

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Rancang Bangun

Rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tertentu, sehingga rancang bangun berarti menganalisa untuk menciptakan suatu sistem atau memperbaiki sistem yang sudah ada pada suatu instansi (Maulani & Nursolihah, 2022). Rancang bangun juga merupakan serangkaian prosedur untuk menerjemahkan hasil analisis ke dalam bahasa pemrograman dan menciptakan sistem baru atau memperbaiki sistem yang telah ada (Talía et al., 2024). Berdasarkan kedua pengertian tersebut, rancang bangun dalam penelitian ini dimaknai sebagai proses membangun aplikasi baru yang belum tersedia, yaitu aplikasi *mobile* berbasis Android untuk memfasilitasi komunikasi antara penyandang tunanetra dan tunarungu.

Tujuan perancangan adalah untuk memenuhi spesifikasi fungsional, memenuhi batasan-batasan media target implementasi dan target sistem komputer, serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan tenaga ahli lain yang terlibat (Nursakti & Parwati, 2022). Kegiatan rancang bangun diawali dengan tahap pengumpulan data yang dilakukan melalui observasi, wawancara, dan studi

pustaka, kemudian dilanjutkan dengan tahap perancangan dan pengembangan sistem serta tahap pengujian untuk menemukan dan mengidentifikasi sebanyak mungkin permasalahan yang ada di dalam sistem (Rohman & Kusuma, 2025). Tahapan tersebut menjadi acuan peneliti dalam menyusun alur kerja rancang bangun aplikasi, mulai dari identifikasi kebutuhan pengguna tunanetra dan tunarungu hingga pengujian fungsional aplikasi.

Dalam penerapannya pada perangkat bergerak, proses rancang bangun mengacu pada *Software Development Life Cycle* (SDLC) dengan model *waterfall*, yaitu model yang berfokus pada pendekatan sistematis yang berurutan dengan pemodelan menggunakan *Unified Modelling Language* (UML), serta memilih platform Android karena merupakan sistem operasi ponsel pintar terpopuler di Indonesia dengan pangsa pasar mencapai 90,67% sehingga berpotensi menjangkau masyarakat yang lebih luas (Laia & Muliono, 2024). Dengan demikian, rancang bangun pada penelitian ini dilaksanakan secara bertahap dan terstruktur, dimulai dari analisis kebutuhan, perancangan model sistem, pengodean program, hingga pengujian aplikasi. Pendekatan ini dipilih agar aplikasi yang dihasilkan benar-benar sesuai dengan kebutuhan komunikasi dua arah antara penyandang tunanetra dan tunarungu

2. Aplikasi

Aplikasi *mobile* atau yang sering disingkat dengan istilah *mobile apps* adalah aplikasi dari sebuah perangkat lunak yang dalam

pengoperasiannya dapat dijalankan di perangkat *mobile* dan memiliki sistem operasi yang mendukung perangkat lunak secara *standalone* (Waruwu et al., 2023). Aplikasi *mobile* adalah perangkat lunak yang dirancang untuk berjalan pada perangkat bergerak seperti *smartphone* dan tablet, yang biasanya diunduh dan diinstal dari toko aplikasi seperti Google *Play Store* untuk Android atau *App Store* untuk iOS, serta dapat dikategorikan ke dalam berbagai jenis berdasarkan fungsinya (Apridonat et al., 2024). Berdasarkan kedua pengertian tersebut, aplikasi dalam penelitian ini dimaknai sebagai perangkat lunak yang dirancang untuk dijalankan pada perangkat bergerak berbasis Android dan didistribusikan kepada pengguna melalui aplikasi.

Pengembangan teknologi terhadap tren aplikasi *mobile* telah menjadi bagian penting dalam kehidupan sehari-hari (Fidela et al., 2023). Aplikasi *mobile* tidak hanya digunakan sebagai sarana hiburan dan komunikasi, tetapi juga berperan dalam peningkatan efisiensi di berbagai bidang seperti pendidikan, kesehatan, bisnis, dan industri (Alda et al., 2025). Kedudukan aplikasi sebagai sarana komunikasi inilah yang menjadi dasar peneliti memilih bentuk aplikasi sebagai media bantu komunikasi bagi penyandang tunanetra dan tunarungu.

Dengan demikian, aplikasi yang dikembangkan dalam penelitian ini berbentuk aplikasi *mobile* berbasis Android yang memuat fungsi *Speech-To-Text* dan *Text-To-Speech*. Aplikasi tersebut dirancang agar dapat dipasang langsung pada *smartphone* pengguna sehingga mudah dibawa dan

digunakan kapan saja. Melalui bentuk aplikasi ini, proses komunikasi antara penyandang tunanetra dan tunarungu diharapkan dapat berlangsung dua arah tanpa memerlukan perantara orang lain.

3. Android

Android merupakan sistem operasi yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan tablet, dengan fungsi sebagai jembatan antara perangkat (*device*) dan penggunanya dalam pemanfaatan berbagai aplikasi, sehingga *user* dapat berinteraksi dengan perangkatnya sendiri dan menjalankan aplikasi-aplikasi yang memudahkannya dalam melaksanakan kegiatan yang berhubungan dengan dunia digital (Rahma et al., 2021). Android ialah sistem operasi *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi, serta memberikan pengembang platform terbuka untuk mengembangkan aplikasi (Hartono & Fauzi, 2021). Berdasarkan kedua pengertian tersebut, Android dalam penelitian ini dipahami sebagai sistem operasi bergerak berbasis Linux yang bersifat terbuka sehingga memungkinkan peneliti membangun sendiri aplikasi sesuai kebutuhan pengguna.

Di tengah kemajuan teknologi informasi, sistem operasi berperan krusial dalam menjalankan berbagai aplikasi dan perangkat lunak yang digunakan dalam kegiatan sehari-hari, dan keberadaan sistem operasi *open source* telah menawarkan alternatif yang lebih fleksibel, terbuka, serta terjangkau bagi pengembang perangkat lunak (Firdausy et al., 2025). Pada masa modern dengan perkembangan teknologi yang pesat seperti sekarang,

hampir setiap orang memiliki perangkat *mobile* sekurang-kurangnya satu perangkat Android, dan dengan memanfaatkan teknologi tersebut berbagai hal dapat dilakukan hanya dengan perangkat Android, seperti berbelanja maupun berkomunikasi dengan seseorang meskipun jaraknya cukup jauh (Hastedi & Sejati, 2024). Kondisi tersebut menunjukkan bahwa Android merupakan platform yang paling mudah dijangkau oleh calon pengguna aplikasi yang dikembangkan dalam penelitian ini.

Berdasarkan uraian di atas, peneliti memilih Android sebagai platform pengembangan aplikasi karena sifatnya yang terbuka dan penyebarannya yang luas di masyarakat. Sifat terbuka tersebut memungkinkan peneliti mengakses pustaka bawaan sistem, khususnya *SpeechRecognizer* dan *Text-To-Speech*, tanpa memerlukan lisensi berbayar. Dengan demikian, aplikasi yang dibangun dapat langsung dipasang pada *smartphone* penyandang tunanetra maupun tunarungu tanpa memerlukan perangkat khusus tambahan.

4. Tunanetra

Secara kebahasaan istilah "tunanetra" tercipta dari dua kata, yaitu "tuna" dan "netra"; dalam Kamus Bahasa Indonesia "tuna" berarti tidak memiliki, tidak punya, atau mengalami luka dan kerusakan, sedangkan "netra" merujuk pada kemampuan penglihatan, sehingga tunanetra dapat diartikan sebagai buta meskipun tidak semua orang yang mengalami kebutaan sepenuhnya tidak dapat melihat atau hidup dalam kegelapan (Lubis et al., 2024). Tunanetra adalah istilah umum yang digunakan untuk

menyebut seseorang yang mengalami gangguan atau hambatan pada indra penglihatan, sehingga untuk mengenali keberadaan suatu objek penyandang tunanetra akan lebih bergantung pada indra lain seperti pendengaran, penciuman, dan peraba (Ohny et al., 2025). Berdasarkan kedua penjelasan tersebut, tunanetra dalam penelitian ini dipahami sebagai kondisi terganggunya fungsi indra penglihatan yang menyebabkan seseorang mengandalkan indra pendengaran sebagai saluran utama penerimaan informasi.

Berdasarkan data dari Kementerian Sosial tahun 2022, jumlah penyandang tunanetra total sebanyak 11.368 dan *low vision* sebanyak 6.818 dengan total penyandang tunanetra sebanyak 18.186 di Indonesia (Izzah & Supriyadi, 2023). Meskipun menghadapi tantangan visual yang kompleks, anak tunanetra *low vision* menunjukkan profil kemampuan yang unik dengan potensi adaptasi yang signifikan dalam aspek intelektual, fungsional, dan artistik (Putri et al., 2024). Data dan temuan tersebut menunjukkan bahwa jumlah penyandang tunanetra di Indonesia cukup besar dan keterbatasan penglihatan mereka bukan merupakan hambatan kemampuan, melainkan hambatan akses terhadap informasi.

Berdasarkan uraian di atas, peneliti menempatkan penyandang tunanetra sebagai salah satu kelompok pengguna utama aplikasi yang dikembangkan dalam penelitian ini. Karena informasi paling mudah diterima melalui pendengaran, aplikasi dirancang menyampaikan pesan dari mitra bicara dalam bentuk suara melalui fitur *Text-To-Speech*, sekaligus

menerima masukan dari penyandang tunanetra melalui suara dengan fitur *Speech-To-Text*. Dengan rancangan tersebut, penyandang tunanetra tidak perlu membaca teks pada layar untuk dapat berkomunikasi dengan penyandang tunarungu.

5. Tunarungu

Secara kebahasaan, istilah tunarungu terbentuk dari kata "tuna" dan "rungu", di mana tuna berarti kurang dan runggu berarti pendengaran (Perdana et al., 2022). Tunarungu merupakan kondisi ketika individu memiliki gangguan pendengaran, baik sebagian maupun total, dan orang dengan gangguan pendengaran umumnya juga memiliki keterbatasan dalam berkomunikasi secara lisan. Ketunarunguan sendiri dapat diklasifikasikan berdasarkan empat hal, yaitu tingkat kehilangan pendengaran, saat terjadinya ketunarunguan, letak gangguan pendengaran secara anatomis, serta etiologinya (Syalviana et al., 2022). Dengan demikian, tunarungu tidak dapat dipandang sebagai satu kondisi tunggal, sehingga solusi komunikasi yang dirancang untuk mereka harus bertumpu pada kanal visual yang tetap dapat diakses oleh seluruh tingkat kehilangan pendengaran.

Berdasarkan data Badan Pusat Statistik, terdapat lebih dari 268.074 penduduk Indonesia yang mengalami tuli kongenital, dan jumlah ini meningkat sekitar 60.000 kasus dalam satu dekade terakhir, sementara rendahnya pemahaman masyarakat terhadap Bahasa Isyarat Indonesia (BISINDO) semakin memperparah kesenjangan komunikasi antara penyandang tunarungu dengan lingkungan sekitarnya (Irawati & Nurdin,

2025). Hambatan komunikasi tersebut disebabkan oleh kekurangan dalam pendengaran dan berbicara, yang memengaruhi pemahaman terhadap pesan yang disampaikan (Sari et al., 2024). Kondisi ini menunjukkan bahwa hambatan utama penyandang tunarungu bukan terletak pada kemampuan berpikir, melainkan pada ketersediaan media yang mampu mengubah pesan suara menjadi bentuk yang dapat mereka lihat.

Dalam penelitian ini, penyandang tunarungu diposisikan sebagai kelompok pengguna kedua yang berkomunikasi melalui kanal teks. Fitur *Speech-To-Text* pada aplikasi berfungsi menampilkan ucapan lawan bicara ke layar dalam bentuk teks, sehingga pengguna tunarungu tetap dapat menangkap isi percakapan tanpa harus mendengar. Sebaliknya, teks yang ditulis pengguna tunarungu akan dibacakan oleh fitur *Text-To-Speech* agar dapat didengar oleh pengguna tunanetra. Dengan pola dua arah ini, komunikasi antara penyandang tunanetra dan tunarungu dapat berlangsung tanpa memerlukan penguasaan bahasa isyarat oleh kedua pihak maupun kehadiran penerjemah.

6. *Speech-To-Text*

Speech to text yang merupakan bagian dari speech recognition adalah bentuk teknologi yang dapat mengenali ucapan manusia sebagai user yang kemudian dikonversikan ke dalam bentuk teks (Karisma et al., 2022). Dengan pengertian tersebut, *Speech-To-Text* pada penelitian ini diposisikan sebagai jembatan yang mengubah ucapan lisan menjadi tulisan sehingga pesan suara dapat dibaca oleh pihak yang tidak mampu mendengarnya.

Fungsi inilah yang menjadi dasar perancangan fitur masukan suara pada aplikasi yang dibangun.

Secara teknis, *Automatic Speech Recognition* (ASR) merupakan teknologi yang memiliki kemampuan dalam mengenali suara secara otomatis dan merubahnya menjadi tulisan (*speech to text*), sehingga penggunaanya dapat mengkonversi suara yang dikeluarkan menjadi sebuah teks yang ditampilkan ke dalam layar (Aziz et al., 2025). Dari penjelasan itu dapat dipahami bahwa proses *Speech-To-Text* berjalan dalam tiga tahap, yaitu penangkapan suara melalui mikrofon, pengenalan pola suara oleh mesin pengenalan ucapan, dan penyajian hasil pengenalan dalam bentuk teks di layar perangkat. Pada aplikasi Android, ketiga tahap tersebut dapat dijalankan secara real time sehingga teks muncul hampir bersamaan dengan saat pengguna selesai berbicara.

Layanan pengenalan suara menerjemahkan sinyal ucapan menjadi teks menggunakan perangkat input ucapan seperti mikrofon ponsel, dan saat ini sistem tersebut digunakan untuk menggantikan peran input keyboard (Saputra et al., 2024). Kemampuan menggantikan papan ketik ini sangat penting bagi penyandang tunanetra, karena mereka tidak perlu lagi mengetik pada layar sentuh yang sulit dijangkau tanpa penglihatan. Oleh karena itu, dalam aplikasi yang dirancang pada penelitian ini *Speech-To-*

Text dipakai untuk menerima ucapan pengguna tunanetra dan menampilkannya sebagai teks yang dapat dibaca oleh pengguna tunarungu.

7. *Text-To-Speech*

Text to Speech (TTS) adalah teknologi yang memungkinkan komputer untuk mengubah teks tertulis menjadi ucapan, dan terdapat dua prinsip yang mendasarinya yaitu merubah teks ke fonem dan merubah fonem ke ucapan (Arrizqi et al., 2021). Berdasarkan pengertian tersebut, *Text-To-Speech* dapat dipahami sebagai kebalikan dari *Speech-To-Text*, yaitu teks yang tersimpan di dalam sistem dibunyikan kembali menjadi suara yang dapat didengar. Pada penelitian ini, kemampuan tersebut dibutuhkan agar pesan yang dituliskan oleh pengguna tunarungu dapat sampai kepada pengguna tunanetra dalam bentuk suara.

Penerapan TTS pada perangkat bergerak telah banyak dilakukan, salah satunya adalah penelitian yang bertujuan untuk merancang dan membangun sebuah aplikasi e-book reader yang dapat mengonversi teks menjadi suara bagi penyandang disabilitas (Ramadhan & Suhirman, 2024). Penelitian tersebut menunjukkan bahwa fitur TTS pada aplikasi Android efektif meningkatkan aksesibilitas bacaan bagi pengguna yang tidak dapat membaca secara visual. Hal ini memperkuat pilihan penulis untuk menempatkan TTS sebagai keluaran utama aplikasi yang dibangun, bukan sekadar fitur tambahan.

Selain untuk aksesibilitas, analisis statistik deskriptif juga membuktikan teknologi TTS secara signifikan meningkatkan kemampuan

membaca pada anak disleksia (Fatmawati et al., 2024). Temuan itu menunjukkan bahwa keluaran suara dari teks bermanfaat bukan hanya bagi tunanetra, tetapi juga bagi siapa pun yang mengalami hambatan dalam membaca tulisan. Oleh karena itu, pada aplikasi yang dirancang dalam penelitian ini *Text-To-Speech* berfungsi membacakan teks masukan dari pengguna tunarungu sehingga percakapan dua arah antara tunanetra dan tunarungu dapat berlangsung tanpa perantara manusia.

8. *Android SpeechRecognizer*

Android menyediakan kemampuan pengenalan ucapan sebagai fitur bawaan sistem operasinya, di mana salah satu fitur smartphone adalah *speech recognition* atau *speech to text* yang digunakan untuk memudahkan pengguna dalam memanfaatkan suara untuk menjalankan aplikasi smartphone tanpa menggunakan keyboard (Yuliadi et al., 2021). Fitur inilah yang di dalam pemrograman Android diwadahi oleh kelas *SpeechRecognizer* beserta *RecognizerIntent* pada paket android *speech*. Dengan demikian pengembang tidak perlu membangun mesin pengenalan suara sendiri, melainkan cukup memanggil layanan pengenalan ucapan yang sudah tersedia pada perangkat.

Secara teknis, perekaman suara dapat dilakukan dengan memanggil fungsi *startRecording()* dan memanggil fungsi *stopRecording()* untuk menghentikan rekaman, dan setelah rekaman berhenti maka Google *Speech API* akan memberikan hasil *speech to text* secara otomatis (Sujjada et al., 2024). Alur tersebut menggambarkan pola kerja *SpeechRecognizer* pada

Android, yaitu aplikasi memulai sesi perekaman, menghentikannya, lalu menerima hasil pengenalan melalui *callback* dalam bentuk daftar teks. Karena layanan ini berjalan di tingkat sistem, aplikasi hanya perlu meminta izin akses mikrofon dan menangani hasil yang dikembalikan.

Pada tahap implementasi, setelah suara direkam, sistem akan menggunakan teknologi *speech to-text* melalui Google *Speech Recognition* API untuk mengkonversi suara pengguna menjadi teks secara otomatis (Evanti & Willay, 2025). Hasil konversi tersebut kemudian dapat langsung diproses lebih lanjut oleh aplikasi, misalnya ditampilkan, dibandingkan, atau dikirim ke penyimpanan data. Dalam penelitian ini, *SpeechRecognizer* dimanfaatkan sebagai pintu masukan bagi pengguna tunanetra, sehingga ucapannya dapat diubah menjadi teks yang dibaca oleh pengguna tunarungu.

9. Android Text-To-Speech

Android Text-To-Speech adalah mesin sintesis ucapan yang telah disediakan langsung oleh sistem operasi Android, sehingga aplikasi tidak perlu membangun mesin suara sendiri. Suhendri et al. (2026) menerangkan bahwa lapisan keempat adalah lapisan sistem operasi Android yang menyediakan *Android Text-To-Speech Engine* untuk melakukan sintesis vokal aktual dari teks yang dikirimkan oleh lapisan *native*. Dengan posisi tersebut, setiap teks yang dihasilkan aplikasi cukup diteruskan ke mesin

bawaan Android untuk diubah menjadi keluaran suara melalui *speaker* perangkat.

Pemanfaatan mesin bawaan ini memungkinkan pengembang membangun aplikasi konversi teks ke suara secara langsung di dalam lingkungan pengembangan resmi Android. Bernadisman & Lesdiansyah (2023) menjelaskan bahwa penelitian mereka telah berhasil membuat sebuah aplikasi *Text to Speech* Berbasis *Android* Menggunakan *XML Java* yang dapat membantu orang-orang yang memiliki keterbatasan berbicara dan juga dapat membantu dalam belajar pengucapan bahasa asing umumnya bahasa inggris dan tigabahasa lainnya. Temuan tersebut menunjukkan bahwa *Android Text-To-Speech* bukan sekadar fitur tambahan, melainkan komponen inti yang dapat menggantikan peran suara bagi pengguna yang tidak dapat berbicara.

Kualitas keluaran suara menjadi faktor penentu keberhasilan penggunaan teknologi ini sebagai alat bantu. Sarif & AR (2024) menyatakan bahwa TTS berbasis kecerdasan buatan dapat menghasilkan suara yang jelas dan akurat sehingga pendengar memahami kata dengan benar dan mampu menirukan pelafalan yang tepat. Pada penelitian ini, *Android Text-To-Speech* dimanfaatkan untuk membacakan teks hasil pengetikan penyandang tunarungu agar dapat didengar oleh penyandang tunanetra, sehingga arah

komunikasi kedua pihak menjadi lengkap dan saling melengkapi dengan fitur *Speech-To-Text*.

10. Kotlin

Kotlin merupakan bahasa pemrograman yang baru dikembangkan oleh JetBrains dan diakui sebagai alternatif yang lebih mudah dipelajari daripada Java untuk pengembangan aplikasi Android, serta dapat pula digunakan untuk membuat aplikasi non-*smartphone* atau aplikasi *native* (Siregar et al., 2024). Dengan karakteristik tersebut, Kotlin dipilih sebagai bahasa utama dalam pembangunan aplikasi komunikasi tunanetra–tunarungu pada penelitian ini karena sintaksisnya ringkas namun tetap kompatibel penuh dengan seluruh pustaka Android yang dibutuhkan, seperti *SpeechRecognizer* dan *TextToSpeech*.

Keunggulan Kotlin juga terlihat dari sisi keringkasan penulisan kode. Pada pengujian pengembangan aplikasi Android berbasis iterasi, Kotlin memerlukan 157 baris kode (LOC), lebih ringkas dibandingkan Java yang memerlukan 169 baris (Sanjaya & Susilo, 2024). Keringkasan tersebut menjadi pertimbangan dalam penelitian ini karena jumlah baris kode yang lebih sedikit memudahkan penulis dalam memelihara dan mengembangkan modul *Speech-To-Text* maupun *Text-To-Speech* pada aplikasi yang dibangun.

Selain faktor keringkasan kode, Kotlin memberikan keuntungan pada sisi konsumsi sumber daya perangkat. Hasil pengujian pemrosesan data bertipe *string* pada aplikasi Android menunjukkan bahwa dalam hal

penggunaan CPU dan memori Kotlin lebih baik dibandingkan Java karena lebih sedikit (Wicaksono & Leong, 2021). Efisiensi CPU dan memori tersebut relevan bagi aplikasi yang dirancang dalam penelitian ini karena proses *Speech-To-Text* dan *Text-To-Speech* berjalan hampir bersamaan pada satu perangkat, sehingga penggunaan Kotlin diharapkan menjaga aplikasi tetap responsif ketika dipakai berkomunikasi secara bergantian oleh penyandang tunanetra dan tunarungu.

11. *Firebase Realtime Database*

Firebase Realtime Database merupakan basis data berbasis *cloud* NoSQL yang menyinkronkan data di semua klien secara *realtime*, dan menyediakan fungsionalitas *offline* (Umasugi et al., 2025). Metode API *Firebase Realtime Database* memproses sinkronisasi secara *realtime* melalui *JavaScript Object Notation* (JSON) yang disediakan (Setyawan, 2024). Dengan karakteristik tersebut, data yang dikirim oleh satu perangkat dapat langsung dibaca oleh perangkat lain tanpa perlu membangun server aplikasi sendiri.

Dalam pengembangan aplikasi *mobile*, *Firebase Realtime Database* umumnya dipakai sebagai media penyimpanan utama yang menggantikan basis data lokal. Sebagai contoh, sebuah aplikasi manajemen data siswa dan nilai dibangun menggunakan *framework* Flutter dengan *Firebase Realtime Database* sebagai media penyimpanan berbasis *cloud* (Setiawan & Rusliyawati, 2025). Pola pemanfaatan seperti ini menunjukkan bahwa

Firebase Realtime Database cocok untuk aplikasi yang menuntut pertukaran data cepat, termasuk aplikasi komunikasi dua arah.

Pilihan basis data pada aplikasi Android sendiri bervariasi, sebab pada penelitian-penelitian pengembangan aplikasi Android sebelumnya umumnya digunakan basis data terpusat seperti MySQL atau *Firebase* (Mulyani et al., 2025). Pada penelitian ini, *Firebase Realtime Database* dipilih karena hasil percakapan berupa teks hasil *Speech-To-Text* perlu dikirim dan dibaca secara langsung oleh pengguna lain pada perangkat yang berbeda. Selain itu, sifatnya yang tidak memerlukan pengelolaan server mandiri memudahkan pengembangan aplikasi *mobile* skala penelitian.

12. *Flowchart*

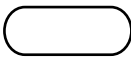
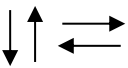
Flowchart adalah menggambarkan secara grafik dari langkah-langkah dan urutan prosedur suatu program; bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika (Nadiyah & Arifin, 2023). *Flowchart*, yang juga dikenal sebagai bagan alur, adalah diagram yang berisi detail dan motivasi yang diperlukan untuk menjalankan prosedur program tertentu (Nadiyah & Arifin, 2023). Dengan demikian, *flowchart* berfungsi sebagai gambaran logika kerja sistem sebelum program benar-benar dituliskan dalam bahasa pemrograman.

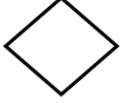
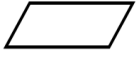
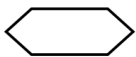
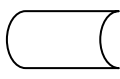
Dalam penggambarannya, diagram alir dapat menjelaskan prosedur yang perlu diikuti untuk menyelesaikan suatu masalah, serta dapat menggunakan simbol-simbol yang ditetapkan oleh *American National*

Standards Institute (ANSI) untuk menggambarkan secara visual tahapan dan alur dalam suatu program (Burhanuddin & Sukirman, 2024). Simbol-simbol yang umum digunakan meliputi simbol awal/akhir (*terminator*), simbol proses, simbol masukan/keluaran, serta simbol keputusan (*decision*) untuk percabangan kondisi. Penggunaan simbol standar tersebut membuat alur program dapat dibaca dan diperiksa kembali oleh pihak lain, termasuk dosen pembimbing maupun penguji.

Pada penelitian ini, *flowchart* digunakan untuk menggambarkan alur kerja aplikasi komunikasi antara penyandang tunanetra dan tunarungu, mulai dari pengguna membuka aplikasi, memilih peran, melakukan masukan suara melalui *Speech-To-Text*, hingga keluaran teks maupun suara melalui *Text-To-Speech*. Alur pengambilan keputusan seperti pemeriksaan izin mikrofon dan ketersediaan koneksi juga digambarkan dengan simbol *decision* agar setiap kemungkinan jalur proses terlihat jelas. Dengan demikian, *flowchart* menjadi dasar sebelum perancangan dilanjutkan ke pemodelan berorientasi objek menggunakan UML.

Tabel 2. 1 Simbol Flowchart

No	Simbol	Nama	Keterangan
1		<i>Terminator</i> (Terminal)	Menyatakan awal (Mulai) dan akhir (Selesai) dari suatu program atau proses.
2		<i>Flow Line</i> (Garis Alir)	Menunjukkan arah aliran proses dari satu langkah ke langkah berikutnya.
3		<i>Process</i> (Proses)	Menyatakan proses pengolahan atau perhitungan data.

4		<i>Decision</i> (Keputusan)	Menyatakan percabangan atau pengambilan keputusan berdasarkan kondisi (ya/tidak).
5		<i>Input/Output</i> (Data)	Menyatakan proses masukan (input) atau keluaran (output) data.
6		<i>Preparation</i> (Persiapan)	Menyatakan pemberian nilai awal atau persiapan suatu variabel/penghitung.
7		<i>Connector</i> (Penghubung)	Penghubung bagian-bagian flowchart pada halaman yang sama.
8		<i>Off-Page Connector</i>	Penghubung flowchart yang berlanjut ke halaman yang berbeda.
9		<i>Document</i> (Dokumen)	Menyatakan masukan/keluaran berupa dokumen atau laporan tercetak.
10		<i>Disk / on-line Storage</i>	Menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>

13. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk memodelkan sistem perangkat lunak, yang dikembangkan oleh *Object Management Group* (OMG) pada tahun 1999 (Perwitasari et al., 2024). Dalam penelitian ini UML digunakan sebagai alat pemodelan berorientasi objek untuk menggambarkan kebutuhan dan struktur aplikasi komunikasi antara penyandang tunanetra dan tunarungu sebelum masuk ke tahap penulisan kode program. Diagram UML yang

digunakan dibatasi pada tiga diagram, yaitu *use case diagram*, *activity diagram*, dan *class diagram*, karena ketiganya sudah cukup untuk menggambarkan kebutuhan fungsional, alur aktivitas, dan struktur data aplikasi.

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Unified Modeling Language* (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, merancang, mendokumentasikan, dan menganalisis sistem perangkat lunak. Penggunaan UML membantu pengembang memahami kebutuhan sistem, mempermudah komunikasi, serta menghasilkan rancangan yang lebih terstruktur sebelum proses implementasi dilakukan.

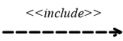
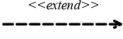
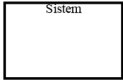
a. *Use Case Diagram*

Use case diagram adalah jenis diagram *Unified Modeling Language* (UML) yang digunakan dalam rekayasa perangkat lunak untuk secara visual merepresentasikan interaksi antara berbagai aktor (pengguna atau sistem eksternal) dan suatu sistem (Rasiban et al., 2024). Pada aplikasi yang dibangun, aktor terdiri atas pengguna tunanetra dan pengguna tunarungu, dengan *use case* seperti memulai percakapan, mengucapkan pesan melalui *Speech-To-Text*, mengirim pesan teks, serta mendengarkan pesan melalui *Text-To-Speech*. Diagram ini membantu memastikan seluruh kebutuhan fungsional aplikasi telah teridentifikasi sesuai peran masing-masing pengguna.

Pada penelitian ini, *Use Case Diagram* melibatkan dua aktor utama, yaitu penyandang tunanetra dan penyandang tunarungu. Penyandang tunanetra menggunakan aplikasi dengan memberikan masukan berupa suara melalui fitur *Speech-To-Text*. Hasil konversi suara menjadi teks kemudian diterima oleh penyandang tunarungu dalam bentuk pesan teks. Sebaliknya, penyandang tunarungu memberikan balasan dengan mengetikkan pesan pada aplikasi, kemudian sistem mengubah teks tersebut menjadi suara menggunakan fitur *Text-To-Speech* agar dapat didengarkan oleh penyandang tunanetra. Melalui *Use Case Diagram*, hubungan antara aktor dan fungsi-fungsi utama sistem dapat digambarkan secara jelas sehingga memudahkan proses analisis kebutuhan dan perancangan sistem.

Tabel 2. 2 Case Diagram

No	Simbol	Nama	Keterangan
1		<i>Actor</i> (Aktor)	Menggambarkan pengguna atau sistem lain yang berinteraksi dengan sistem.
2		<i>Use Case</i>	Menggambarkan fungsionalitas atau layanan yang disediakan sistem kepada aktor.
3		<i>Association</i> (Asosiasi)	Menghubungkan aktor dengan use case yang berinteraksi dengannya.
4		<i>Generalization</i>	Relasi pewarisan, yaitu sebuah elemen merupakan bentuk khusus dari elemen lain.

5		<i>Include</i>	Menunjukkan use case yang selalu memanggil (menyertakan) <i>use case</i> lain.
6		<i>Extend</i>	Menunjukkan use case yang dapat memperluas perilaku use case lain pada kondisi tertentu.
7		<i>System Boundary</i>	Batas ruang lingkup sistem yang memisahkan sistem dari aktor.

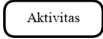
b. *Activity Diagram*

Activity Diagram adalah salah satu elemen penting dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan sisi dinamis dari suatu sistem (Mandansari et al., 2025). Pada penelitian ini *activity diagram* menggambarkan alur aktivitas mulai dari pengguna membuka aplikasi, memilih peran, melakukan masukan suara atau teks, hingga aplikasi menampilkan teks maupun membunyikan suara hasil konversi. Percabangan seperti pemeriksaan izin mikrofon dan keberhasilan pengenalan suara juga digambarkan agar setiap kemungkinan alur aktivitas terlihat jelas.

Pada penelitian ini, *Activity Diagram* digunakan untuk menggambarkan alur komunikasi antara penyandang tunanetra dan penyandang tunarungu melalui aplikasi yang dikembangkan. Diagram ini menunjukkan urutan aktivitas yang dilakukan oleh masing-masing aktor ketika melakukan proses komunikasi menggunakan fitur *Speech-To-Text* dan *Text-To-Speech*.

Proses komunikasi diawali ketika penyandang tunanetra menyampaikan pesan melalui suara. Sistem kemudian mengonversi suara tersebut menjadi teks menggunakan fitur *Speech-To-Text* dan menampilkan hasilnya kepada penyandang tunarungu. Setelah membaca pesan yang diterima, penyandang tunarungu memberikan balasan dengan mengetikkan teks pada aplikasi. Selanjutnya, sistem mengubah teks tersebut menjadi suara menggunakan fitur *Text-To-Speech* sehingga pesan dapat didengarkan oleh penyandang tunanetra. Melalui *Activity Diagram*, alur aktivitas setiap aktor dapat divisualisasikan secara sistematis sehingga mempermudah proses analisis, implementasi, dan pengujian fungsi-fungsi yang terdapat pada aplikasi.

Tabel 2. 3 Activity Diagram

No	Simbol	Nama	Keterangan
1	●	<i>Initial Node</i>	Titik awal dimulainya suatu rangkaian aktivitas.
2		<i>Action / Activity</i>	Aktivitas atau aksi yang dilakukan dalam alur proses.
3		<i>Decision / Merge</i>	Percabangan atau penggabungan alur berdasarkan suatu kondisi.
4		<i>Fork / Join</i>	Pemisahan (<i>fork</i>) atau penggabungan (<i>join</i>) alur aktivitas yang berjalan paralel.
5		<i>Control Flow</i>	Menunjukkan arah aliran dari satu aktivitas ke aktivitas berikutnya.
6	⦿	<i>Final Node</i>	Titik akhir berakhirnya rangkaian aktivitas.

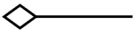
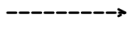
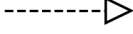
c. Class Diagram

Class Diagram menggambarkan serta deskripsi dari *class*, atribut dan objek serta hubungan satu sama lain (Harlina et al., 2025). Pada aplikasi ini *class diagram* memuat kelas-kelas utama seperti pengguna, sesi percakapan, dan pesan, beserta atribut dan relasi antar kelas yang menjadi dasar penyusunan struktur data pada *Firestore Realtime Database*. Dengan demikian, ketiga diagram tersebut saling melengkapi: *use case diagram* untuk kebutuhan, *activity diagram* untuk alur proses, dan *class diagram* untuk struktur data aplikasi.

Pada penelitian ini, *Class Diagram* digunakan untuk menggambarkan struktur kelas yang membentuk aplikasi komunikasi antara penyandang tunanetra dan penyandang tunarungu. Diagram ini menunjukkan hubungan antarkelas yang berperan dalam pengelolaan data pengguna, proses komunikasi, konversi suara menjadi teks menggunakan fitur *Speech-To-Text*, konversi teks menjadi suara menggunakan fitur *Text-To-Speech*, serta penyimpanan data pada *Firestore Realtime Database*. Setiap kelas memiliki atribut dan metode yang sesuai dengan tanggung jawabnya masing-masing sehingga seluruh fungsi aplikasi dapat dijalankan secara terstruktur. Hubungan antarkelas, seperti *association* maupun bentuk relasi lainnya, digunakan untuk menggambarkan keterkaitan antarobjek dalam mendukung proses komunikasi yang terjadi pada aplikasi. Melalui *Class Diagram*, struktur sistem dapat divisualisasikan secara jelas sehingga memudahkan proses implementasi, pemeliharaan, dan pengembangan perangkat lunak.

Tabel 2. 4 *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Menggambarkan kelas yang terdiri atas nama kelas, atribut, dan operasi/metode.
2		<i>Association</i>	Relasi struktural yang menghubungkan dua kelas.
4		<i>Generalization / Inheritance</i>	Relasi pewarisan; subkelas mewarisi atribut dan operasi dari superkelas.

5		<i>Aggregation</i>	Relasi keseluruhan-bagian yang bersifat lemah (bagian dapat berdiri sendiri).
6		<i>Composition</i>	Relasi keseluruhan-bagian yang bersifat kuat (bagian bergantung pada keseluruhan).
7		<i>Dependency</i>	Ketergantungan; perubahan pada satu kelas dapat memengaruhi kelas lain.
8		<i>Realization</i>	Relasi antara antarmuka (<i>interface</i>) dengan kelas yang mengimplementasikannya.

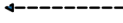
d. *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan interaksi antar objek, secara khusus menjabarkan *behavior* sebuah skenario tunggal, serta menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati beberapa objek tersebut (Dini et al., 2025). Dengan demikian *sequence diagram* melengkapi *use case diagram*, *activity diagram*, dan *class diagram* karena menempatkan setiap pertukaran pesan pada urutan waktu yang jelas. Dalam penelitian ini *sequence diagram* digunakan untuk memodelkan urutan pesan antara pengguna tunanetra dan tunarungu dengan komponen aplikasi, yaitu dari penerimaan suara oleh *Android SpeechRecognizer*, pengiriman teks ke *Firebase Realtime Database*, hingga pembacaan teks balasan oleh *Android Text-To-Speech*.

Pada penelitian ini, *Sequence Diagram* digunakan untuk memodelkan urutan interaksi antara penyandang tunanetra,

penyandang tunarungu, dan sistem selama proses komunikasi berlangsung. Diagram ini menggambarkan pertukaran pesan yang terjadi sejak pengguna memulai komunikasi hingga sistem memberikan respons sesuai dengan fungsi yang dijalankan. Urutan proses diawali ketika penyandang tunanetra memberikan masukan berupa suara kepada aplikasi. Sistem kemudian memproses masukan tersebut menggunakan fitur *Speech-To-Text* sehingga menghasilkan pesan dalam bentuk teks yang dapat dibaca oleh penyandang tunarungu. Selanjutnya, penyandang tunarungu memberikan balasan dengan mengetikkan pesan pada aplikasi. Sistem memproses pesan tersebut menggunakan fitur *Text-To-Speech* sehingga menghasilkan keluaran berupa suara yang dapat didengarkan oleh penyandang tunanetra. Melalui *Sequence Diagram*, urutan pengiriman pesan, respons sistem, serta interaksi antarobjek dapat divisualisasikan secara sistematis sehingga mempermudah proses analisis, implementasi, dan pengujian sistem.

Tabel 2. 5 Sequence Diagram

No	Simbol	Nama	Keterangan
1		<i>Object / Lifeline</i>	Objek beserta garis hidup (<i>lifeline</i>) selama interaksi berlangsung.
2		<i>Activation</i>	Periode saat objek aktif menjalankan suatu proses.
3		<i>Message</i>	Pesan atau pemanggilan metode dari satu objek ke objek lain.
4		<i>Return Message</i>	Pesan balikan (<i>return</i>) sebagai respons atas pesan yang dikirim.
5		<i>Self Message</i>	Pesan yang dikirim objek kepada dirinya sendiri.

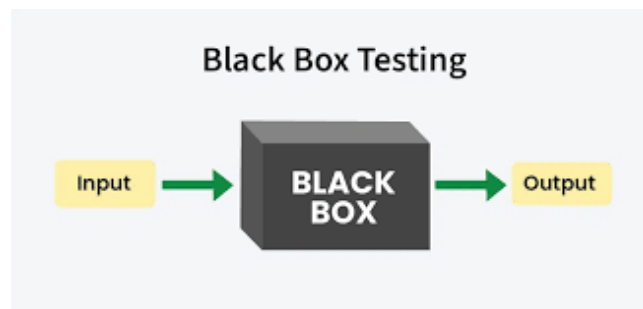
14. *Black Box Testing*

Black box testing adalah pengujian yang hanya menguji bagian luar dari perangkat lunak dan merupakan teknik pengujian yang berfokus pada kebutuhan fungsional perangkat lunak berdasarkan spesifikasi kebutuhan perangkat lunak tersebut (Pratama et al., 2023). *Black box testing* adalah metode pengujian yang berfokus pada spesifikasi fungsional perangkat lunak tanpa memperhatikan struktur internalnya (Zahra et al., 2024). Dengan demikian pengujian ini menilai kesesuaian antara masukan yang diberikan dan keluaran yang dihasilkan sistem, bukan kebenaran baris kode di dalamnya.

Black box testing yang juga disebut sebagai *functional testing* atau *behavioral testing* memandang perangkat lunak seperti "kotak hitam", di

mana penguji hanya memperhatikan *input* dan *output* tanpa mengetahui implementasi internal sistem (Maulida et al., 2025). Metode ini dapat menemukan beberapa kategori kesalahan seperti fungsi yang tidak benar atau tidak ada, kesalahan antarmuka, kesalahan pada struktur data dan akses basis data, kesalahan performansi, serta kesalahan inisialisasi dan terminasi (Andika et al., 2025). Cakupan kesalahan tersebut membuat *black box testing* sesuai digunakan pada tahap akhir pengembangan, yaitu ketika seluruh fitur telah terintegrasi dan siap diuji dari sudut pandang pengguna.

Dalam penelitian ini *black box testing* dipilih sebagai metode pengujian karena aplikasi yang dibangun akan digunakan langsung oleh penyandang tunanetra dan tunarungu, sehingga yang paling penting dinilai adalah kesesuaian fungsi terhadap kebutuhan pengguna. Pengujian dilakukan dengan menyusun sejumlah skenario uji pada setiap fitur utama, yaitu pengubahan suara menjadi teks melalui *Android SpeechRecognizer*, pengiriman dan penerimaan pesan melalui *Firebase Realtime Database*, serta pembacaan teks menjadi suara melalui *Android Text-To-Speech*. Hasil setiap skenario kemudian dibandingkan dengan hasil yang diharapkan untuk menentukan apakah seluruh fungsi aplikasi telah berjalan sesuai spesifikasi sebelum aplikasi digunakan.



Gambar 2. 1 *Black Box*

B. Kajian Empiris

Beberapa penelitian terdahulu telah mengkaji pemanfaatan teknologi *speech-to-text* dan *text-to-speech* sebagai alat bantu komunikasi bagi penyandang disabilitas, dengan fokus, metode, dan capaian yang berbeda-beda.

Pertama, Ibnurhus & Septa (2025) mengembangkan aplikasi "TalkBuddy" berbasis Android sebagai media komunikasi dan pembelajaran inklusif bagi mahasiswa tunarungu di Universitas Pamulang. Penelitian tersebut menggunakan metode *Research and Development* dengan tahapan pengembangan multimedia Luther-Sutopo, yaitu *concept, design, material collecting, assembly, testing, dan distribution*, serta memanfaatkan pustaka Google *Speech-to-Text* dan Google *Text-to-Speech*. Hasil *alpha test* oleh ahli IT, ahli media, dan *reviewer* disabilitas menunjukkan seluruh aspek memperoleh nilai di atas 80 dengan kualifikasi "sangat layak", kecuali aspek *memorability* dari ahli IT yang memperoleh 77,8 dengan kualifikasi "layak". Meskipun demikian, aplikasi tersebut diarahkan pada kebutuhan mahasiswa tunarungu dalam konteks perkuliahan, sehingga belum menyediakan jalur komunikasi timbal balik yang secara khusus melayani penyandang tunanetra.

Kedua, Vedu et al. (2025) mengembangkan alat bantu belajar taktil bagi anak *deafblind* (tunanetra sekaligus tunarungu) yang mengubah gerakan bibir pembicara menjadi teks memakai *3D Convolutional Neural Network* dan *Bidirectional Long Short-Term Memory*, kemudian menerjemahkan teks tersebut menjadi kode Morse yang disampaikan melalui perangkat *wearable*. Hasil evaluasi eksperimental model konversi teks tersebut menunjukkan *word error rate* sebesar 2% dan tingkat akurasi 98%, lebih baik daripada model LipNet yang memiliki *word error rate* 4,8%. Penelitian ini menegaskan bahwa alat bantu yang diusulkan mampu menembus hambatan komunikasi bagi anak dengan hambatan pendengaran dan penglihatan. Penelitian tersebut memiliki kesamaan tujuan dengan penelitian ini, yaitu menjembatani komunikasi dua kelompok disabilitas sensorik, namun berbeda pada medianya karena masih bergantung pada perangkat keras *wearable* berbasis kode Morse, sedangkan penelitian ini memanfaatkan telepon pintar Android yang sudah umum dimiliki pengguna.

Ketiga, Kim et al. (2024) melakukan evaluasi *usability* terhadap aplikasi *mobile* pembantu komunikasi bagi pengguna dengan hambatan pendengaran di Korea, dengan pendekatan campuran yang melibatkan tiga belas partisipan tunarungu berusia 19–34 tahun serta para ahli. Aplikasi yang dievaluasi mengenali suara dari satu pembicara dan mengubahnya menjadi teks yang ditampilkan dengan jeda sekitar dua detik bergantung pada kondisi jaringan (Kim dkk., 2024). Hasil pengukuran menunjukkan skor kepuasan rata-rata 4,38 dari maksimum 7 poin, sedangkan skor *system usability scale* secara

keseluruhan rata-rata 66,73 dengan simpangan baku 17,42 yang berarti berada di antara kategori *acceptable* dan *good*. Temuan tersebut memperlihatkan bahwa akurasi pengenalan suara dan desain antarmuka masih menjadi titik lemah aplikasi *speech-to-text* bagi tunarungu, sehingga penelitian ini menempatkan kesederhanaan antarmuka dan kejelasan umpan balik sebagai pertimbangan utama perancangan.

Keempat, Rafiadly et al. (2023) merancang aplikasi Android "Naviku" yang memanfaatkan teknologi *Text-to-Speech* untuk memberikan informasi navigasi berupa suara agar tunanetra dapat bergerak secara mandiri, dikembangkan dengan prinsip *Agile Software Development* melalui metode *Test Driven Development*. Seluruh fitur yang diuji memperoleh status keberhasilan 100%, sedangkan pengujian dua mesin *Text-to-Speech* menunjukkan *Word Error Rate* Google TTS sebesar 2,86 yang lebih rendah dibandingkan Samsung TTS sebesar 4,06. Namun, pemanfaatan *Text-to-Speech* pada penelitian tersebut terbatas pada penyampaian informasi navigasi, bukan sebagai sarana percakapan antarpengguna.

Kelima, Lavric et al. (2024) menyusun kajian komprehensif mengenai teknologi asistif bagi penyandang tunanetra dengan menyoroti integrasi *Visible Light Communications* dan kecerdasan buatan. Kajian tersebut menyatakan bahwa selama beberapa dekade terakhir suara sintetis terus berkembang dan menjadi hampir hadir di mana-mana pada berbagai aplikasi, mulai dari asisten virtual hingga alat bantu baca pada telepon pintar dan tablet, khususnya bagi individu tunanetra melalui konversi *text-to-speech*. Kajian ini juga mencatat

bahwa interaksi teks dan suara memungkinkan konversi *text-to-speech* maupun sebaliknya berlangsung secara *real-time*. Dengan demikian, kajian tersebut memberi landasan bahwa pemanfaatan *text-to-speech* pada perangkat bergerak merupakan arah pengembangan teknologi asistif yang telah teruji, sekaligus memperkuat pilihan teknologi dalam penelitian ini.

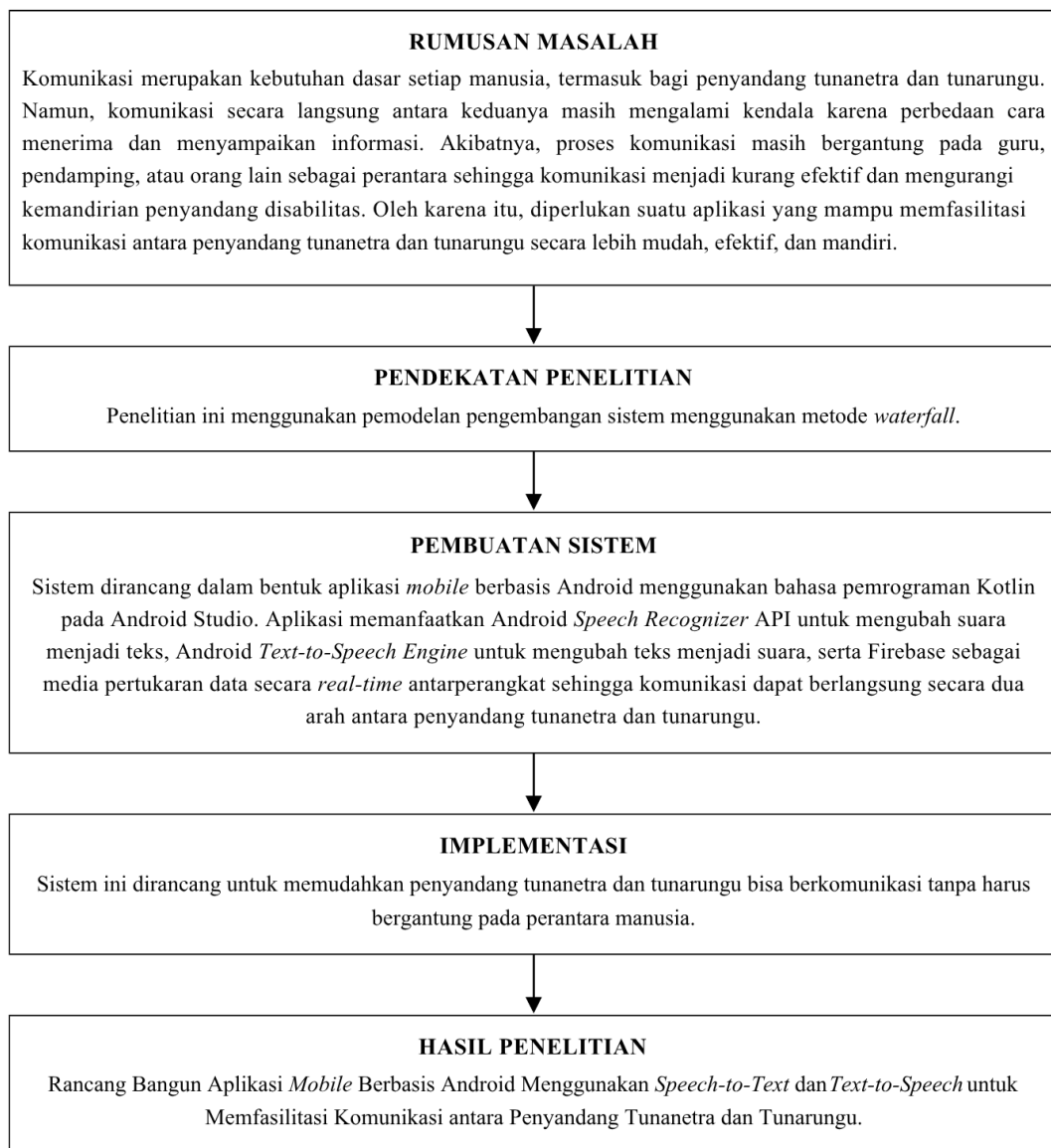
Keenam, Beny et al. (2024) mengembangkan alat komunikasi interaktif "INCOM" bagi tunanetra dan tunarungu pada proses pembelajaran dengan metode ADDIE, yang mencakup tahapan *Analysis*, *Design*, *Development*, dan *Implementation*. Hasil penelitian menunjukkan media tersebut sangat layak digunakan dengan validasi ahli media 83,2% dan ahli materi 80%, sementara uji coba pada mahasiswa disabilitas memperoleh 85,4% untuk kemudahan memahami materi dan 81,2% untuk kreativitas. Namun, media tersebut dirancang untuk konteks pembelajaran di kelas dan belum berbentuk aplikasi Android yang mengintegrasikan *speech-to-text* dan *text-to-speech* dalam satu alur percakapan dua arah.

Berdasarkan keenam penelitian tersebut, dapat disimpulkan bahwa teknologi *speech-to-text* dan *text-to-speech* telah terbukti efektif sebagai alat bantu komunikasi bagi penyandang disabilitas. Akan tetapi, sebagian besar penelitian masih menempatkan kedua teknologi tersebut secara terpisah, yaitu *speech-to-text* untuk kebutuhan tunarungu dan *text-to-speech* untuk kebutuhan tunanetra, serta banyak di antaranya berhenti pada tataran rancangan atau berbasis perangkat keras khusus. Kesenjangan inilah yang menjadi dasar penelitian ini, yaitu merancang dan membangun aplikasi *mobile* berbasis

Android yang mengintegrasikan *speech-to-text* dan *text-to-speech* dalam satu alur percakapan dua arah secara *real-time*, sehingga penyandang tunanetra dan tunarungu dapat berkomunikasi langsung tanpa bantuan pihak ketiga.

C. Kerangka Berfikir

Berdasarkan latar belakang dan kajian teori diatas maka kerangka berfikir penelitian ini adalah sebagai berikut:



Gambar 2. 2 Kerangka Berpikir