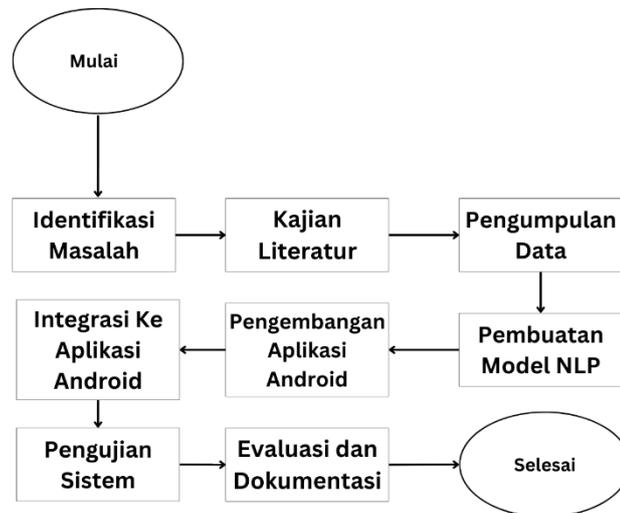


## BAB III

### TAHAPAN PELAKSANAAN

#### 3.1 Langkah-langkah pelaksanaan

Bagian ini akan menjelaskan tahapan-tahapan pelaksanaan tugas akhir, dengan pengantar yang bertujuan memberikan gambaran umum sebelum memasuki pembahasan lebih rinci di *sub-bab* berikutnya.



Gambar 3. 1 Langkah Pelaksanaan

Berikut ini adalah langkah-langkah pelaksanaan yang penulis lakukan dalam penelitian ini:

(1) Identifikasi Masalah

Mengidentifikasi kebutuhan lansia untuk memiliki aplikasi yang dapat menjawab pertanyaan seputar kesehatan lansia dan menganalisis kebutuhan integrasi, seperti model NLP yang mampu memprediksi label, merupakan langkah penting dalam pengembangan aplikasi. Model ini nantinya akan diintegrasikan ke dalam aplikasi untuk memberikan solusi yang tepat guna dan mudah diakses oleh pengguna lansia.

## (2) Studi Literatur

Melakukan studi literatur terkait teknologi pengenalan suara, NLP, dan kebutuhan pengguna mencakup kajian literatur mendalam mengenai NLP, model *IndoBERT*, serta implementasi *machine learning* pada aplikasi *Android*. Selain itu, dilakukan analisis terhadap penggunaan *ONNX* dan bagaimana integrasinya dapat diterapkan dalam aplikasi *Android* untuk mendukung fungsionalitas yang optimal.

## (3) Pengumpulan Data

Pengumpulan data dilakukan dengan merujuk pada berbagai artikel dan jurnal yang membahas keluhan umum pada lansia, seperti yang dipublikasikan oleh RSC Carolus, Golang Tang, dan Sabda Artikel. Data ini dikategorikan

- berdasarkan keluhan-keluhan kesehatan yang sering dialami oleh lansia, meliputi pusing, nyeri sendi, gangguan tidur, nafsu makan berkurang, kulit kering, dan rambut rontok.

Dataset yang digunakan dalam penelitian ini dikategorikan berdasarkan beberapa klasifikasi utama keluhan lansia, meliputi pusing, nyeri sendi, gangguan tidur, nafsu makan berkurang, kulit kering, dan rambut rontok. Untuk kategori pusing, dataset mencakup penyebab pusing, cara meredakan pusing, dan langkah pencegahannya. Pada kategori nyeri sendi, dataset berisi penyebab nyeri, cara mengurangi nyeri, langkah pencegahan, dan aktivitas yang aman dilakukan saat nyeri. Sementara itu, untuk gangguan tidur, dataset memuat penyebab gangguan tidur, cara tidur nyenyak, serta pola tidur yang sehat. Keluhan lainnya seperti nafsu makan berkurang mencakup penyebab penurunan nafsu makan, cara meningkatkannya, dan pola makan sehat. Pada kategori kulit kering, dataset berfokus pada cara merawat kulit kering serta kebiasaan menjaga kesehatan kulit. Sedangkan untuk rambut rontok, dataset memuat penyebab rambut rontok, cara merawat rambut, menjaga kesehatan rambut, aktivitas yang aman dilakukan saat rambut rontok, penggunaan obat, dan langkah pencegahannya.

Dataset ini dibagi menjadi tiga bagian, yaitu training dataset (70%) untuk melatih model, validation dataset (20%) untuk mengevaluasi performa selama pelatihan, dan test dataset (10%) untuk mengukur kemampuan generalisasi model. Pembagian dataset ini dilakukan untuk memastikan model dapat mengolah data dengan baik dan memberikan hasil yang akurat. Referensi keluhan yang digunakan memastikan data relevan dengan masalah yang sering dihadapi oleh lansia, sehingga model NLP yang dikembangkan memiliki tingkat akurasi dan relevansi yang tinggi.

(4) Pembuatan Model NLP

Membuat aplikasi *Android* dilakukan dengan menerapkan rancangan yang telah dibuat, mencakup beberapa tahapan penting. Pertama, preprocessing data dilakukan dengan membersihkan teks, seperti menghapus simbol dan melakukan normalisasi teks, serta menerapkan tokenisasi menggunakan *tokenizer IndoBERT*. Kedua, pelatihan model dilakukan dengan fine-tuning model *IndoBERT* menggunakan dataset yang telah diproses. Selanjutnya, evaluasi model dilakukan untuk mengukur akurasi, *precision*, *recall*, dan *f1-score* guna memastikan performa model. Terakhir, model yang telah dilatih diekspor ke dalam format *ONNX* untuk mempermudah integrasi ke aplikasi *Android*.

(5) Pengembangan Aplikasi *Android*

Pengembangan aplikasi *Android* melibatkan perancangan desain antarmuka (UI) yang mencakup fitur penjadwalan serta chat berbasis model NLP. Selain itu, dilakukan implementasi fitur dasar seperti penjadwalan dengan menggunakan *Room Database* untuk menyimpan data jadwal secara lokal. Aplikasi ini juga dilengkapi dengan fitur perintah suara, yang diimplementasikan menggunakan *SpeechRecognizer* dan *regex* untuk memungkinkan pembuatan jadwal berbasis suara dengan cara yang lebih interaktif dan mudah digunakan.

(6) Integrasi Model ke Aplikasi *Android*

Integrasi model ke aplikasi *Android* dilakukan dengan memuat model *ONNX* menggunakan *ONNX Runtime*. Model ini digunakan untuk melakukan prediksi label berdasarkan input teks maupun suara yang diberikan oleh pengguna. Setelah label berhasil diprediksi, sistem akan memilih jawaban yang sesuai dari *answer\_bank.json*, sehingga memberikan respons yang relevan dan akurat kepada pengguna.

(7) Pengujian Sistem

Pengujian sistem dilakukan melalui beberapa tahapan untuk memastikan kualitas dan fungsi aplikasi. Pengujian *Black Box* dilakukan untuk menilai apakah aplikasi telah berfungsi sebagaimana dirancang, terutama dalam hal prediksi jawaban dan fitur pengelolaan jadwal.. *White box* testing dilakukan untuk memeriksa logika program dan memastikan integrasi model *ONNX* dengan aplikasi berjalan lancar. Selain itu, evaluasi performa aplikasi dilakukan untuk menilai kemampuan aplikasi dalam menghasilkan jawaban yang akurat berdasarkan prediksi model.

(8) Evaluasi dan Dokumentasi

Mengevaluasi hasil sistem berdasarkan pengujian dan membuat dokumentasi akhir.

### 3.2 Metode Pengujian

Pada tahapan ini, pengujian dilakukan untuk memastikan bahwa aplikasi asisten digital pintar berbasis *Android* ini berfungsi secara optimal, stabil, dan dapat memenuhi kebutuhan pengguna, terutama pengguna lansia. Pengujian yang digunakan adalah *White Box* Testing dan *Black Box* Testing. Setiap metode pengujian memiliki fokus, pendekatan, dan tujuan tertentu, yang bersama-sama dapat memberikan hasil pengujian menyeluruh dari sisi internal dan eksternal aplikasi:

#### 3.2.1 *Black box testing*

*Black Box Testing*, atau yang sering disebut sebagai *Behavioral Testing*, adalah metode pengujian perangkat lunak yang berfokus pada fungsionalitas

aplikasi tanpa memperhatikan struktur internal atau kode yang digunakan. Pengujian ini dilakukan "dari luar," artinya penguji hanya melihat interaksi antara input yang diberikan dan output yang dihasilkan oleh sistem, tanpa mengetahui bagaimana aplikasi mencapai hasil tersebut.

Pada penelitian ini, *Black Box Testing* dilakukan untuk menguji fungsionalitas utama aplikasi yang dihadapi oleh pengguna, terutama terkait pengenalan suara, pemrosesan bahasa alami, dan kemudahan interaksi untuk pengguna lansia. Berikut adalah teknik dan tahapan yang diterapkan dalam *Black Box Testing*:

Berikut adalah proses pengujian *black box testing*:

Tabel 3. 1 Pengujian *Black box testing*

No	Fitur	Keterangan	Hasil Sistem
1	Pengujian Fungsionalitas	Menguji setiap fitur utama aplikasi seperti pemrosesan input suara, respons NLP, dan tampilan antarmuka untuk memastikan sistem bekerja sesuai spesifikasi. Skenario penggunaan umum, seperti perintah sederhana, diuji.	Sistem dapat merespons dan memberikan keluaran sesuai harapan pengguna.
2	Pengujian Kegunaan	Melibatkan pengguna lansia untuk menilai apakah antarmuka dan interaksi aplikasi sederhana, mudah dipahami, serta inklusif untuk pengguna dengan keterbatasan fisik atau teknis.	Aplikasi nyaman dan intuitif digunakan oleh pengguna lansia.
3	Pengujian Ketahanan	Menguji aplikasi dengan masukan tidak terduga, seperti perintah suara yang tidak dikenal atau dengan variasi dialek, untuk memastikan aplikasi tetap merespons dengan benar atau memberikan pesan kesalahan yang sesuai.	Sistem dapat menangani masukan ekstrim tanpa kegagalan atau memberikan pesan kesalahan tepat.

4	Pengujian Kinerja	Menilai respons dan kecepatan aplikasi dalam menerima perintah suara atau pemrosesan bahasa alami, serta mengukur konsumsi daya dan penggunaan memori pada perangkat dengan spesifikasi berbeda.	Aplikasi berjalan lancar dan responsif di berbagai perangkat <i>Android</i> .
---	-------------------	--	---

### 3.2.2 *White Box Testing*

*White Box Testing*, juga dikenal sebagai *Structural Testing* atau *Glass Box Testing*, adalah metode pengujian perangkat lunak yang berfokus pada pemeriksaan internal dari kode dan logika program. Dalam pengujian ini, penguji memiliki akses langsung ke kode sumber aplikasi dan memahami alur logika yang digunakan. Dengan kata lain, pengujian ini dilakukan "dari dalam," memungkinkan penguji melihat bagaimana kode berfungsi pada setiap kondisi:

Tabel 3. 2 *White Box Testing*

No	Algoritma	Deskripsi
1	Pengujian Jalur (Path Testing)	Menguji jalur logis dalam algoritma pengenalan suara dan NLP untuk memastikan semua kondisi percabangan berjalan.
2	Pengujian Kondisi (Condition Testing)	Menguji struktur percabangan seperti if-else dan switch-case dalam algoritma untuk memastikan keluaran logika tepat.
3	Pengujian Aliran Data (Data Flow Testing)	Menguji aliran data dalam algoritma, seperti inialisasi, penggunaan, dan penyimpanan variabel, agar data konsisten.