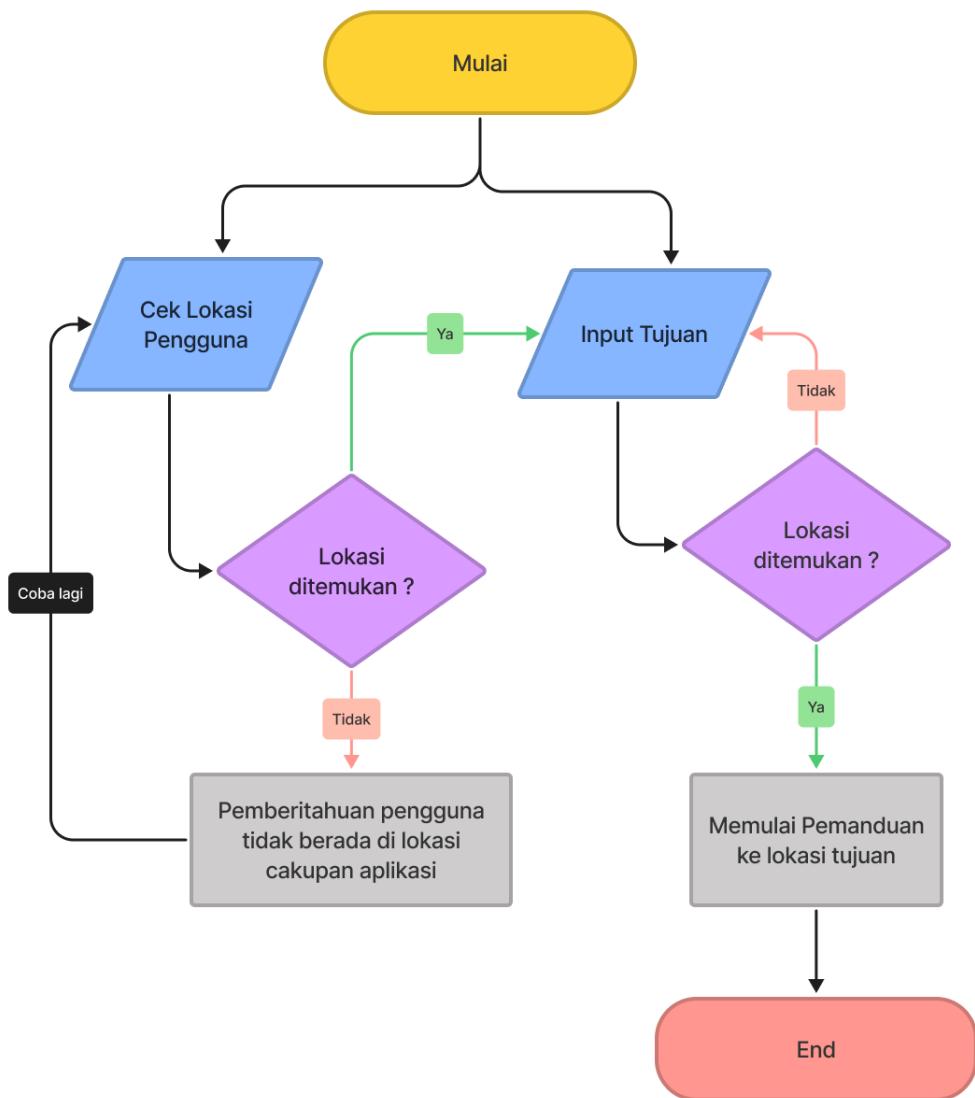


Gambar 4.4. Alur Kerja Sistem

- *Input* diterima dari pengguna setelah mengucapkan *hotword*.
- Pada proses 1 aplikasi dan perangkat akan memindai kekuatan sinyal BLE terdekat serta menentukan posisi pengguna.
- Pada proses 2, setelah posisi pengguna ditentukan, aplikasi akan melakukan pengecekan *input* dari pengguna pada *database* lokasi menggunakan model yang telah tersedia oleh penelitian lainnya.
- Pada proses 3, aplikasi akan memprediksi jarak tujuan serta rute terdekat menuju tujuan pengguna menggunakan algoritma Kalman Filter.
- Hasil yang dihasilkan dari proses-proses sebelumnya berupa notifikasi dan vibrasi serta *text-to-speech* atau berupa ucapan dari rute yang akan di tempuh oleh pengguna, seperti "Belok ke kanan dalam 5 langkah", "Telah sampai di tujuan , Ruangan Jurusan Informatika", dsb.

5. Flow Diagram

Flow Diagram menunjukkan bagaimana proses tahapan melakukan navigasi pada sistem seperti pada gambar 4.5 berikut ini.



Gambar 4.5. *Flow Diagram Navigasi Indoor*

6. Desain Prototipe

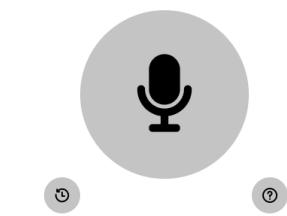
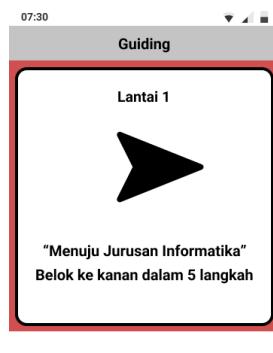
Desain prototipe menunjukkan perkiraan tampilan aplikasi yang akan dibuat. Desain ini dirancang berdasarkan beberapa skenario pada sisi pengguna baik tunanetra dan bukan tunanetra.



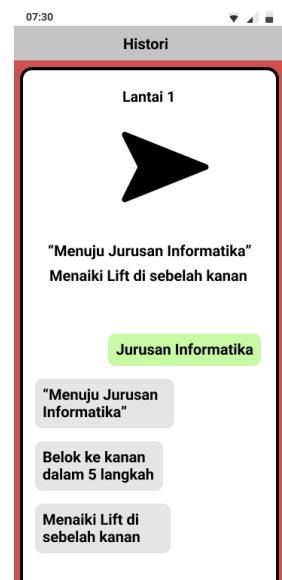
(a) Pop up trigger by hotword



(b) Input tujuan dan lokasi pengguna



(c) Proses Navigasi ke Tujuan



(d) Histori Navigasi

Gambar 4.6. Tampilan Halaman Aplikasi Navigasi Indoor untuk Tunanetra

4.2. Pembuatan Sistem

4.2.1. Persiapan data

1. Data Model *Voice Recognition*

Data yang digunakan dalam penelitian ini terdiri atas data *pre-trained* model yang diperoleh dari penelitian yang berada dalam satu penelitian yang sama pada *roadmap* pada gambar 3.2. *Pre-trained* model dimanfaatkan sebagai titik awal untuk merespons perintah suara dari pengguna, kemudian perintah suara akan dijemahkan ke dalam bentuk teks yang akan diterima oleh *smartphone* pengguna. Teks yang diterima akan digunakan sebagai input yang akan menjadi lokasi tujuan serta pemilihan rute. Berikut adalah *list* dari nama-nama ruangan yang ada pada data *Pre-trained* model.

Tabel 4.1. Data Ruangan & Perintah

No.	Nama Ruangan	Keterangan
1	ruang cctv	Lantai 1
2.	ruang kepegawaian	
3.	ruang tata usaha	
4.	ruang tu	
5.	ruang kemahasiswaan	
6.	ruang akademik	
7.	ruang dekan	Lantai 2
8.	ruang wakil dekan satu	
9.	ruang wd satu	
10.	ruang wakil dekan dua	
11.	ruang wd dua	
12.	ruang wakil dekan tiga	
13.	ruang wd tiga	Lantai 3
14.	jurusin informatika	
15.	jurusin d tiga mi	
16.	jurusin farmasi	
17.	ruang seminar	
18.	laboratorium jaringan komputer	
19.	lab jaringan komputer	
20.	laboratorium jaringan	
21.	lab jaringan	
22.	laboratorium jarkom	
23.	lab jarkom	

Table 4.1 continued from previous page

24.	laboratorium basis data	
25.	lab basis data	
26.	laboratorium data mining	
27.	lab data mining	
28.	laboratorium multimedia	
29.	lab multimedia	
30.	laboratorium gis	
31.	lab gis	
32.	mushala	
33.	surau	
34.	tempat shalat	
35.	toilet wanita	
36.	toilet pria	
37.	toilet perempuan	
38.	toilet laki laki	
39.	toilet cewek	
40.	toilet cowok	
41.	kamar mandi perempuan	Ruang Umum
42.	kamar mandi wanita	
43.	kamar mandi cewek	
44.	kamar mandi pria	
45.	kamar mandi laki laki	
46.	kamar mandi cowook	
47.	wc perempuan	
48.	wc wanita	
49.	wc cewek	
50.	wc pria	
51.	wc laki laki	
52.	wc cowok	
53.	pintu masuk	
54.	pintu utama	
55.	pintu masuk utama	
56.	pintu keluar	

2. Data Rute dan Denah Lokasi Penelitian

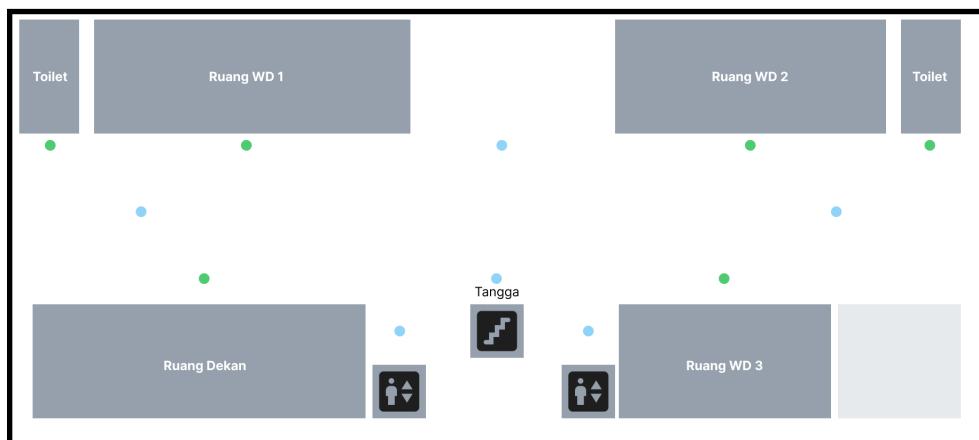
Rute yang akan digunakan berlokasi di gedung A FMIPA lantai 1 sampai dengan lantai 3. Data rute yang digunakan akan dipetakan dengan menggunakan bantuan BLE. Data titik-titik BLE disimpan dalam *database* lokal berupa MAC Address, nilai RSSI, dan nama perangkat BLE. BLE berjumlah 40 unit dengan BLE untuk ruangan berjumlah 25 ditandai dengan titik berwarna hijau dan BLE untuk *connector* berjumlah 15 ditandai dengan titik berwarna biru. Jarak masing-masing BLE berada di sekitar 3 s/d 8 meter dari masing-masing BLE mengikuti jarak antara ruangan. Denah lokasi penelitian dapat dilihat pada gambar berikut.



Gambar 4.7. Denah lantai 1 Gedung A FMIPA dengan BLE

Lantai 2

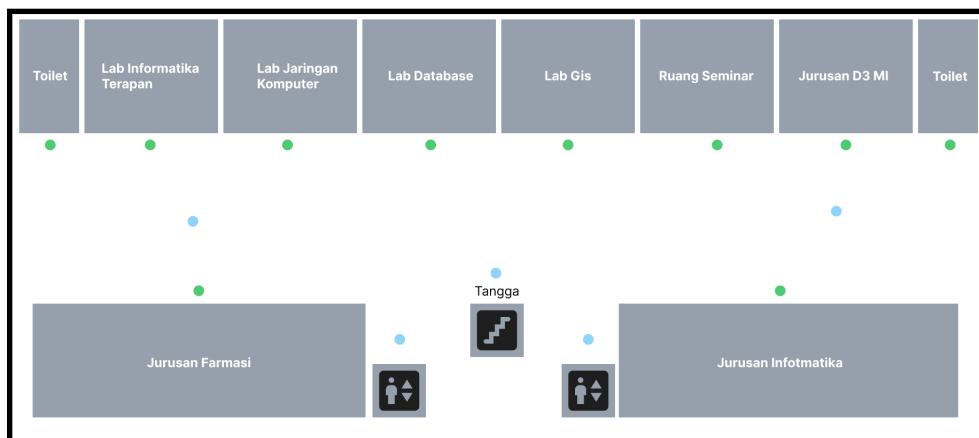
12 BLE Connector = 6
Ruangan = 6



Gambar 4.8. Denah lantai 2 Gedung A FMIPA dengan BLE

Lantai 3

14 BLE Connector = 5
Ruangan = 10



Gambar 4.9. Denah lantai 3 Gedung A FMIPA dengan BLE

4.2.2. Pembuatan Aplikasi

1. Konfigurasi Model *Voice Recognition* dan *Text-to-speech*

Pada tahapan ini model dipasangkan ke dalam aplikasi dengan menggunakan VOSK API dan Google *text-to-speech* dengan menggunakan fungsi Talkback pada Android sehingga aplikasi dapat menerima input dari pengguna serta memberikan informasi *feedback* ke pengguna saat dipandu.

Implementasi model dipanggil ke dalam bagian build.gralde diikuti dengan implementasi *path* dari model yang ada. Untuk Talkback diimplementasi dengan memanggil permission seperti pada gambar 4.10.

```
class TextToSpeechHelper(private val context: Context) {
    private lateinit var textToSpeech: TextToSpeech
    private val accessibilityManager = context.getSystemService(Context.ACCESSIBILITY_SERVICE) as AccessibilityManager

    fun initializeTTS() {
        textToSpeech = TextToSpeech(context) { status ->
            if (status != TextToSpeech.ERROR) {
                textToSpeech.language = Locale.US
            }
        }
    }

    fun playText(text: String) {
        if (accessibilityManager.isTouchExplorationEnabled) {
            // TalkBack aktif, baca teks menggunakan TextToSpeech
            textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null, null)
        } else {
            // Jika TalkBack tidak aktif, cukup cetak teks ke konsol
            println(text)
        }
    }
}
```

Gambar 4.10. Potongan Code untuk Implementasi Talkback

Implementasi Talkback dilakukan agar pengguna dapat menerima respon dari smartphone. VOSK API diimplementasikan agar pengguna dapat memberikan input tujuan.

2. Data akurasi lokasi pengguna serta akurasi rute

Data yang akan diperoleh dari proses koneksi BLE terdekat dengan RSSI sebagai tolak ukur akurasi lokasi pengguna, sedangkan untuk akurasi rute menggunakan data pemetaan rute dan lokasi tujuan yang telah dipetakan pada aplikasi dengan memanfaatkan RSSI dari beberapa koneksi BLE terdekat yang saling terhubung menggunakan Kalman Filter. Implementasi perhitungan Kalman Filter dapat dilihat pada gambar 4.11 berikut.

```
class KalmanFilter(private val initialState: DoubleArray, private val initialCovariance: Array<DoubleArray>) {  
    private var state: DoubleArray = initialState.copyOf()  
    private var covariance: Array<DoubleArray> = initialCovariance.copyOf()  
  
    fun predict(): DoubleArray {  
        // Langkah 1: Prediksi nilai state selanjutnya berdasarkan model gerakan  
        val transitionMatrix = arrayOf(doubleArrayOf(1.0, 0.0), doubleArrayOf(0.0, 1.0))  
        state = multiplyMatrix(transitionMatrix, state)  
  
        // Langkah 2: Prediksi kovarian state selanjutnya berdasarkan model gerakan dan proses noise  
        val processNoise = arrayOf(doubleArrayOf(0.01, 0.0), doubleArrayOf(0.0, 0.01))  
        covariance = addMatrices(multiplyMatrix(multiplyMatrix(transitionMatrix, covariance), transposeMatrix(transitionMatrix)), processNoise)  
  
        return state  
    }  
  
    fun update(measurement: DoubleArray): DoubleArray {  
        // Langkah 1: Hitung residual (perbedaan antara pengukuran aktual dan prediksi)  
        val residual = subtractVectors(measurement, state)  
  
        // Langkah 2: Hitung matriks H, yang menghubungkan pengukuran ke state  
        val measurementMatrix = arrayOf(doubleArrayOf(1.0, 0.0), doubleArrayOf(0.0, 1.0))  
  
        // Langkah 3: Hitung error kovarian pengukuran  
        val measurementNoise = arrayOf(doubleArrayOf(0.01, 0.0), doubleArrayOf(0.0, 0.01))  
        val measurementCovariance = addMatrices(multiplyMatrix(multiplyMatrix(measurementMatrix, covariance), transposeMatrix(measurementMatrix)), measurementNoise)  
  
        // Langkah 4: Hitung matriks Kalman Gain  
        val kalmanGain = multiplyMatrix(covariance, multiplyMatrix(transposeMatrix(measurementMatrix), inverseMatrix(measurementCovariance)))  
  
        // Langkah 5: Perbarui state dan kovarian berdasarkan pengukuran aktual  
        state = addVectors(state, multiplyMatrix(kalmanGain, residual))  
        covariance = multiplyMatrix(subtractMatrices(createIdentityMatrix(2), multiplyMatrix(kalmanGain, measurementMatrix)), covariance)  
  
        return state  
    }  
}
```

Gambar 4.11. Potongan Code untuk perhitungan Kalman Filter

Implementasi Kalman Filter dengan menghitung titik lokasi pengguna secara *real-time* dengan mengambil nilai RSSI dari BLE, kemudian nilai yang diperoleh diteruskan untuk menghitung akurasi rute tujuan pengguna. Implementasi perhitungan akurasi rute dapat dilihat pada gambar 4.13 berikut.

```

fun findShortestPath(startNodeId: Int, targetNodeId: Int): List<FloorNode> {
    // Implementasi algoritma A* untuk mencari jalur terpendek dari startNodeId ke targetNodeId
    val openList = mutableListOf<FloorNode>()
    val closedList = mutableListOf<FloorNode>()
    val parentMap = mutableMapOf<FloorNode, FloorNode>()
    val gScore = mutableMapOf<FloorNode, Double>()
    val fScore = mutableMapOf<FloorNode, Double>()

    // Inisialisasi gScore dan fScore untuk semua node dengan nilai tak terhingga
    for (floorNodes in floorGraph.values) {
        for (node in floorNodes) {
            gScore[node] = Double.POSITIVE_INFINITY
            fScore[node] = Double.POSITIVE_INFINITY
        }
    }

    val startNode = floorGraph.values.flatten().first { it.id == startNodeId }
    val targetNode = floorGraph.values.flatten().first { it.id == targetNodeId }
    gScore[startNode] = 0.0
    fScore[startNode] = heuristicCostEstimate(startNode, targetNode)

    openList.add(startNode)

    while (openList.isNotEmpty()) {
        val current = openList.minByOrNull { fScore[it] ?: Double.POSITIVE_INFINITY } ?: break
        if (current == targetNode) {
            return reconstructPath(parentMap, current)
        }

        openList.remove(current)
        closedList.add(current)

        for (neighbor in getNeighbors(current)) {
            if (neighbor in closedList) {
                continue
            }

            val tentativeGScore = gScore[current]!! + distanceBetween(current, neighbor)

            if (neighbor !in openList) {
                openList.add(neighbor)
            } else if (tentativeGScore >= gScore[neighbor]!!) {
                continue
            }

            parentMap[neighbor] = current
            gScore[neighbor] = tentativeGScore
            fScore[neighbor] = gScore[neighbor]!! + heuristicCostEstimate(neighbor, targetNode)
        }
    }

    return emptyList()
}

```

Gambar 4.12. Potongan Code untuk perhitungan Akurasi Rute

Implementasi akurasi rute menggunakan algoritma A* dengan mencari jalur terpendek dengan memanfaatkan nilai yang diperoleh oleh Kalman Filter.

3. Pembuatan Aplikasi

Aplikasi dibuat dengan menggunakan bahasa pemrograman Kotlin dengan menggunakan *jetpack compose* melalui IDE Android Studio. Media penyimpanan data BLE dan lokasi menggunakan basis data SQLite. Basis data disimpan secara lokal dan dapat diakses menggunakan query SQL berbasis DAO ROOM. BLE dan data lokasi tersimpan berupa Id, RSSI, MAC Address, nama ruangan, alias, dan lokasi lantai.

Proses input pengguna dapat memanggil aplikasi secara langsung menggunakan suara dan memberi lokasi tujuan. Titik mulai pengguna diakses menggunakan Bluetooth dengan mencari BLE terdekat dengan lokasi pengguna. Proses output berupa suara yang berupa text yang dibacakan

menggunakan fitur Talkback pada smartphone saat permintaan inputan, proses pemanduan, serta sampai ke tujuan.



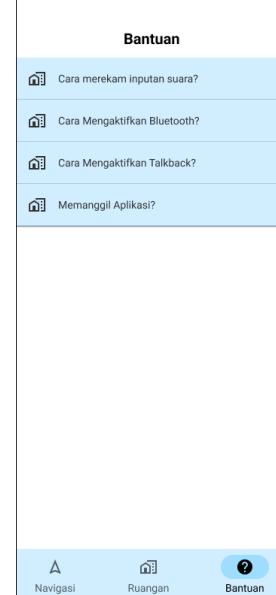
(a) Pop up trigger by hotword



(b) Input tujuan dan lokasi pengguna



(c) Daftar Ruangan



(d) Bantuan

Gambar 4.13. Tampilan Halaman Aplikasi Navigasi Indoor untuk Tunanetra

4.3. Pengujian Sistem

4.3.1. Akurasi rute

Pengujian akurasi posisi dan rute pengguna menggunakan gabungan titik BLE terdekat dengan mengukur kekuatan RSSI, kemudian menggunakan letak dan posisi BLE yang telah dipetakan dengan menghitung menggunakan Kalman Filter yang disertai dengan pembaharuan prediksi titik lokasi pengguna ke tujuan dan menghitung kesalahan prediksi dan estimasi dengan data aktual. Pengujian menggunakan 3 skema pengujian yang di uji pada 5 pengguna tunanetra seperti pada Tabel 4.2 berikut

Tabel 4.2. Skema Pengujian Akurasi Pengambilan Rute di Gedung A FMIPA Optimal dalam MSE (*Mean Squared Error*)

No.	Titik Awal	Tujuan	Pengguna 1	Pengguna 2	Pengguna 3	Pengguna 4	Pengguna 5	Rute Optimal
1	Pintu masuk utama	Toilet lantai 1	1.252	10.254	3.325	2.902	0.873	0.577
2	Pintu masuk utama	Jurusan Informatika	4.213	6.321	3.126	4.234	5.432	2.332
3	Jurusan Informatika	Pintu Keluar Utama	5.432	4.213	6.783	5.126	4.786	2.213

Data Aktual berupa data pengembang dengan menggunakan perhitungan jarak BLE berdasarkan jarak antar BLE dan tujuan akhir secara langsung secara berurut mengikuti BLE terdekat, didapati MSE (*Mean Squared Error*) seperti pada Tabel 4.2. Terdapat pengguna mendapatkan rute yang mendekati rute optimal yang dipilih oleh aplikasi, sehingga pengguna yang mendekati rute optimal memiliki MSE rendah yang berarti mendekati akurat. Terdapat juga pengguna yang memiliki MSE tinggi dapat dipengaruhi oleh beberapa faktor seperti sinyal BLE, lokasi BLE, tembok, lokasi pengguna, dan saran rute yang diberikan melewati rute yang jauh dari rute optimal.

4.3.2. Pengujian Usabilitas UMUX

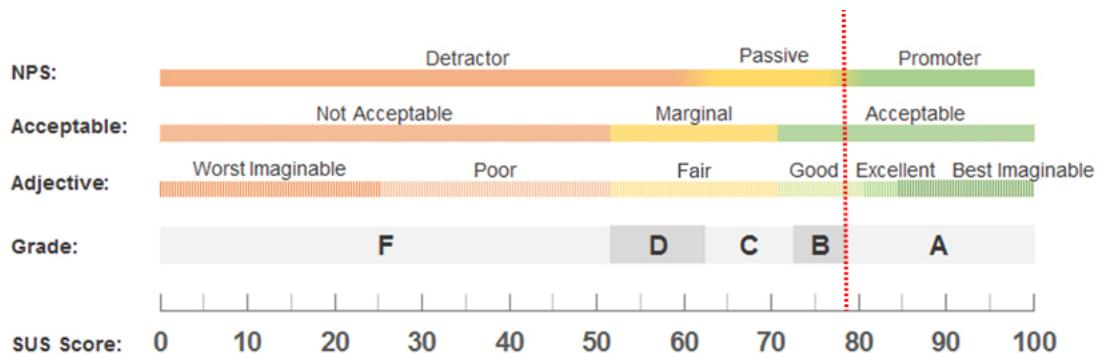
Pengujian usabilitas bertujuan untuk menguji kelayakan dan kegunaan dari sistem yang akan digunakan oleh pengguna. Sebelum pengujian ini dilakukan, adapun skema pengujian yang telah dibuat dapat dilihat pada Tabel 4.2.

Pengujian dengan metode UMUX dilakukan dengan menanyakan pertanyaan kepada responden. Kuesioner berisi 4 pertanyaan dengan menggunakan skala 1-7. Hasil pengujian UMUX aplikasi dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3. Hasil Pengujian UMUX Aplikasi Navigasi

Responden	Kode Pertanyaan				Skor UMUX
	U1	U2	U3	U4	
1	6	2	7	3	83,33
2	5	3	5	2	70,83
3	7	2	6	2	87,5
4	5	2	4	2	70,83
5	5	2	6	2	79,16
Rata-Rata				78,33	

Berdasarkan hasil pengujian UMUX yang telah dilakukan diatas hasil rata-rata pengujian Aplikasi ini mendapatkan skor sebesar 78,33%. Dapat dilihat bahwa aplikasi yang telah dibangun memiliki skor interpretasi "**dapat diterima**", seperti pada gambar 4.14 berikut.



Gambar 4.14. Skor UMUX ke SUS

4.3.3. Pengujian Fungsionalitas Menggunakan BlackBox

Pengujian *Blackbox* dilakukan dengan tujuan untuk menguji fungsionalitas dari aplikasi dengan menjalankan aplikasi tersebut apakah sesuai dengan alur bisni yang diinginkan. Pengujian ini melihat fungsi yang tidak sesuai pada aplikasi dan kesalahan-kesalahan aplikasi dalam mengerjakan suatu perintah. Fitur aplikasi yang diuji dapat dilihat pada Tabel 4.4.

Tabel 4.4. Pengujian Blackbox Aplikasi Navigasi

No.	Nama Pengujian	Skenario	Tampilan	Hasil
1.	Menghidupkan Bluetooth.	Klik tombol Allow pada notifikasi yang muncul.	Bluetooth akan menyala.	Berhasil
2.	Menghidupkan TalkBack.	Klik tombol Allow pada notifikasi yang muncul.	TalkBack akan menyala.	Berhasil
3.	Menghidupkan Mikrofon.	Klik tombol Allow pada notifikasi yang muncul.	Mikrofon akan menyala.	Berhasil
4.	Melihat daftar ruangan.	Klik icon gedung.	Diarahkan ke halaman daftar ruangan.	Berhasil
5.	Melihat daftar bantuan.	Klik icon bantuan.	Diarahkan ke halaman Bantuan.	Berhasil
6.	Memanggil Aplikasi dengan Hotword.	Mengucapkan "Hai Pandu" pada layar mana saja.	Aplikasi terbuka dan siap memandu.	Berhasil
7.	Memeriksa lokasi pengguna.	Proses berjalan di belakang layar.	Notifikasi pengguna berada atau tidak di lokasi.	Berhasil
8.	Melihat Detail Ruangan.	Klik item pada halaman ruangan.	Diarahkan ke halaman detail ruangan.	Berhasil
9.	Proses memandu.	Klik icon mikrofon atau mengucapkan "Hai Pandu" lalu memasukkan tujuan.	Dipandu menuju tujuan dengan navigasi suara.	Berhasil
10.	Melihat detail bantuan.	Klik item pada bantuan.	Diarahkan ke halaman detail ruangan.	Berhasil
11.	Pengecekan Lokasi yang tersedia.	Pengguna memasukkan tujuan, proses pengecekan lokasi dan ruangan berjalan di belakang layar.	Notifikasi ruangan tersedia atau tidak, pengguna berada pada lokasi penelitian atau tidak.	Berhasil
12.	Proses intrupsi untuk pembatalan atau pergantian tujuan lokasi ditengah proses pemanduan.	Pengguna memasukkan tujuan, proses pengecekan lokasi dan ruangan berjalan di belakang layar.	Aplikasi terbuka dan meminta masukkan pengguna serta proses pemanduan akan berjalan.	Berhasil

Berdasarkan hasil Black Box Testing dari tabel diatas menunjukkan bahwa Aplikasi dapat berjalan dengan baik dibuktikan dengan "berhasil" pada kolom hasil pengujian masing-masing fitur yang dikerjakan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut.

1. Indoor Positioning System telah berhasil diimplementasikan dengan menggunakan metode Kalman Filter untuk memprediksi lokasi pengguna di dalam ruangan atau gedung (*indoor*).
2. Model pengenalan suara dapat berhasil memprediksi kata yang disebutkan oleh keempat data uji tambahan pada perangkat *smartphone*, serta dapat memandu pengguna ke tujuan.
3. Berdasarkan hasil pengujian akurasi Kalman Filter dengan MSE (*Mean Squared Error*) Terdapat pengguna mendapatkan rute yang mendekati rute optimal yang dipilih oleh aplikasi, sehingga pengguna yang mendekati rute optimal memiliki MSE rendah yang berarti mendekati akurat. Terdapat juga pengguna yang memiliki MSE tinggi dapat dipengaruhi oleh beberapa faktor seperti sinyal BLE, lokasi BLE, tembok, lokasi pengguna, dan saran rute yang diberikan melewati rute yang jauh dari rute optimal.
4. Sistem pengenalan suara telah berhasil diimplementasikan pada aplikasi berbasis Android dengan menggunakan Vosk dan Talkback untuk memandu pengguna.
5. Berdasarkan hasil pengujian usabilitas menggunakan metode UMUX, Aplikasi Navigasi dapat diterima dan mudah digunakan dilihat dari tingkat pemahaman pengguna.

5.2. Saran

Penelitian yang telah dilakukan masih memiliki banyak kekurangan sehingga perlu dikembangkan agar menjadi lebih baik. Berikut beberapa saran yang diberikan.

- (a) Pada penelitian berikutnya dapat menggunakan metode lain dalam membangun sistem aplikasi agar mendapatkan performa yang terbaik dari *dataset* yang ada.
- (b) Pada penelitian berikutnya dapat menambahkan *dataset* serta menambahkan *speaker* yang ada untuk meningkatkan performa dari sistem pengenalan suara dan rute navigasi.
- (c) Pada penelitian selanjutnya dapat menambahkan jenis pengujian dengan cara yang lain seperti menguji sistem navigasi dengan menambahkan *virtual assistant* seperti *chatbot*.
- (d) Sebaiknya ditambahkan lagi jumlah Beacon di setiap ruangan supaya meningkatkan keakuratan klasifikasi.
- (e) Pengoptimisasi algoritma perhitungan jarak juga dibutuhkan, guna mengurangi waktu komputasi saat melakukan pencarian ruangan apabila ruangan sudah banyak.

DAFTAR KEPUSTAKAAN

- Ammann, P. dan Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.
- Bai, L., Ciravegna, F., Bond, R., dan Mulvenna, M. (2020). A low cost indoor positioning system using bluetooth low energy. *Ieee Access*, 8:136858–136871.
- Barnum, C. M. (2020). *Usability testing essentials: ready, set... test!* Morgan Kaufmann.
- Brena, R. F., García-Vázquez, J. P., Galván-Tejada, C. E., Muñoz-Rodriguez, D., Vargas-Rosales, C., dan Fangmeyer, J. (2017). Evolution of indoor positioning technologies: A survey. *Journal of Sensors*, 2017.
- Cantón Paterna, V., Calveras Auge, A., Paradells Aspas, J., dan Perez Bullones, M. A. (2017). A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering. *Sensors*, 17(12):2927.
- Cephei, A. (2019). Introduction: Vosk.
- Craig, C. dan Gerber, A. (2015). *Learn Android Studio: Build Android Apps Quickly and Effectively*. Apress.
- Deemer, P., Benefield, G., Larman, C., dan Vodde, B. (2012). A lightweight guide to the theory and practice of scrum. *Ver*, 2:2012.
- Finstad, K. (2010). The usability metric for user experience. *Interacting with Computers*, 22(5):323–327.
- Ihsan, M., Suwastika, N. A., dan Putrada, A. G. (2018). Analisis implementasi kalman filter untuk peningkatan akurasi pendektsian pada sistem monitoring area terlarang sepeda motor berbasis m2m. *eProceedings of Engineering*, 5(1).
- Jaya, T. S. (2018). Pengujian aplikasi dengan metode blackbox testing boundary value analysis (studi kasus: kantor digital politeknik negeri lampung). *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1):45–48.
- Jonathan, J. dan Suyanto, Y. (2020). Sintesis suara bernyanyi dengan teknologi text-to-speech untuk notasi musik angka dan lirik lagu berbahasa indonesia. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 10(1):1–10.
- Kaluža, M., Beg, K., dan Vukelić, B. (2017). Analysis of an indoor positioning systems. *Zbornik Veleučilišta u Rijeci*, 5(1):13–32.
- Karimi, H. A. (2015). *Indoor wayfinding and navigation*. CRC Press.

- Ko, E. dan Kim, E. Y. (2017). A vision-based wayfinding system for visually impaired people using situation awareness and activity-based instructions. *Sensors*, 17(8):1882.
- Lewis, J. R. (2018). Measuring perceived usability: The csuq, sus, and umux. *International Journal of Human–Computer Interaction*, 34(12):1148–1156.
- Li, G., Geng, E., Ye, Z., Xu, Y., Lin, J., dan Pang, Y. (2018). Indoor positioning algorithm based on the improved rssi distance model. *Sensors*, 18(9):2820.
- Li, Q., Li, R., Ji, K., dan Dai, W. (2015). Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 74–77. IEEE.
- Lin, Y.-W. dan Lin, C.-Y. (2018). An interactive real-time locating system based on bluetooth low-energy beacon network. *Sensors*, 18(5):1637.
- Mustikarini, W., Hidayat, R., dan Bejo, A. (2019). Real-time indonesian language speech recognition with mfcc algorithms and python-based svm. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 3(2):55–60.
- Nasution, A. dkk. (2018). Perancangan aplikasi push notification berbasis android. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 4(2):149–154.
- Puspitasari, R. (2020). Rancang Bangun Aplikasi Kehadiran Perkuliahinan Berbasis Teknologi Indoor Positioning System Menggunakan Bluetooth Low Energy dan Metode Klasifikasi K-NN. *ETD Unsyiah*.
- Qin, F., Zuo, T., dan Wang, X. (2021). Ccpos: Wifi fingerprint indoor positioning system based on cdae-cnn. *Sensors*, 21(4):1114.
- Rawat, D. B. dan Ghafoor, K. Z. (2018). *Smart cities cybersecurity and privacy*. Elsevier.
- Schwaber, K. dan Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21(19):1.
- Shaheen, J. A., Asghar, M. A., dan Hussain, A. (2017). Android os with its architecture and android application with dalvik virtual machine review. *International Journal of Multimedia and Ubiquitous Engineering*, 12(7):19–30.
- Sideeq, M. M. dan Qasem, N. (2016). Smart wall based on active frequency selective wallpaper. *ZANCO Journal of Pure and Applied Sciences*, 28(2).

- Subhan, F., Hasbullah, H., Rozzyev, A., dan Bakhsh, S. T. (2011). Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *2011 International Conference on Information Science and Applications*, pages 1–9. IEEE.
- Welch, G. F. (2020). Kalman filter. *Computer Vision: A Reference Guide*, pages 1–3.
- Wesfix, T. (2017). *Branding itu Dipraktekin*. Gramedia Widiasarana Indonesia.
- WHO (accessed Oktober 23, 2020). World report on vision. <https://www.who.int/publications-detail-redirect/world-report-on-vision>.
- Xiao, A., Chen, R., Li, D., Chen, Y., dan Wu, D. (2018). An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras. *Sensors*, 18(7):2229.
- Xu, S., Chen, L., Wang, C., dan Rud, O. (2016). A comparative study on black-box testing with open source applications. In *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 527–532. IEEE.
- Yadav, R. K., Bhattacharai, B., Gang, H.-S., dan Pyun, J.-Y. (2019). Trusted k nearest bayesian estimation for indoor positioning system. *IEEE Access*, 7:51484–51498.
- Yudha, D. P., Hasbi, B. I., dan Sukarna, R. H. (2018). Indoor positioning system berdasarkan fingerprinting received signal strength (rss) wifi dengan algoritma k-nearest neighbor (k-nn). *ILKOM Jurnal Ilmiah*, 10(3):274–283.
- Zafari, F., Gkelias, A., dan Leung, K. K. (2019). A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 21(3):2568–2599.