

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

PROPOSAL

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
JUNI, 2022**

PENGESAHAN PROPOSAL

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

DESIGN OF NAVIGATION SYSTEM FOR ROUTE GUIDANCE APPLICATION FOR VISUALLY IMPAIRED PERSON BASED ON INDOOR POSITIONING

Oleh:

Nama : Yuda Aditya
NPM : 1608107010030
Jurusan : Informatika

Menyetujui:

Pembimbing I

Pembimbing II

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

NIP. 198003262014041001

NIP.

Mengetahui:

Ketua Jurusan Informatika FMIPA
Universitas Syiah Kuala,

Dr. Muhammad Subianto, S.Si., M.Si

NIP. 196812111994031005

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 proposal 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.

Proposal yang berjudul **“Rancang Bangun Sistem Navigasi pada Aplikasi *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 Proposal ini.
2. Bapak Kurnia Saputra, S.T., M.Sc., selaku Dosen Pembimbing I dan , selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan dan arahan kepada penulis, sehingga penulis dapat menyelesaikan Proposal 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 Proposal ini.
7. Seluruh teman-teman seperjuangan Jurusan Informatika Unsyiah 2016 lainnya.

Penulis juga menyadari segala ketidaksempurnaan 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 Proposal ini. Harapan penulis semoga tulisan ini dapat bermanfaat bagi banyak pihak dan untuk perkembangan ilmu pengetahuan.

Banda Aceh, Juni 2022

Penulis

DAFTAR ISI

PENGESAHAN PROPOSAL	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vi
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	4
2.1. <i>Indoor Positioning System</i>	4
2.2. <i>Route Guidance System / Wayfinding and Navigation System</i> . .	5
2.3. Kalman Filter	5
2.4. <i>Fingerprinting</i>	5
2.5. <i>Received Signal Strength Indicator</i> (RSSI)	6
2.6. <i>Bluetooth Low Energy</i> (BLE)	7
2.7. <i>Beacon</i>	7
2.8. <i>Speech Recognition</i>	8
2.9. <i>Text-to-Speech</i> (TTS)	8
2.10. KALDI Toolkit	8
2.11. VOSK API	8
2.12. Android	9
2.13. Android Studio	10
2.14. SCRUM	10
2.15. <i>Black Box Testing</i>	12
2.16. <i>Usability Testing</i>	13
2.17. <i>System Usability Scale</i> (SUS)	13
BAB III METODOLOGI PENELITIAN	15
3.1. Waktu dan Lokasi Penelitian	15
3.2. Alat dan Bahan	15
3.3. <i>Roadmap</i> Penelitian	15
3.4. Metode Penelitian	19
3.4.1. Data Suara	19
3.4.2. <i>Pre-processing</i> Data Suara	20
3.4.3. Persiapan Data untuk Kaldi	20
3.4.4. Ekstraksi Fitur	22
3.4.5. Akustik Model	22
3.4.6. Model	23
3.4.7. Model Terlatih	23

DAFTAR GAMBAR

Gambar 2.1	Ilustrasi Metode <i>Fingerprinting</i> (Yudha dkk., 2018)	6
Gambar 2.2	Ilustrasi Metode Pengembangan Menggunakan Scrum (Schwaber dan Sutherland, 2011).	12
Gambar 3.1	<i>Roadmap</i> Penelitian Fase 1	17
Gambar 3.2	<i>Roadmap</i> Penelitian Fase 2	18
Gambar 3.3	Diagram Perancangan Sistem Pengenalan Ucapan	19

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 di antaranya memiliki gangguan penglihatan atau kebutaan, di mana setidaknya 1 miliar di antaranya memiliki gangguan penglihatan yang dapat dicegah atau belum ditangani (WHO, 2020).

Seiring perkembangan zaman di mana kemudahan berkomunikasi dapat dibantu dengan teknologi yang berkaitan dengan suara telah banyak dikembangkan. Selain itu dengan adanya teknologi ini 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, hal simpel seperti menuju ke toilet, ke kamar tidur, dan dapur sebagai contoh.

Untuk mengurangi kesulitan tunanetra dan membantu mereka melokalisasi posisi mereka dan menemukan destinasi tujuan mereka. Penelitian dan teknologi terbaru berfokus pada *Global Positioning Systems* (GPS). Namun, saat ini sistem yang menggunakan sensor GPS bekerja dengan baik sebagai alat pencari rute di lingkungan 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 (Virrantaus dkk., 2001). Teknologi IPS ini nantinya akan di gunakan sebagai dasar dari *Wayfinding System*

atau *Route Guidance System* dan akan dipadukan dengan *Speech Command Recognition*.

Speech Recognition merupakan salah satu bentuk dari teknologi yang disebut *Artificial Intelligence (AI)*. *Speech Recognition* sendiri merupakan teknologi pada perangkat komputer agar dapat mengenali dan memahami kata-kata yang diucapkan oleh manusia. Kata-kata yang diucapkan memiliki gelombang suara tertentu, gelombang tersebut akan diterima oleh perangkat akan diterjemahkan menjadi format yang dapat dimengerti oleh perangkat. Dengan begitu perangkat dapat mengidentifikasi serta memahami kata yang dimasukkan yang merupakan konsep dasar dari *Voice Command*. 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. Sedangkan *Text-to-Speech* merupakan teks yang diterjemahkan oleh perangkat menjadi suara sebagai keluaran dari perangkat sehingga suara dapat didengar oleh pengguna. 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, yaitu IPS dengan menggunakan *Bluetooth Low Energy (BLE)* dan *Speech Command Recognition* sebagian 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 konektivitas yang luas (Puspitasari, 2020). Pada proses implementasi, Gedung Fakultas Matematika dan Ilmu Pengetahuan Alam USK akan dipasang beberapa alat transmisi BLE pada setiap ruang-ruang penting yang sering dijadikan tujuan, seperti ruangan Jurusan Informatika, Ruang Akademik, Toilet dsb. 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 masukan 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 *Voice Command Recognition* untuk *Route Guidance* berbasis *Indoor Positioning* untuk tunanetra.
2. Bagaimana aplikasi dapat membantu serta memandu tunanetra agar sampai ke tujuan berdasarkan lokasi 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 tingkat akurasi dari lokasi pengguna serta akurasi dari pemandu saat memandu pengguna dengan suara.

1.3. Tujuan Penelitian

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

1. Membangun model sistem pengenalan suara untuk digunakan pada aplikasi *route guidance* berbasis *indoor positioning*.
2. Menganalisis tingkat akurasi yang dihasilkan dari pembangunan model sistem pengenalan suara.

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.
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, di mana 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. 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).

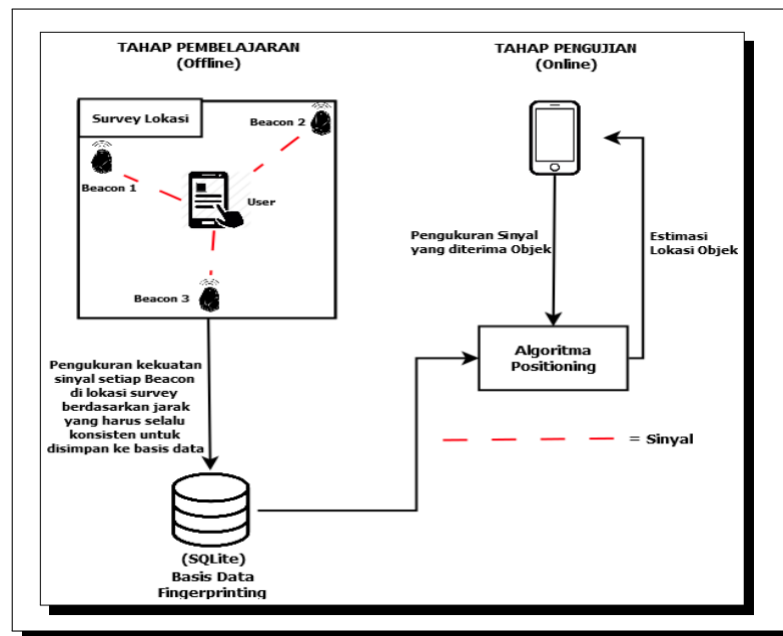
2.4. Fingerprinting

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 transmitter dan *receiver*.

Transmitter untuk teknologi IPS yang ditunjukkan untuk WLAN disebut dengan AP, sedangkan transmitter 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), di mana 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*), di mana 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 2.1.



Gambar 2.1. Ilustrasi Metode *Fingerprinting* (Yudha dkk., 2018)

2.5. Received Signal Strength Indicator (RSSI)

Received Signal Strength Indicator atau RSSI adalah metode pengukuran jarak yang memperoleh sinyal dari transmitter 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 transmitter (Puspitasari, 2020).

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

No.	<i>Signal Noise Ratio</i> (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

Nilai RSSI didefinisikan dengan bilangan negatif. Semakin tinggi bilangan negatifnya, 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 pada 2.1

2.6. *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 spesialis 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).

2.7. *Beacon*

BLE memancarkan sinyal dari alat transmiter yang beroperasi menggunakan baterai. Alat transmiter tersebut disebut dengan *Beacon*. *Beacon* merupakan alat pendeteksi 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 mendeteksi 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).

2.8. *Speech Recognition*

Speech Recognition adalah kemampuan suatu komputer agar dapat mengenali apa yang diucapkan oleh seseorang berdasarkan sinyal suara yang diucapkan oleh seseorang atau bisa disebut sebagai *Automatic Speech Recognition* (ASR)(Jurafsky dan Martin, 2008). *Speech Recognition* merupakan antarmuka alami untuk berkomunikasi dengan peralatan komputer seperti *smart home appliances*, *personal assistants*, atau *smartphone*. ASR Juga berguna untuk *general transcription*, contohnya untuk membuat *captions* secara otomatis untuk audio atau video (mentranskripsikan film atau video atau *live discussions*) (Jurafsky dan Martin, 2008). Menjadi suatu kemudahan bagi manusia jika komputer dapat memahami apa yang diucapkan oleh manusia dan karena hal itu juga manusia dapat dengan mudah mengoperasikan komputer karena adanya teknologi yang *Automatic Speech Recognition*. serta *speech synthesis* atau *text-to-speech* (TTS) (Jurafsky dan Martin, 2008).

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

Speech Syntesis atau text-to-speech merupakan kebalikan dari ASR dalam memetakan teks ke bentuk gelombang akustik. TTS memiliki berbagai macam aplikasi. 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, seperti mendiang Steven Hawking seorang ahli astrofisika berbicara dengan memanipulasi sistem TTS (Jurafsky dan Martin, 2008).

2.10. *KALDI Toolkit*

Kaldi adalah toolkit open-source untuk speech recognition, Kaldi ditulis dalam bahasa C++ dan di bawah lisensi Apache v2.0, Kaldi bergantung dengan library finite-state transducers (menggunakan OpenFst) serta library aljabar linier ekstensif menggunakan "Basic Linear Algebra Subroutines" (BLAS) dan "Linear Algebra PACKage" (LAPACK) (Povey dkk., 2011). Pada penelitian ini akan digunakan model yang dihasilkan oleh Kaldi untuk membangun model yang akan digunakan pada VOSK API.

2.11. *VOSK API*

Vosk api merupakan toolkit untuk speech recognition, di mana 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.

2.12. 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.

2.13. Android Studio

Android Studio merupakan *Integrated Development Environment* (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.

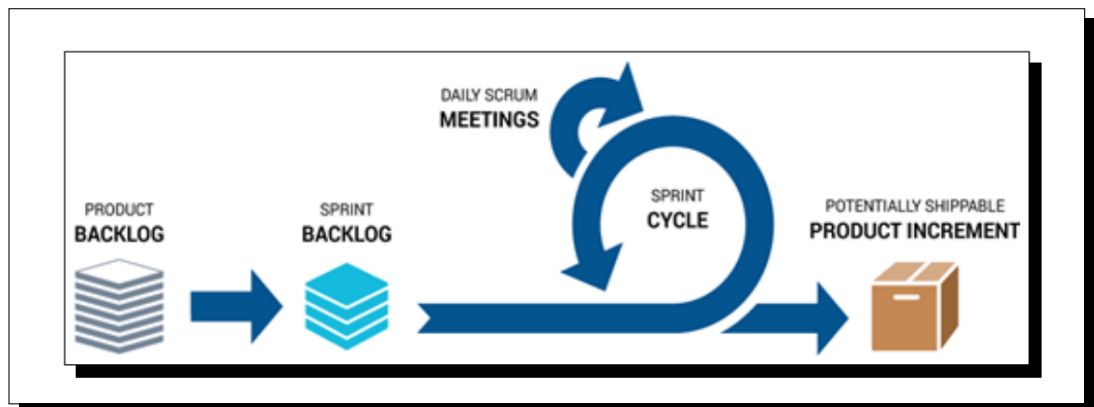
2.14. SCRUM

Scrum adalah *development framework* di mana 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 memperbaharui 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* di mana *Product Owner*, satu orang yang 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* di mana 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 2.2 di bawah ini.



Gambar 2.2. Ilustrasi Metode Pengembangan Menggunakan Scrum (Schwaber dan Sutherland, 2011).

2.15. *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, di antaranya penguji tidak perlu mengetahui suatu bahasa pemrograman, pengujian dilakukan dari sudut pandang pengguna agar dapat mengetahui suatu inkonsistensi 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).

2.16. 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 (Rubin dan Chisnell, 2008). Dengan *usability testing*, didapati 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 (Wahl, 2000).

2.17. System Usability Scale (SUS)

Kuesioner standar yang paling sering digunakan untuk penilaian suatu kegunaan merupakan *System Usability Scale* (SUS). Dari awal kuesioner ini relatif tidak menguntungkan, namun dimasa depan kemungkinan besar SUS akan menjadi suatu ukuran yang populer dari persepsi kegunaan. Mendapatkan nilai pada SUS cukup rumit, karena pola pada *item* yang bergantian dan memiliki keputusan awal untuk memanipulasi skor berada di antara 0 sampai 100 (Lewis, 2018). John Brooke

mengembangkan SUS pada tahun 1996 yang berisi 10 pertanyaan dasar dan sederhana tentang kegunaan suatu sistem, berikut adalah pertanyaannya (Kaya dkk., 2019):

1. Saya pikir saya ingin menggunakan sistem ini dengan sering.
2. Saya menemukan sistem yang tidak rumit.
3. Saya pikir sistemnya mudah digunakan.
4. Saya pikir saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.
5. Saya menemukan berbagai fungsi dalam sistem ini terintegrasi dengan baik.
6. Saya pikir ada terlalu banyak inkonsistensi dalam sistem ini.
7. Saya membayangkan bahwa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.
8. Menurut saya sistem ini sangat rumit untuk digunakan.
9. Saya merasa sangat percaya diri menggunakan sistem ini.
10. Saya perlu belajar banyak hal sebelum saya bisa menggunakan sistem ini.

Pertanyaan tersebut akan diberikan kepada peserta uji dan menjawab dengan skala antara 1 (Sangat tidak setuju) dan 5 (Sangat setuju). Menurut Brooke hasil dari jawaban akan dievaluasi dalam kisaran 0 sampai 4. Dapat dilihat pada pertanyaan yang memiliki nomor ganjil maka memiliki makna positif dan pertanyaan bernomor genap memiliki makna negatif. Pada pertanyaan positif dilakukan penilaian yang mana skor pengguna dikurangi 1 poin dan pada pertanyaan negatif dilakukan penilaian yang mana skor pengguna dikurangi 5 poin. Setelah itu jumlah skor yang didapat dikalikan 2,5 untuk membuat kisaran antar 0 sampai 100. Skor SUS rata-rata dan skala peringkat kata sifat dihitung untuk setiap aplikasi seluler di Android dan iOS (Kaya dkk., 2019).

BAB III

METODOLOGI PENELITIAN

3.1. Waktu dan Lokasi Penelitian

Penelitian ini dilakukan di Fakultas Matematika dan Ilmu Pengetahuan Alam Gedung A Universitas Syiah Kuala. Waktu yang dibutuhkan untuk penelitian ini adalah 4 bulan terhitung dari bulan Maret 2021 hingga Juli 2021.

3.2. Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini meliputi perangkat keras, perangkat lunak, dan data. Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

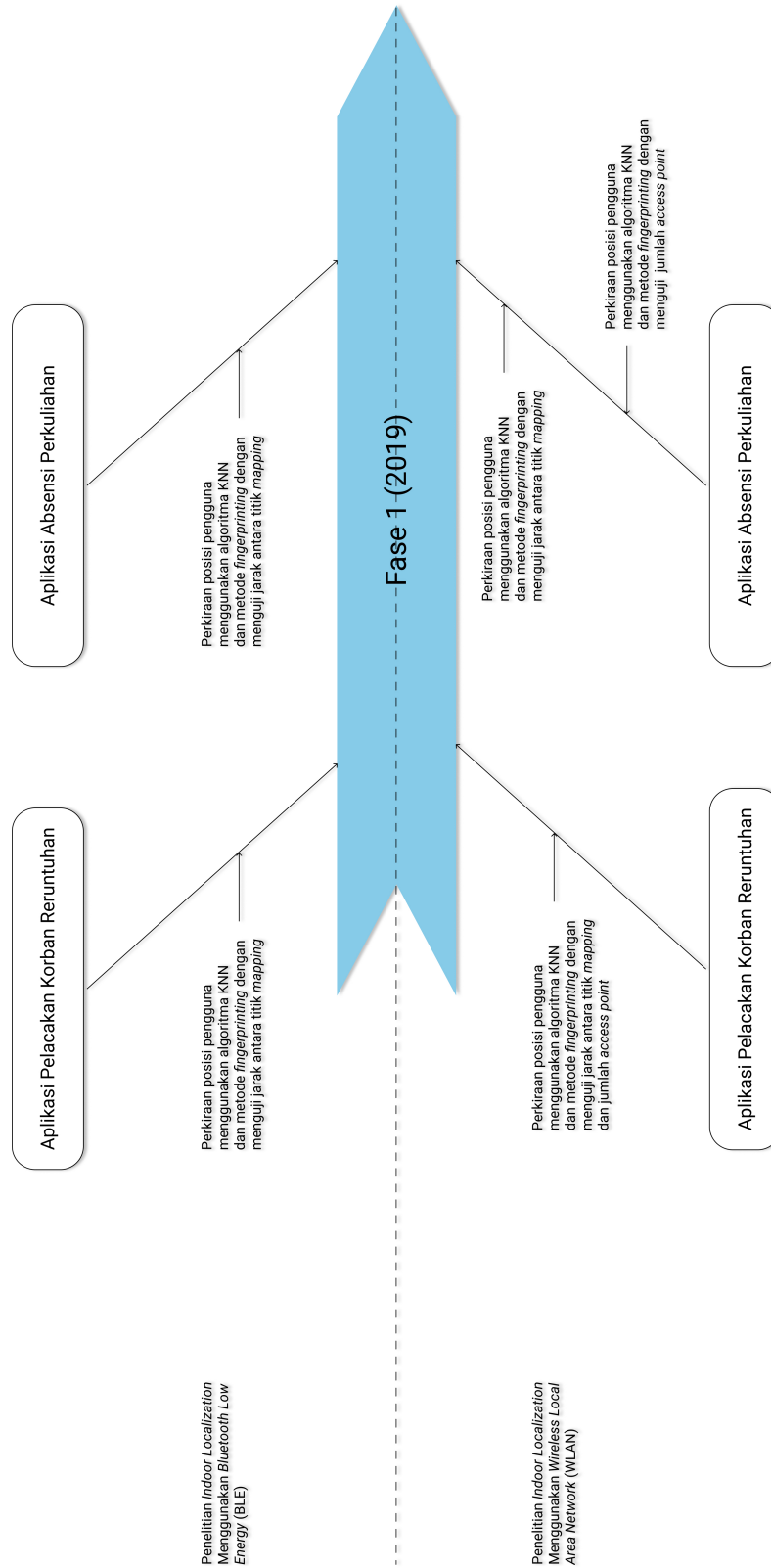
- Windows 10 64 bit
- Linux Ubuntu 20.04.1
- Kaldi
- Vosk-api
- Python
- Visual Studio Code 1.50.1
- Google Form
- Freemake Audio Converter 1.1.9
- SRILM 1.7.3

Sedangkan perangkat keras yang digunakan pada penelitian ini adalah 1 unit Laptop Acer Aspire A514-52G dengan RAM 12 GB, *Processor* Intel® Core™ i5-10210U @ 1.60 GHz (8 CPUs), 2.1GHz, Kartu grafis NVIDIA GeForce MX250 dan Harddisk 1 TB.

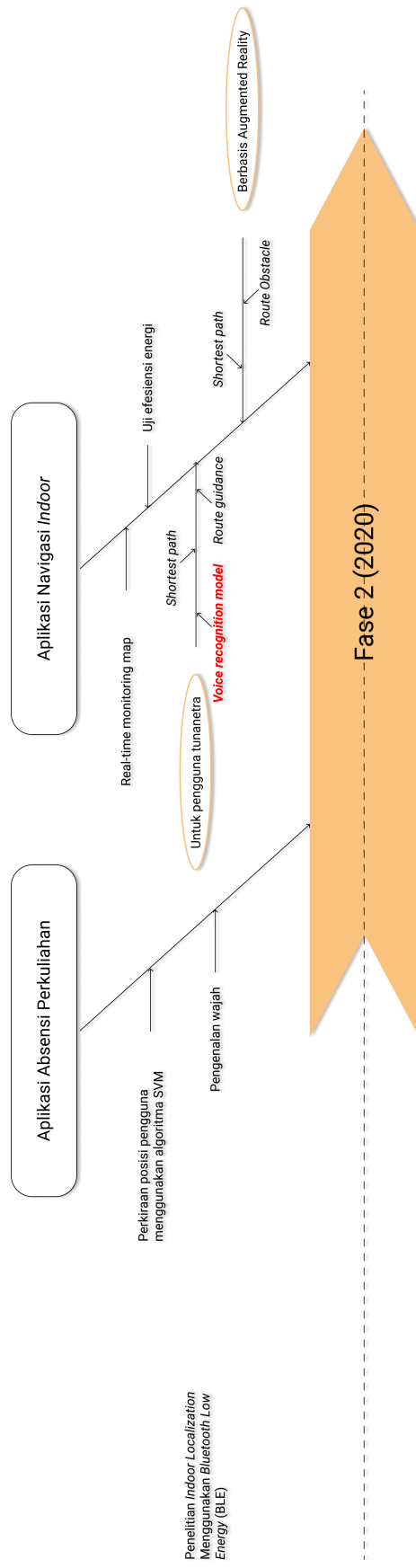
3.3. Roadmap Penelitian

Roadmap pada penelitian ini merupakan diagram yang menggambarkan rangkaian beberapa penelitian yang memiliki kesinambungan dalam rentang waktu 2019 sampai dengan 2020 yang dibagi menjadi 2 fase. Fase pertama pada tahun 2019 memiliki dua fokus penelitian *indoor localization* dengan menggunakan *Bluetooth Low Energy* (BLE) dan menggunakan *Wireless Local Area Network*

(WLAN) dapat dilihat pada gambar 3.1. Lalu pada fase kedua pada tahun 2020 lebih berfokus pada penelitian *indoor localization* dengan menggunakan BLE dapat dilihat pada gambar 3.2. Aplikasi Navigasi *Indoor* dibangun untuk Fakultas MIPA, karena memiliki gedung yang cukup luas dan memiliki banyak sub Gedung yang dibagi menjadi 6 bagian dari Gedung A sampai Gedung F. Aplikasi ini dapat berjalan dan memberikan arahan pada pengguna di dalam ruangan. Salah satu bagian dari aplikasi Navigasi *Indoor* ini adalah aplikasi *route guidance* untuk pengguna tuna netra. Aplikasi *route guidance* ini menggunakan *voice recognition* sebagai penghubung untuk mempermudah bagi pengguna tuna netra agar dapat menjalankan aplikasi ini. 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 model *voice recognition* seperti yang ditunjukkan pada gambar 3.2 dengan kata dicetak tebal berwarna merah sebagai berikut.



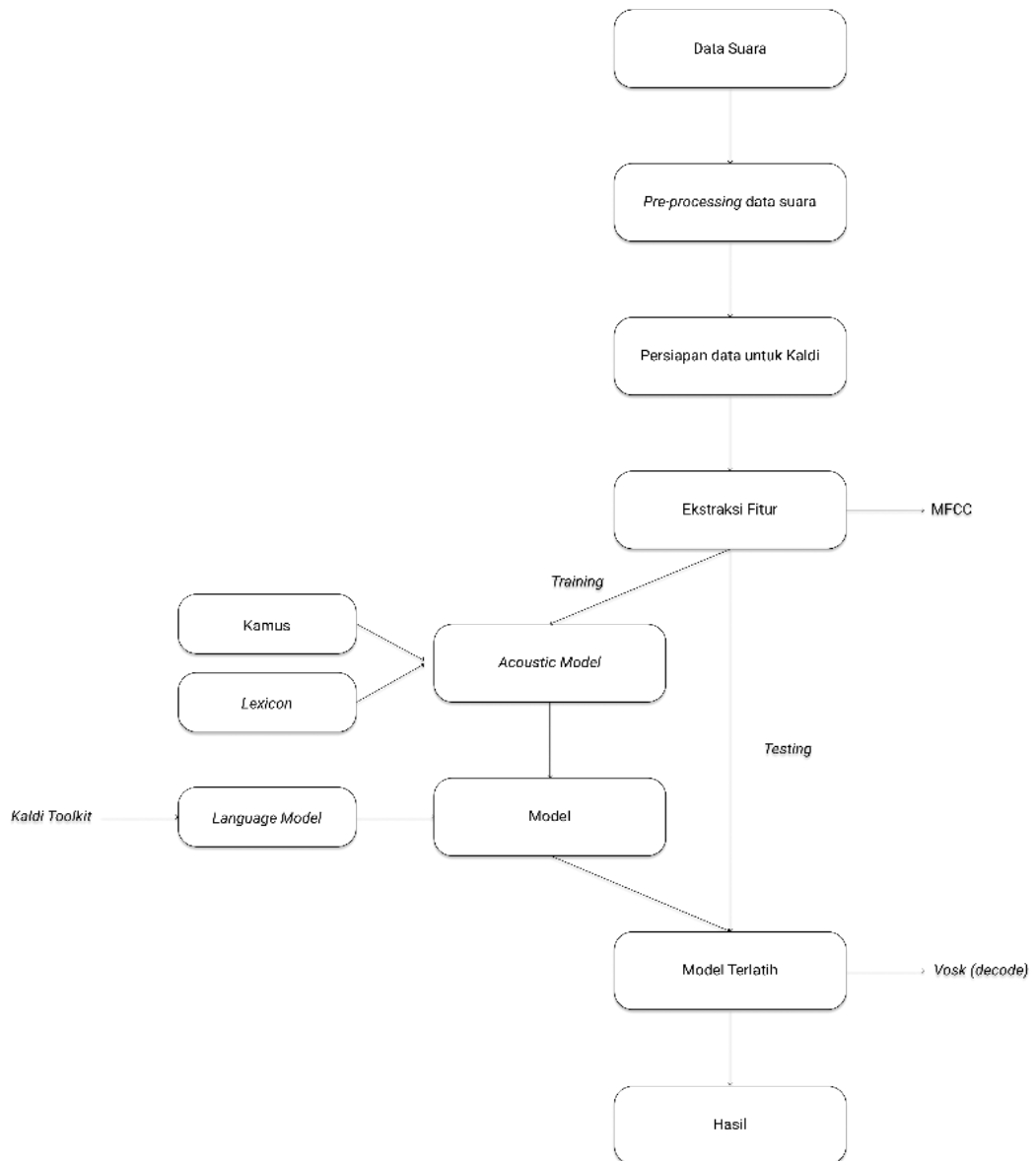
Gambar 3.1. Roadmap Penelitian Fase 1



Gambar 3.2. Roadmap Penelitian Fase 2

3.4. Metode Penelitian

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



Gambar 3.3. Diagram Perancangan Sistem Pengenalan Ucapan

3.4.1. Data Suara

Pada tahap ini memiliki 2 hal yang dilakukan, yang pertama pembuatan transkrip dari nama-nama ruangan pada Gedung A FMIPA Unsyiah dan pengambilan data suara dari narasumber. Pada penulisan transkrip dilakukan dengan menggali informasi nama-nama ruangan pada Gedung A FMIPA dari beberapa orang. Setelah itu dilakukan proses pengambilan data suara oleh beberapa narasumber yang berjumlah 20 orang, memiliki rentang umur 17 tahun sampai

dengan 58 tahun, memiliki 10 orang yang memiliki jenis kelamin laki-laki dan 10 orang memiliki jenis kelamin perempuan. Pengambilan suara dilakukan dengan 2 cara. Pertama melalui proses perekaman suara masing-masing melalui *smartphone* masing-masing lalu di *input* melalui *form* yang sudah dibuat dengan Google Form. Kedua dilakukan perekaman suara dengan narasumber secara langsung tanpa melalui Google Form. Sebelum proses perekaman suara, narasumber diberikan 56 daftar nama-nama ruangan pada gedung A FMIPA Unsyiah serta diberi pengarahan dalam pengucapan dari daftar yang telah diberikan.

3.4.2. *Pre-processing* Data Suara

Pada tahap ini dilakukan *pre-processing* terhadap data suara yang sudah dimasukkan oleh narasumber. Tahap ini dilakukan karena narasumber melakukan proses perekaman suara secara mandiri dengan *smartphone* masing-masing, jadi memiliki tipe *file* audio yang berbeda-beda. *File* audio diubah menjadi tipe *file* *.wav dengan *mono channel*, 16KHz *sample rate* dan *sample size* 16 bit. Aturan tersebut sesuai dengan yang diterima oleh Kaldi dan Vosk-api. Proses mengubah tipe *file* dengan menggunakan aplikasi Freemake Audio Converter, aplikasi ini dapat diinstalasi secara gratis dan dapat digunakan untuk sistem operasi Windows. Selain mengubah tipe *file*, pada tahapan ini juga dilakukan pencocokan data berdasarkan nama-nama ruangan yang dikumpulkan menjadi satu *folder*. Sehingga pada penelitian ini memiliki 20 *folder* berbeda yang berisikan 56 *file*. 56 *file* tersebut merupakan jumlah narasumber atau yang dapat kita sebut di sini adalah *speaker*.

3.4.3. Persiapan Data untuk Kaldi

Pada tahapan ini ada beberapa *file* yang harus disiapkan untuk membuat model *speech recognition*, yaitu ada persiapan data akustik untuk membuat *acoustic model* dan *language data* untuk membuat *language model*. Sebelum mempersiapkan beberapa *file* yang dibutuhkan, data suara yang telah dikumpulkan di bagi menjadi 2 bagian, di mana 80% *speaker* untuk *data training* dan 20% *speaker* untuk *data testing*. Berikut adalah penjelasan dari persiapan data akustik dan *language data*:

1. Persiapan data akustik

Pada tahapan ini, ada 5 *file* yang harus dibuat agar Kaldi dapat memahami data audio yang akan diproses. Berikut adalah penjelasan *file* yang harus dibuat:

- Spk2gender

Pada *file* ini berisikan informasi tentang nama yang diasumsikan sebagai *id* dari *speaker* dan jenis kelamin *speaker* tersebut. *File* ini memiliki

pattern <speakerID> <jenis kelamin>, di mana jenis kelamin diinisialkan f sebagai perempuan dan m sebagai pria.

- *Wav.scp*

Pada *file* ini berisikan informasi *utteranceID* dan lokasi audio tersebut ditambahkan dengan nama file tersebut, dalam hal ini penulisan *utteranceID* merupakan *speakerID* yang disambung dengan nama-nama ruangan yang diucapkan pada audio tersebut. Sehingga file ini memiliki *pattern <uterranceID> <lokasi_file_audio>*.

- *Text*

Pada *file* ini berisikan informasi tentang *utteranceID* dan transkrip dari nama-nama ruangan di Gedung A FMIPA Unsyiah yang diucapkan oleh *speaker*. Sehingga file ini memiliki *pattern <uterranceID> <text_transcription>*.

- *Utt2spk*

Pada *file* ini berisikan informasi berupa *utteranceID* dan *speakerID*, agar sistem pengenalan suara dapat mengetahui nama-nama ruangan yang diucapkan oleh *speaker* tertentu. Sehingga file ini memiliki *pattern <uterranceID> <speakerID>*.

- *Corpus*

Pada *file* ini berisikan informasi berupa transkrip dari nama-nama ruangan di Gedung A FMIPA Unsyiah yang diucapkan oleh *speaker*, namun *file* ini disimpan pada lokasi yang berbeda dari beberapa *file* yang sudah disebutkan sebelumnya. *File* ini memiliki *pattern <text_transcription>* yang ditulis per baris, sehingga *file* pada penelitian ini terdapat 56 baris.

2. Persiapan *language data*

Pada tahapan ini, ada 4 *file* berkaitan dengan *language model*. Berikut adalah penjelasan *file* yang harus dibuat:

- *Lexicon*

Pertama membuat *file* yang bernama *lexicon.txt*. Pada *file* ini berisikan setiap kata dari *dictionary* dengan penambahan *phone transcriptions* atau penyebutan dari kata tersebut. Seperti contoh, jika ada kata 'satu' maka penyebutan dari kata tersebut adalah sa tu. Namun, jika ada suara yang diam akan dideteksi sebagai *silence phone* dan penyebutannya akan ditandai dengan 'sil'. Sehingga file ini memiliki *pattern <word> <phone 1> <phone 2>*.

- *Nonsilence phones*

Lalu pada *file* kedua diberi nama *nonsilence_phones.txt*. Pada *file* ini berisikan *list phone transcriptions* yang *nonsilence phones*. Yang termasuk pada *file* ini adalah semua *phones* pada *file lexicon* kecuali *phones* seperti ‘sil’ atau diam. *File* ini memiliki *pattern*: *<phone>*.

- *Silence phones*

Pada *file* ketiga diberi nama *silence_phones.txt*. Pada *file* ini berisikan *list phone transcription* yang diam atau ‘sil’ pada bagian *phone*. *File* ini memiliki *pattern*: *<phone>*.

- *Optional silence*

Pada *file* keempat diberi nama *optional_silence.txt*. Pada *file* ini berisikan *list* pilihan lainnya dari *silence phone transcription*. *File* ini memiliki *pattern*: *<phone>*.

3.4.4. Ekstraksi Fitur

Setelah tahapan sebelumnya selesai dilakukan, selanjutnya masuk dalam tahapan ekstraksi fitur. Ekstraksi fitur dilakukan dengan menggunakan metode *Mel Frequency Ceptral Coefficient* (MFCC), agar dapat mengidentifikasi konten linguistik dan membuang suara latar belakang serta suara yang tidak dibutuhkan dalam proses ini. Ekstraksi fitur pada penelitian ini memanfaatkan *open source toolkit* untuk pengolahan suara yang tersedia yaitu Kaldi Toolkit. Pada perhitungan MFCC untuk penelitian ini ada hal yang harus ditentukan, yaitu menentukan jumlah *cepstral coefficient* sebanyak 13 *cepstral coefficient* dan jumlah *frame* dalam suatu *file* sepanjang 25 *millisecond* dan memiliki *window step* sebesar 10 *millisecond*.

3.4.5. Akustik Model

Pada bagian ini merupakan tahapan untuk melakukan model akustik pada data training yang telah melewati proses ekstraksi fitur dengan MFCC. Tahapan ini menggunakan proses *machine learning* dengan metode *Deep Neural Network* (DNN) dan dilakukan juga pengambilan nilai *CTC Loss* yang merepresentasikan akurasi dari *training*. Pada tahap awal *training* dimulai dengan menentukan jumlah *file* yang masuk ke dalam perhitungan ke *deep learning* sebagai bahan dari pembelajaran atau pembagi data. Setelah itu nilai dari MFCC *feature* dimasukkan, lalu label untuk proses pembelajaran dimasukkan juga. Berikutnya dilakukan inisialisasi *learning rate*, maksimal *epoch* dan minimum *loss* yang digunakan. Lalu pada proses perhitungan *neural network* diinisialkan bobot yang dipakai. Selanjutnya metode *deep learning* dengan *input* nilai MFCC *feature* dan target

dihitung. Terakhir menampilkan nilai *loss* dari hasil perhitungan yang didapat. Hal tersebut dilakukan sampai mendapatkan minimum nilai *loss* dan maksimum nilai *epoch*. Pada penelitian ini digunakan *multilayer perseptron* sebagai layer pembelajarannya.

3.4.6. Model

Setelah mendapatkan hasil dari *acoustic modelling* dengan Kaldi, selanjutnya melakukan *language modelling* dengan menggunakan SRILM (*SRI Language Modeling Toolkit*). *Language modelling* berguna agar dapat membedakan kata dan frasa yang terdengar serupa pada ucapannya sehingga didapatkan model pengenalan ucapan yang terlatih.

3.4.7. Model Terlatih

Setelah model pengenalan ucapan sudah selesai dibangun, lalu model tersebut diletakan pada Vosk yang merupakan *open source speech recognition toolkit*, Vosk dapat berjalan pada Android dengan API secara *offline*. Data *testing* yang sebelumnya sudah dipisahkan, diuji pada model tersebut sehingga mendapatkan keluaran berupa *text*. Model ini akan dapat berjalan pada aplikasi Android *route guidance* untuk tuna netra berbasis *Indoor Positioning* yang akan dibangun.

DAFTAR KEPUSTAKAAN

- Ammann, P. dan Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.
- 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.
- 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.
- Jurafsky, D. dan Martin, J. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2.
- 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.
- Kaya, A., Ozturk, R., dan Gumussoy, C. A. (2019). Usability measurement of mobile applications with system usability scale (sus). In *Industrial engineering in the big data era*, pages 389–400. Springer.
- 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). The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590.
- 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.
- Nasution, A. dkk. (2018). Perancangan aplikasi push notification berbasis android. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 4(2):149–154.
- Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., dkk. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society.
- Puspitasari, R. (2020). Rancang Bangun Aplikasi Kehadiran Perkuliahan Berbasis Teknologi Indoor Positioning System Menggunakan Bluetooth Low Energy dan Metode Klasifikasi K-NN. *ETD Unsyiah*.
- Rubin, J. dan Chisnell, D. (2008). *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.
- 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., Rozyyev, 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.
- Virrantaus, K., Markkula, J., Garmash, A., Terziyan, V., Veijalainen, J., Katanosov, A., dan Tirri, H. (2001). Developing gis-supported location-based services. In *Proceedings of the Second International Conference on Web Information Systems Engineering*, volume 2, pages 66–75. IEEE.
- Wahl, N. J. (2000). Student-run usability testing. In *Thirteenth Conference on Software Engineering Education and Training*, pages 123–131. IEEE.
- Welch, G. F. (2020). Kalman filter. *Computer Vision: A Reference Guide*, pages 1–3.
- Wesfix, T. (2017). *Branding itu Dipraktikin*. Gramedia Widiasarana Indonesia.
- WHO (accessed Oktober 23, 2020). World report on vision. <https://www.who.int/publications-detail-redirect/world-report-on-vision>.

- 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.
- 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.