BAB I

PENDAHULUAN

A. Latar Belakang

Komunikasi bahasa antar manusia dengan manusia merupakan salah satu cara yang paling efektif untuk menyampaikan maksud dan tujuan seseorang dalam menyampaikan informasi untuk memudahkan seseorang dalam menyelesaikan pekerjaan. Informasi dapat disampaikan dalam bentuk teks, citra, suara, dan lain-lain. Pemakaian teknologi komputer untuk menjalin komunikasi manusia dengan komputer sudah menjadi suatu kebutuhan, jika komputer mengerti ucapan yang dimaksudkan manusia bisa menjadi suatu kemudahan dalam pengoperasian komputer, seperti *voice command*, akses kontrol sistem berbasis suara, dan identifikasi suara untuk keamanan sistem.

Perkembangan *speech recognition (speech to text)* berjalan cukup pesat pada saat ini dilihat dari banyaknya jurnal yang membahas mengenai *speech to text*. Suara manusia mempunyai karakteristik yang sangat kompleks, satu kata yang diucapkan oleh orang yang berbeda bisa menghasilkan karakteristik suara yang berbeda, namun suatu sistem diharuskan dapat mengenali sebagai suatu kata yang sama. Selain itu faktor yang mempengaruhi suara adalah kesehatan, psikologi, umur, dan jenis kelamin seseorang.

Speech to text memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan

dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasikan kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan.

B. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, dapat dirumuskan permasalahan yang dihadapi yaitu bagaimana membuat suatu aplikasi *speech to text* yang dapat mengenali ucapan kata berbahasa Indonesia.

C. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Aplikasi ini hanya mendeteksi suara
2. Aplikasi yang digunakan yaitu *Android Studio*
3. Output yang dihasilkan yaitu berupa Teks

D. Manfaat Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah membuat aplikasi *speech to text* berbahasa Indonesia. Manfaat dari penelitian skripsi ini adalah

1. Sebagai dasar dalam penelitian pengembangan aplikasi *speech to text*
2. Membantu pengguna yang membutuhkan konversi suara ke teks seperti orang yang tidak bisa menulis.

E. Sistematika Penulisan

Sistematika penulisan dibagi menjadi beberapa bagian yaitu.

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang masalah, maksud dan tujuan yang akan dicapai dalam pelaksanaan skripsi ini. Penulis juga memberikan batasan-batasan masalah pada bagian akhir dan Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Menguraikan teori yang erat hubungannya dengan cara kerja komponen yang digunakan dan penjelasan umum dalam pembuatan Aplikasi dengan.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan mengenai tahapan-tahapan yang dilalui dalam penyelesaian tugas akhir ini, yaitu tempat penelitian dan waktu penelitian, alat dan bahan penelitian, metode pengumpulan data, tahapan penelitian, serta metode pengujian.

BAB IV HASIL DAN PEMBAHASAN

Berisi tentang rancangan dan pembuatan aplikasi yang meliputi perangkat lunak serta cara kerja keseluruhan dan pengujian terhadap cara kerja *software* itu sendiri yaitu basis *path* (*prosedure* programnya).

BAB V PENUTUP

Bab ini merupakan bagian terakhir yang merangkum semua hasil yang diperoleh pada saat penyusunan tugas akhir dan saran-saran yang dapat membantu penyempurnaan pembahasan tugas akhir ini.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Sistem Pembaca Pesan *Text To Speech* Berbasis Android Menggunakan *Framework Google Api*, Nurpadila Muslimin, Universitas Hasanuddin Makassar 2019. Pembaca pesan yang berjalan pada sistem operasi Android dan terintegrasi dengan sistem text to speech dapat menjadi solusi alternatif bagi pengguna yang ingin melihat atau mendengarkan pesan yang disampaikan kepadanya tanpa harus mengganggu aktifitas. Sistem pembaca pesan *text to speech* akan menerima pesan instan dari *Telegram*, *Line* dan *WhatsApp* menjadi ucapan dan bekerja di belakang layar serta mengkonversi teks pesan yang diterima jika pesan mengandung singkatan. Dengan menggunakan *Android Studio*, bahasa pemograman *Java* dan sistem manajemen *database SQLite* serta *Notification Listener Sevice* membuat sistem mampu menerima *pack*, *tittle* dan *text* dari pemberitahuan pesan *Telegram*, *Line* dan *WhatsApp* yang ditampilkan pada layar *smartphone* sehingga sistem berjalan dibelakang layar tanpa harus membuka aplikasi utama.

Aplikasi *Speech To Text* Berbahasa Indonesia Menggunakan Mel-*Frequency Cepstral Coefficient Dan Hidden Markov Model*, Eko Widayanto, Universitas Diponegoro Semarang 2020. Pengenalan ucapan adalah suatu teknik yang memungkinkan sistem komputer untuk menerima *input* berupa kata yang

diucapkan oleh seseorang tanpa memperdulikan identitas hingga dimengerti oleh komputer. Kata-kata yang ditangkap dan dikenali oleh komputer bisa jadi hasil akhir, untuk sebuah aplikasi seperti *command* dan *control*, penginputan data, dan persiapan dokumen. Aplikasi *speech to text* berbahasa Indonesia dapat menjadi solusi untuk mengenali kata dari sebuah ucapan. Aplikasi ini dikembangkan menggunakan *MelFrequency Cepstral Coefficient* (MFCC) yang digunakan untuk ekstraksi ciri dari sinyal ucapan dan *Hidden Markov Model* untuk pembentukan model dan menentukan nilai kemiripan kata yang tidak diucapkan dengan kata-kata yang ada dalam sebuah model. Data pengujian menggunakan 10 *speaker* (5 laki-laki dan 5 perempuan) yang masing-masing mengucapkan 15 kata dan untuk setiap katanya diucapkan sebanyak 4 kali.

Aplikasi Dongeng Berbasis *Text To Speech* Untuk Platform Android, Uswatun Hasanah, Politeknik Negeri Bengkalis 2018. Anak-anak suka mendengarkan dongeng karena memberikan kesempatan kepada mereka untuk berimajinasi dengan menggambarkan peristiwa pada dongeng ke dalam khayalan. Dikarenakan banyak anak-anak menyukai dongeng, banyak peluang pengembangan aplikasi dongeng seperti menggunakan *text to speech*. Maka dari itu dilakukan pengembangan aplikasi dongeng menggunakan konsep *Text to Speech* untuk mengkonversikan text dongeng menjadi suara sebagai media pengganti membaca dongeng. Tujuan penelitian ini adalah untuk membuat aplikasi dongeng *text to speech* untuk platform android. Metode yang digunakan dalam penelitian ini ialah menggunakan metode *waterfall*. Aplikasi ini dirancang dengan menggunakan pemograman *java* dan *editor eclipse*. Data yang digunakan

berupa teks dongeng dan pemanfaatan *google text to speech* untuk menghasilkan keluaran suara. Hasil akhir berupa aplikasi pada perangkat *android*, dan telah dilakukan pengujian. Hasil pengujian menunjukkan Aplikasi dongeng *text to speech* sudah mampu membaca dongeng, walaupun dari segi intonasi suara belum alami sebagai mana manusia membaca dongeng. Pengujian sistem dilakukan bahwa aplikasi ini hanya bisa digunakan pada perangkat android versi *Lollipop* (5.1.1) untuk menghasilkan keluaran suara bahasa Indonesia secara *offline*. Dari segi tampilan aplikasi, aplikasi ini belum Fleksibel sehingga hanya terlihat baik pada perangkat tertentu.

B. Pengenalan Ucapan

Pengenalan ucapan atau suara (*speech recognition*) adalah suatu teknik yang memungkinkan sistem komputer untuk menerima *input* berupa kata yang diucapkan. Kata-kata tersebut diubah bentuknya menjadi sinyal *digital* dengan cara mengubah gelombang suara menjadi sekumpulan angka lalu disesuaikan dengan kode-kode tertentu dan dicocokkan dengan suatu pola yang tersimpan dalam suatu perangkat. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi.

Ukuran kosakata (*vocabulary*) dari sistem pengenalan suara mempengaruhi kompleksitas, parameter pelatihan dan akurasi sistem. Beberapa aplikasi pengenalan suara hanya memerlukan beberapa kata, sedangkan yang lainnya memerlukan kamus yang sangat besar (misalnya mesin pendiktean). Terdapat 4 jenis ukuran kosakata, yaitu:

a. Kosakata ukuran kecil *(small vocabulary)* yang terdiri dari puluhan kata.
b. Kosakata ukuran sedang *(medium vocabulary)* yang terdiri dari ratusan kata.
c. Kosakata ukuran besar*(large vocabulary)* yang terdiri dari ribuan kata,
d. Kosakata ukuran sangat besar *(very large vocabulary)*yang terdiri dari puluhan ribu kata.

Teknologi pengenalan suara berkembang dengan begitu pesat. Beberapa jenis diantaranya yaitu:

a. Sintesis suara *(speech synthesis)*: dari teks ke suara.
b. Pengenalan pembicara *(speaker recognition)*: dari suara ke identitas pembicara.
c. Pendiktean *(dictation)*: dari suara ke teks.
d. Rangkuman suara *(speech summarization)*: dari suara ke teks sederhana.
e. Pengkategorian suara *(speech categorization)*: dari suara ke label kelas.
f. Pengertian suara *(speech understanding)*: dari suara ke representasi makna suara.
g. Pemrosesan dialog *(dialog processing)*: dari suara ke makna suara yang interaktif.
h. Penerjemah suara *(speech translation)*: dari suara ke suara dalam bahasa lain.

Kegunaan aplikasi pengenal ucapan yaitu:

1. Pada bidang komunikasi:

a. Sebagai komando suara pada komputer untuk melakukan perintah seperti membuka, menyimpan, menutup *file* dan sebagainya.
b. Sebagai alat pendiktean untuk membuat laporan dimana aplikasi akan menuliskan teks sesuai dengan yang diucapkan pembicara.

2. Pada bidang kesehatan, alat pengenal ucapan digunakan untuk membantu para penyandang cacat agar dapat beraktivitas dengan memerintahkan alat-alat bantu melalui suaranya.
   a. Pada bidang militer, aplikasi alat pengenal ucapan digunakan pada pengatur lalu lintas udara yang dikenal dengan *Air Traffic Controllers* (ATC). Alat ini dipakai untuk memberi keterangan mengenai keadaan lalu lintas udara seperti radar, cuaca, dan navigasi dengan kata lain sebagai pengganti operator yang memberikan informasi kepada pilot dengan cara berdialog.
   b. Secara umum, pengenal ucapan banyak digunakan karena mudah yakni hanya menggunakan suara dan cepat prosesnya.

Kekurangan pengenalan ucapan yaitu:

3. Rawan terhadap gangguan sinyal suara lain terutama di tempat yang ramai.
   a. Kata-kata yang diucapkan dapat sulit dikenali karena cara pengucapan yang berbeda walaupun oleh pembicara yang sama. Intonasi, logat, dan kecepatan pengucapan sangat mempengaruhi.
   b. Bahasa lisan seringkali diucapkan tidak sesuai dengan kaidah tata bahasa yang baku.
   c. Jumlah kata yang dapat dikenali masih terbatas.

C. Pengertian Aplikasi

Aplikasi menurut Jogiyanto (2018:12) adalah penggunaan dalam suatu komputer, instruksi (*instructiom*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memprosesinput menjadi output . Menurut Kamus Kamus Besar Bahasa Indonesia (2018 : 52) , "Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atauketentuan bahasa pemrograman tertentu". Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset* . Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemproses kata dan *Web Browser*. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app. Secara *historis*, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. *App* adalah *sofware* yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah

ini untuk merefleksikan medan pertempuran persaingan yang baru, yang paralel dengan yang terjadi antar sistem operasi yang dimunculkan.

D. *Speech Recognition*

Speech recognition adalah kemampuan mesin atau program untuk mengidentifikasi kata-kata dan *frase* dalam bahasa lisan dan mengkonversikannya ke format yang dapat dibaca mesin. *Software* pengenalan suara dasar memiliki kosakata yang terbatas kata-kata dan *frase* dan hanya dapat mengidentifikasi ini jika mereka berbicara dengan sangat jelas (Margareth, 2008).

Adapun tipologi *speech recognition* adalah (Saini, 2013):

1. *Speaker Dependent*: yaitu sistem yang memerlukan pengguna untuk melatih sistem sesuai dengan suaranya.
2. *Speaker Independent* : yaitu sistem yang tidak memerlukan pengguna untuk melatih sistem yaitu mereka yang dikembangkan untuk beroperasi untuk setiap pembicara.
3. Pengenal kata Terisolasi: yaitu sistem yang menerima satu kata dalam satu waktu. Sistem pengenalan ini memungkinkan kita untuk berbicara secara alamiterus menerus.
4. Sistem kata tersambung : yaitu sistem yang memungkinkan pembicara untuk berbicara perlahan dan jelas setiap kata dengan jeda singkat yaitu ucapan yang direncanakan.
5. Sistem pengenalan spontan: yaitu sistem yang memungkinkan kita untuk berbicara secara spontan.

Pada teknologi *speech recognition* terdapat beberapa macam kata yang dijadikan *input*, ada beberapa macam kata yakni *Isolated word, Connected word, Continuous speech, Spontaneous speech* (Yanuar et al, 2010).

E. Recognation

```
init {
        recognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_WEB_SEARCH)

recognizerIntent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
context.packageName)
        recognizerIntent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3)
        recognizerIntent.putExtra(RecognizerIntent.EXTRA_PREFER_OFFLINE,
true)

        initializeRecognizer()
    }

    private fun initializeRecognizer() {
        if (SpeechRecognizer.isRecognitionAvailable(context)) {
            speech = SpeechRecognizer.createSpeechRecognizer(context)
            speech?.setRecognitionListener(this)
            callback?.onPrepared(if (speech != null)
RecognitionStatus.SUCCESS else RecognitionStatus.FAILURE)
        } else {
            callback?.onPrepared(RecognitionStatus.UNAVAILABLE)
        }
    }

    fun destroyRecognizer() {
        muteRecognition(false)
        speech?.destroy()
    }

    fun startRecognition() {
        if (!isListening) {
            isListening = true
            speech?.startListening(recognizerIntent)
        }
    }

    fun stopRecognition() {
        speech?.stopListening()
    }

    fun cancelRecognition() {
        speech?.cancel()
    }
```

```
    private fun muteRecognition(mute: Boolean) {
        val flag = if (mute) AudioManager.ADJUST_MUTE else
AudioManager.ADJUST_UNMUTE
        audioManager.adjustStreamVolume(AudioManager.STREAM_MUSIC, flag,
0)
        audioManager.adjustStreamVolume(AudioManager.STREAM_SYSTEM,
flag, 0)
    }

    override fun onBeginningOfSpeech() {
        callback?.onBeginningOfSpeech()
    }

    override fun onReadyForSpeech(params: Bundle) {
        muteRecognition((shouldMute != null && shouldMute) ||
!isActivated)
        callback?.onReadyForSpeech(params)
    }

    override fun onBufferReceived(buffer: ByteArray) {
        callback?.onBufferReceived(buffer)
    }

    override fun onRmsChanged(rmsdB: Float) {
        callback?.onRmsChanged(rmsdB)
    }

    override fun onEndOfSpeech() {
        callback?.onEndOfSpeech()
    }

    override fun onError(errorCode: Int) {
        if (isActivated) {
            callback?.onError(errorCode)
        }
        isActivated = false
        isListening = false

        when (errorCode) {
            SpeechRecognizer.ERROR_RECOGNIZER_BUSY ->
cancelRecognition()
            SpeechRecognizer.ERROR_SPEECH_TIMEOUT -> {
                destroyRecognizer()
                initializeRecognizer()
            }
        }

        startRecognition()
    }

    override fun onEvent(eventType: Int, params: Bundle) {
        callback?.onEvent(eventType, params)
    }
```

```
    override fun onPartialResults(partialResults: Bundle) {
        val matches =
partialResults.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION)
        if (isActivated && matches != null) {
            callback?.onPartialResults(matches)
        }
    }

    override fun onResults(results: Bundle) {
        val matches =
results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION)
        val scores =
results.getFloatArray(SpeechRecognizer.CONFIDENCE_SCORES)
        if (matches != null) {
            if (isActivated) {
                isActivated = false
                callback?.onResults(matches, scores)
            } else {
                matches.forEach {
                    if (it.contains(other = activationKeyword,
ignoreCase = true)) {
                        isActivated = true
                        callback?.onKeywordDetected()
                        return@forEach
                    }
                }
            }
        }

        isListening = false
        startRecognition()
    }
}
```

F. *Java*

Menurut Shalahuddin (2008:1), *java* adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source.*

1. Sebagian Fitur dari *Java*

a. *Java Virtual Machine (JVM)*

Menurut Shalahuddin (2008:10), *JVM* adalah sebuah mesin

imajiner *(maya)* yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. *JVM* menyediakan spesifikasi *hardware* dan *platform* dimana kompilasi kode *Java* terjadi. Spesifikasi inilah yang membuat aplikasi berbasis *Java* menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM. Aplikasi program *Java* diciptakan dengan *file* teks berekstensi. *Java.* Program ini dikompilasi menghasilkan satu berkas *bytcode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *Java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

b. *Garbage Collection*

Banyak bahasa pemrograman lain yang mengijinkan seorang programmer mengalokasikan memori pada saat dijalankan. Program *Java* melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek–objek yang tidak digunakan lagi.

c. *Code Security*

Code Security terimplementasi pada *Java* melalui penggunaan *Java Runtime Environment (JRE). Java* menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.

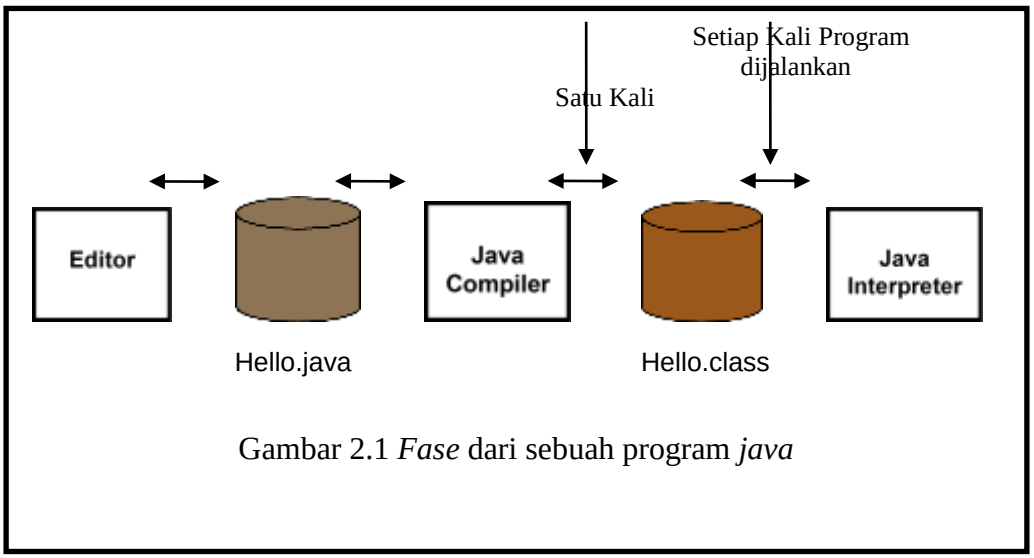
1. Pertama, *class-loader* menangani pemuatan kelas *Java* ke runtime *interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas–kelas yang berasal dari *localdisk* dengan kelas–

kelas yangdiambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas–kelas yang berasal dari *localdisk* yang dimuat terlebih dahulu.

2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan danmenjamin *bytecode* memenuhi aturan–aturan dasar bahasa *Java.*
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem *windowing*.

d. *Fase-fase* pemrogram *Java*

Gambar di bawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program *Java*:



Gambar 2.1 *Fase* dari sebuah program *java*

1. *Editor* merupakan aplikasi untukmembuat atau mengedit aplikasi *java*.
2. *Hello.java* merupakan *file java* yang sudah mempunyai kode-kode *java* dan di simpan dengan extensi *java*.

3. *Java compiler* merupakanmerubah *file java* yang extensi *java* menjadi *file* yang *extensi class*.
4. *Hello.class* merupakan hasil *file java compiler* yang asalnya berupa *file extensi java*.
5. *Java interpreter* merupakan menterjemahkan baris per baris kode ke dalam bahasa mesin dan menampilkan hasilnya.

G. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi baru. *Android* merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkan. Pada saat perilisan perdana *android*, 5 November 2007, android bersama *open handset alliance* menyatakan mendukung pengembangan *open source* pada perangkat lunak *mobile*. Di lain pihak, *google* merilis kode-kode *android* dibawah lisensi *apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat *seluler*. Safaat (2011:1).

1. *Android* SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi, sistem operasi, *middleware* dan aplikasi yang di

release oleh *google*. Saat ini disediakan *android* SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi platform *android* menggunakan bahasa pemrograman *java*. Sebagai platform aplikasi-netral, *android* memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang buka merupakan aplikasi bawaan *handphone*. Safaat.

H. *Database SQLite*

Menurut Jay A. Kreibich (2010,12) *SQLite* merupakan paket perangkat lunak yang bersifat *public domain* yang menyediakan sistem manajemen basis data relasional atau RDBMS. Sistem basis data relasional digunakan untuk menyimpan *record* yang didefinisikan oleh pengguna pada ukuran tabel yang besar dan memproses perintah *query* yang kompleks dan menggabungkan data dari berbagai tabel untuk menghasilkan laporan dan rangkuman data. Kata '*Lite'* pada *SQLite* tidak menunjuk pada kemampuannya, melainkan menunjuk pada sifat dari *SQLite*, yaitu ringan ketika dihubungkan dengan kompleksitas pengaturan, *admin istrative overhead*, dan pemakaian sumber.*SQLite* memiliki fitur-fitur sebagai berikut :

1. Tidak memerlukan *server* Arsitektur *SQLite* tidak memiliki arsitektur *client server*. Kebanyakan sistem *database* skala besar memiliki paket *server* yang besar yang membentuk mesin *database*.
2. *Single File Database SQLite* mengemas seluruh *database* ke dalam suatu *single file*. *Single file* tersebut berisi *layout database* dan data aktual yang berada pada tabel dan indeks yang berbeda. Format *file* dapat digunakan

pada banyak *platform* dan dapat diakses pada mesin manapun tanpa memperhatikan *native byte order* ataupun ukuran kata. Pengemasan *database* kedalam suatu *file* tunggal memudahkan pengguna untuk membuat, menyalin, ataupun mem-*backupimage database* yang berada di dalam media penyimpanan

3. *Zero Configuration SQLite* tidak membutuhkan apapun untuk melakukan instalasi dan konfigurasi. Dengan mengeliminasi *server* dan menggabungkan *database* secara langsung ke dalam aplikasi, maka pengguna tidak perlu mengetahui bahwa mereka sedang menggunakan *database*.
4. *Embedded Device Support* Ukuran *code* dari *SQLite* bersifat kecil dan penggunaan sumber daya yang konservatif membuatnya cocok digunakan untuk *embedded system* yang berjalan terbatas pada sistem operasi.
5. Fitur-fitur yang unik *SQLite* menggunakan sistem dengan tipe dinamis untuk tabel-tabel. *SQLite* memungkinkan pengguna untuk memasukkan nilai ke dalam kolom tanpa memperhatikan tipe data. Pada beberapa cara pemakaiannya, sistem yang bertipe dinamis pada *SQLite* mirip dengan sistem yang ditemukan pada bahasa *scripting* yang populer, yang sering memiliki sebuah tipe skalar yang dapat menerima semua tipe data dari *integer* sampai *string*. Fitur lainnya adalah kemampuan untuk memanipulasi lebih dari satu basis data pada satu waktu. *SQLite* mempunyai kemampuan dalam menghubungkan sebuah koneksi *database* tunggal dengan banyak *file basis* data secara bersamaan. Hal ini

memungkinkan *SQLite* umtuk memproses *SQL* statement yang menjembatani beberapa basis data sekaligus.

6. *Compatible License SQLite* dan *SQLite code* tidak memiliki lisensi pengguna dan tidak dilindungi oleh *GNU's Not Unix* (GNU) *General Public License* (GPL) atau lisensi *open source* sejenisnya. Hal ini berarti pengguna dapat melakukan apapun dengan *source code SQLite*, sehingga *library code* dapat digunakan dengan berbagai cara, dimodifikasi dengan berbagai cara dan didistribusikan dengan berbagai cara.
7. *Highly reliable* Sejumlah tes telah dilakukan sebelum l*ibrary* SQLite masing-masing dirilis. Hal ini dilakukan untuk mempertahankan tingkat kehandalan yang tinggi.

I. UML (*Unified Modeling Language*)

Menurut Adi nugroho dalam buku "Rekayasa Perangkat Lunak Beriorentasi Objek" (2010:6) UML (*Unified Modeling Language*) adalah 'bahasa' pemodelan untuk *system* atau perangkat lunak yang berparadigma 'beriorentasi objek'. Pemodelang (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

a. *Use Case* Diagram

Use case diagram bersifat statis, memperlihatkan himpunan *use-case* dan aktor-aktor. Diagram ini sangat penting terutama untuk memodelkan ataupun mengorganisasikan perilkau dari system yang dibutuhkan pengguna.

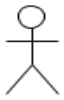
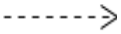
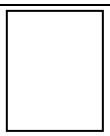
b. *Activity* Diagram

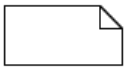
Activity Diagram bersifat dinamis, merupakan tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainya dalam suatu sistem.

c. *Class* Diagram

Class Diagram bersifat statis tetapi sering pula memuat kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.

Tabel 2.1 Simbol Diagram *Use Case*

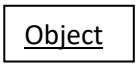
NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya sehingga elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan suatu paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

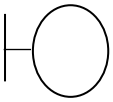
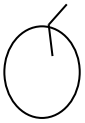
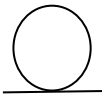
NO	GAMBAR	NAMA	KETERANGAN
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Tabel 2.2 Simbol Diagram *Activity*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 2.3 Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Object (Partisipan)*	Object atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah *class* (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma.
2		*Actor*	Actor jug dapat berkomunikasi dengan *object*, maka *acto*r juga dapat diurutkan sebagai kolom.
3		*Life line*	*Life line* mengidentifikasikan sebuah keberadaan *object* dalam basis waktu. Notasi untuk *lifeline* adalah garis putus vertical yang ditarik oleh sebuah *object*.
4		*Collaboration*	*Activication* dinotasikan sebagai kotak segi empat yang digambar pada sebuah *lifeline action* mengidentifikasikan sebuah *object* yang akan melakukan sebuah aksi.

NO	GAMBAR	NAMA	KETERANGAN
5		*Boundary*	*Boundary* terletak diantara *system* dengan dunia sekelilingnya. Semua *form*, laporan-laporan, antar muka ke perangkat keras seperti printer atau *scanner* dan antar muka ke *system* lainya adalah termasuk dalam kategori.
6		*Control*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Entity*	*Entity* digunakan menangani informasi yang munkin akan disimpan secara permanen. *Entity* bias juga merupakan sebuah *table*.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan Skripsi ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat.

Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

B. Lokasi dan Waktu Penelitian

Lokasi penelitian dilaksanakan di dalam ruangan dan di luar ruangan Adapun waktu penelitian dilakukan selama 5 (lima) bulan dimulai pada bulan Juni sampai dengan bulan Oktober tahun 2021.

Tabel 3.1 Jadwal pelaksanaan penelitian

NO	Uraian Kegiatan	Bulan 2021				
		Juni	Juli	Agustus	September	Oktober
1	Studi Literatur					
2	Perancangan aplikasi					
3	Pembuatan aplikasi					
4	Pengujian					

C. Alat dan Bahan

Dalam Proses Penelitian dan pembuatan aplikasi, maka diperlukan beberapa alat untuk membantu dalam pengerjaan dapat dilihat pada tabel di bawah ini:

Tabel 3.2 Alat dan Bahan

No	Uraian	Qty
1	Alat tulis kantor	4
2	Alat pengukur panjang (roll meter)	1
3	Handphone	2
4	Laptop	1

D. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui beberapa cara yaitu :

1. Analisis Data

   Menganalisa data-data yang sebelumnya telah dikumpulkan.

2. Perancangan Program

   Sebagai pedoman dalam penulisan program atau kode-kode agar berjalan sesuai rencana.

3. Uji Coba Program

   Pengujian program dilakukan untuk memastikan bahwa program yang dibuat dapat berjalan dengan baik.

4. Evaluasi

   Sistem yang telah selesai dibangun perlu adanya evaluasi untuk menemukan kelemahan yang terdapat pada program yang telah

dibangun tadi, yang nantinya bisa digunakan sebagai acuan untuk memperbaiki program sehingga lebih sempurna.

E. Jenis Data

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

2. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, *website* dan jurnal.

F. Tahapan Penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

   Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan Data

   Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

3. Analisis

   Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

   Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

   Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

   Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh *user*.

G. Metode Pengujian Aplikasi

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

   Uji coba *white box testing* merupakan metode perancangan *testcase* yang mengunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

   Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* menginputkan password *user* yang salah, untuk kesalahan semacam ini akan

memberikan suatu informasi kepada *user* mengenai kesalahan yang di lakukan.

2. Black Box

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika *user* belum menginput data yang harusnya di input sesuai ketentuan sistem yang di jalankan dan sistem memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

H. Kerangka Pikir

Speech to text memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut.

↓

Tujuan yang ingin dicapai dalam penelitian ini adalah membuat aplikasi *speech to text* berbahasa Indonesia. Manfaat dari penelitian skripsi adalah sebagai dasar dalam penelitian pengembangan aplikasi *speech to text*, membantu pengguna yang membutuhkan konversi suara ke teks seperti orang yang tidak bisa menulis.

↓

Sistem yang telah selesai dibangun perlu adanya evaluasi untuk menemukan kelemahan yang terdapat pada program yang telah dibangun tadi, yang nantinya bisa digunakan sebagai acuan untuk memperbaiki program sehingga lebih sempurna.

↓

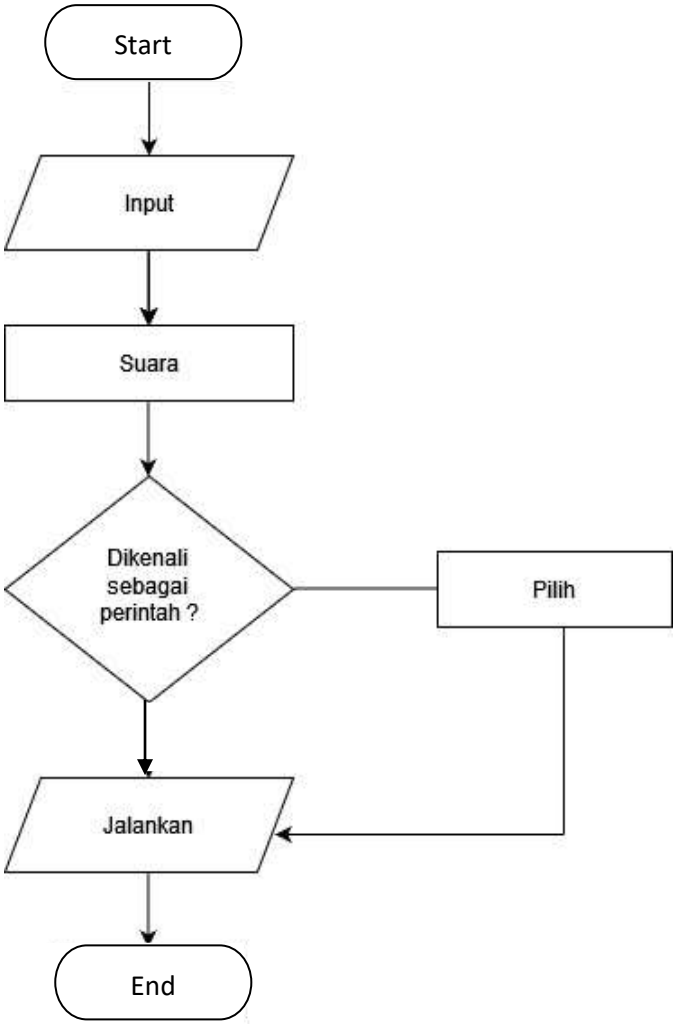
Aplikasi *voice to text* ini bekerja dengan memanfaatkan suara sebagai inputan yang akan diproses sehingga user tidak perlu mengetikkan kata yang dikirimkan. Aplikasi *voice to text* juga akan menerjemahkan hasil konversi suara ke teks

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem Yang Berjalan

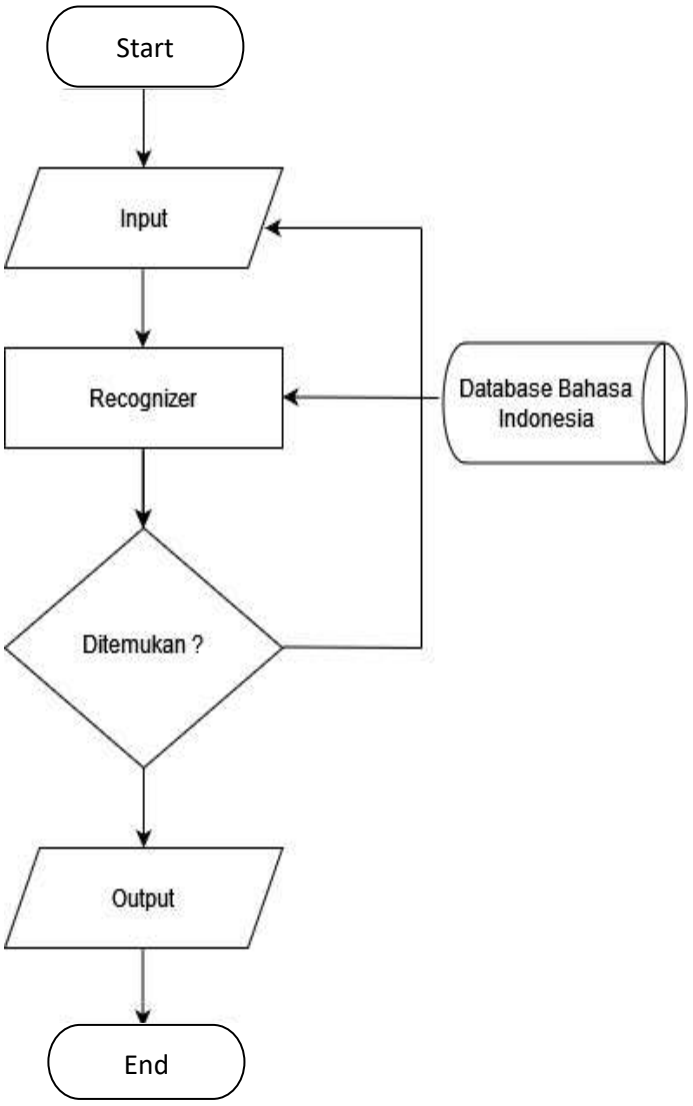
Berikut merupakan diagram dari sistem yang berjalan pada penelitian ini sebagai berikut :



Gambar 4.1. Sistem yang berjalan

B. Sistem Yang diusulkan

Berikut merupakan diagram dari sistem yang diusulkan pada penelitian ini sebagai berikut :

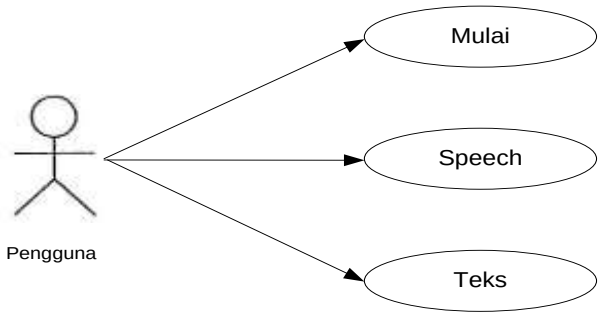


Gambar 4.2. Sistem yang diusulkan

Dari diagram alur aplikasi ini, dapat dilihat bahwa inputan berupa suara pengguna. Langkah-langkah untuk menggunakan aplikasi ini yakni pertama-tama *user* harus menekan tombol yang ada pada menu utama. Kemudian pengguna

mengucapkan kata dengan menggunakan bahasa indonesia. Kemudian *block recognizer* akan melakukan proses *recognition* dan mengubah suara yang masuk menjadi *text*. Hasil keluaran berupa *text* bahasa Indonesia yang di ucapkan.

1. *Use Case* Diagram



Gambar 4.3. Diagram *Use Case* Sistem

Penjelasan *Use Case* Diagram di atas :

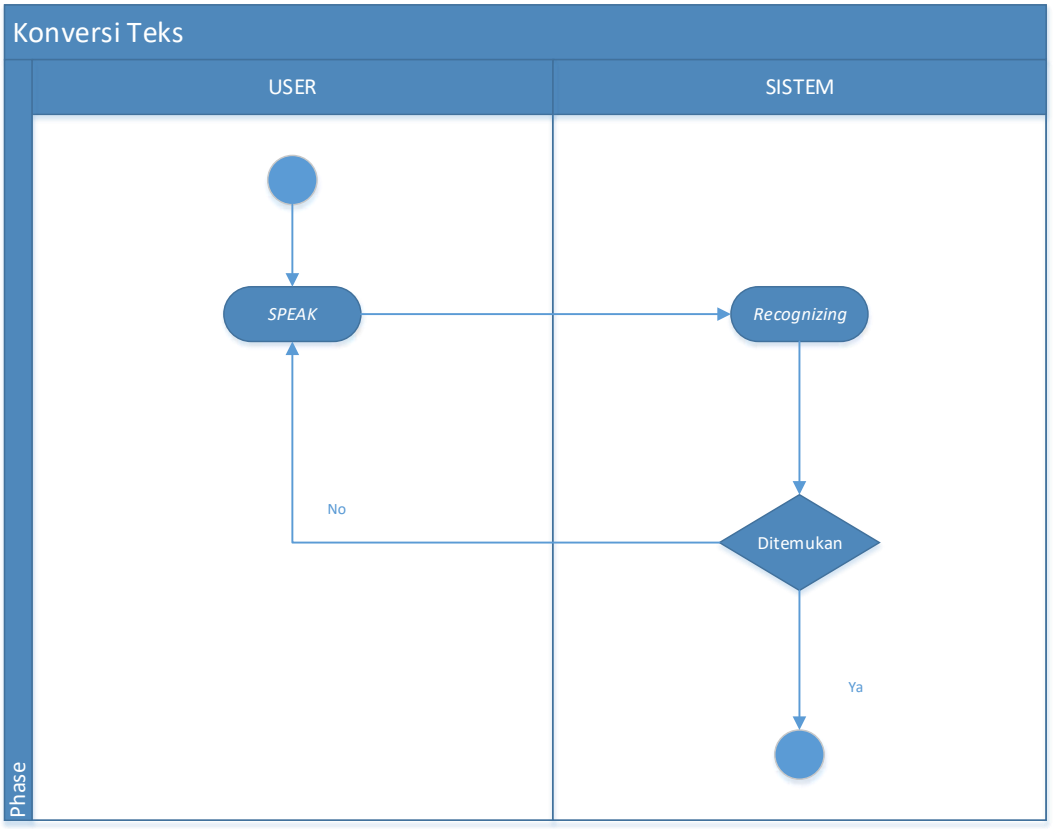
a. *Actor* Pengguna

Tabel 4.1 *Sequence* Diagram

Nama *Use Case*	Deskripsi *Use Case*
Pengguna	*Use Case* ini menjelaskan bahwa ketika pengguna atau *user* menggunakan aplikasi tersebut
Mulai	*Use Case* ini menjelaskan bahwa setelah *user* membuka aplaikasi tersebut untuk memulai
Speech	*Use Case* ini menjelaskan bahwa setelah pengguna menekan tombol *speech* maka pengguna diharapkan mengeluarkan suara agar aplikasi dapat membaca suara tersebut.
Teks	*Use Case* ini menjelaskan bahwa ketika pengguna sudah mengeluarkan suara kemudian aplikasi mendeteksi suara tersebut maka aplikasi akan menyalin suara menjadi sebuah teks pada aplikasi tersebut

2. *Activity* Diagram

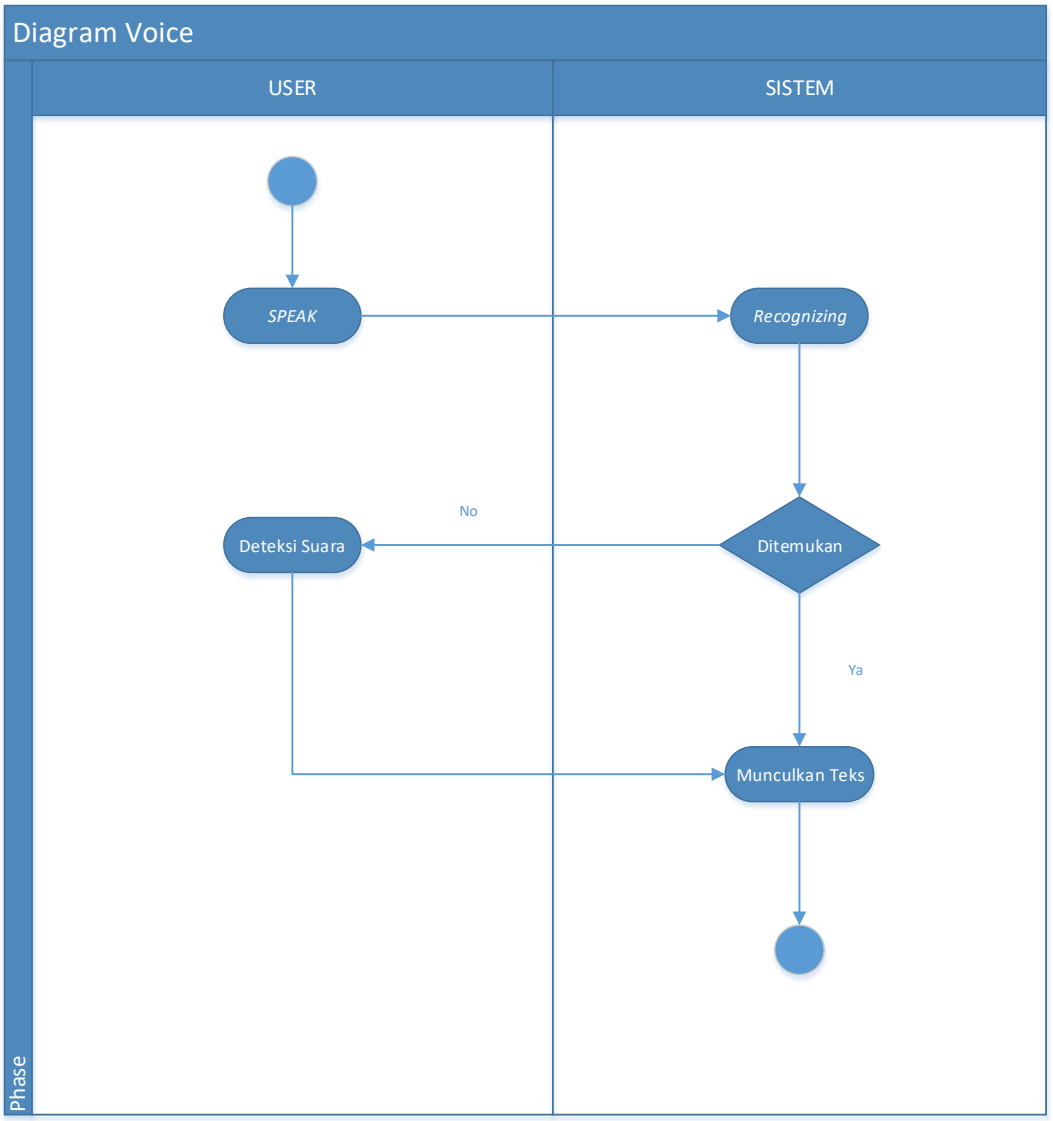
a. *Activity* Diagram konversi teks



Gambar 4.4. *Activity* Diagram konversi teks

Tahapan dari konversi teks dimana *user* menginputkan suara kemudian inputan suara akan di proses, sehingga input suara bisa diubah menjadi teks sesuai dengan inputan yang tidak terganggu oleh kebisingan di area sekitar pada saat aplikasi digunakan.

b. *Activity* Diagram *Voice*

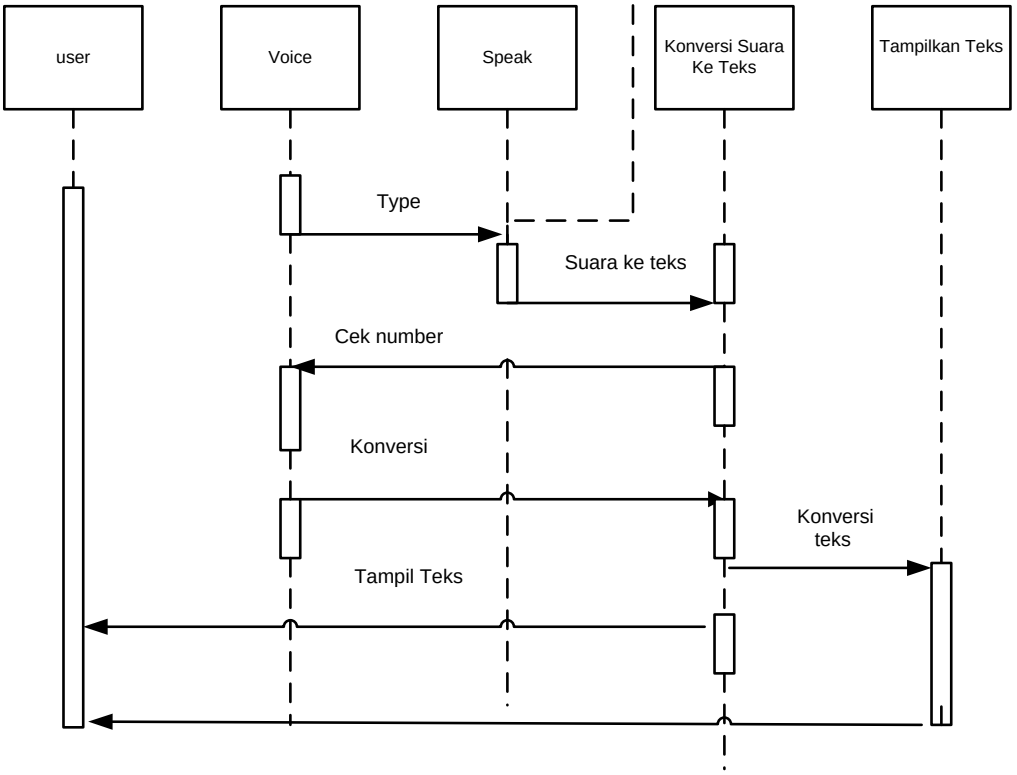


Gambar 4.5. *Activity* Diagram *voice*

Penjabaran dari aliran kerja *voice* pada tahapan ini *user* menginputkan suara dan suara akan diproses dan kata terkhir dalam inputan suara akan dikenali sehingga input suara bisa diubah menjadi teks. Jika sistem tidak mengenali *voice* maka sistem akan memberikan pilihan aplikasi untuk mengirimkan hasil dari konversi teks yang telah diproses.

3. *Sequence* Diagram

   a. *Sequence* Diagram Aplikasi



Gambar 4.6. *Sequence* Diagram

C. Proses Perhitungan dengan *Hidden Markov Model* (HMM)

1. Markov Model

Markov Model biasa disebut sebagai *Markov Chain* atau Rantai *Markov*. Model ini ditemukan oleh *Andrey Markov* yang berdasar kepada teori probabilitas yang dapat digunakan untuk memodelkan sebuah rangkaian kejadian berdasarkan atas waktu. Pada *Markov Model*, probabilitas pada sebuah state hanya bergantung pada nilai probabilitas pada state sebelumnya. Dimana sifat ini biasa dikenal dengan karakteristik *Markov* (Wiggers & RothKrantz, 2003).

Model ini merupakan bagian dari *finite state* atau *finite automaton*.*Finite automaton* sendiri adalah kumpulan state yang transisi antar state-nya dilakukan berdasarkan masukan observasi. Pada rantai *markov*, setiap transisi antar *state* berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua transisi yang keluar dari sebuah simpul sama dengan satu (Aria, 2013).

2. *Hidden Markov Model*

HMM merupakan model stokastik dimana suatu sistem yang dimodelkan diasumsikan sebagai *markov* proses dengan kondisi yang tidak terobservasi. Suatu HMM dapat dianggap sebagai jaringan *Bayesian* dinamis yang sederhana(*simplest dynamic Bayesian network*) (Prasetyo, 2010).

Hidden Markov Model (HMM) adalah sebuah sistem yang diasumsikan sebuah proses *Markov* dengan parameter yang tak diketahui, dan tantangannya adalah menentukan parameter-parameter tersembunyi *(hidden)* dari parameter yang dapat diamati (Lestary, 2010). Setiap kondisi memiliki distribusi kemungkinan disetiap output yang berbeda. Oleh karena itu urutan langkah yang dibuat oleh HMM memberikan suatu informasi tentang urutan *state*. Sifat tersembunyi *(hidden)* berarti bahwa walaupun parameter model diketahui, model tersebut tetap tersembunyi. Secara umum (Adami, 2010), HMM terdiri atas elemen-elemen berikut :

a. Himpunan nilai *output* observasi $O = \{o_1, o_2, \dots, o_M\}$, dimana$M$ adalah jumlah simbol observasi.

b. Himpunan *state* $\Omega = \{1,2,\dots,N\}$. Dimana *N*menyatakan jumlah *state* yang terdapat pada HMM.

c. Himpunan probabilitas transisi antar *state*. Diasumsikan bahwa *state* berikutnya tergantung pada *state* pada saat ini. Asumsi ini menyebabkan proses perhitungan menjadi lebih mudah dan efisien untuk dilakukan. Probabilitas transisi dapat dinyatakan dengan sebuah matriks A= { } dimana adalah probabilitas transaksi dari *state i* ke *state j*. Sebagai contoh :

$$a_{ij} = P(s_t = j | s_{t-1} = i), \qquad 1 \le i, j \le N \quad \text{.....(2.11)}$$

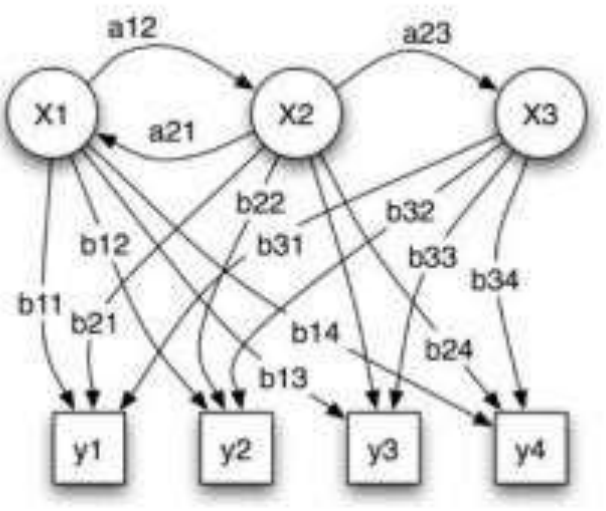
dimana merupakan state pada waktu ke-t.

d. Himpunan probabilitas *output B = { }* pada setiap *state*. Yang juga disebut probabilitas emisi,) adalah probabilitas dari simbol *output* pada *state* yang didefinisikan sebagai

$$b_i(k) = P(v_t = o_k | s_t = i) \quad \text{.....(2.12)}$$

dimana adalah simbol observasi pada waktu ke-*t.*

e. Himpunan *state* awal, dimana adalah probabilitas state *I* menjadi *state* awal pada urutan *state* HMM.



Gambar 4.7 Parameter Probabilistik pada *Hidden Markov Model*

Dimana :

x = kondisi

y = observasi yang mungkin

a = kemungkinan keadaan transisi b = kemungkinan *output*

3. Penyelesaian masalah dengan HMM

Dalam penggunaannya terdapat 3 permasalahan dasar pada HMM untuk dapat melakukan pengenalan terhadap suara (Uchat, 2009).Pertama , masalah evaluasi. Dimana diberikan sebuah λ dari HMM dan barisan observasi O = O1,O2,...,Ot dimana terdapat probabilitas obsevasi yang dihasilkan oleh model p{O | λ}.Kedua, masalah *decoding* diberikan sebuah model λ dan barisan observasi O = O1,O2,...,Ot dimana kemiripan maksimal barisan state di model yang menghasilkan observasi. Ketiga adalah masalah pembelajaran dimana diberikan model λ dan barisan pengamatan O = O1,O2,...,Ot dimana menyusaikan parameter λ = (A,B, π) untuk kita harus memaksimalkan p{O | λ) .2.4.1

Evaluation (evaluasi) 23 Diberikan barisan obervasi O = O1,O2,...,Ot, sebuah model $\lambda=(A,B,\pi)$ and $p\{O \mid \lambda)$. Ini dapat dihitung menggunakan probabilitas sederhana, namum proses perhitungan ini memiliki kompleksitas NT. Hal tersebut akan menghasilkan perhitungan dengan nilai yang sangat besar. Sehingga digunakan metode lain untuk yang menggunakan variabel tambahan yaitu αt (i), yang dinamakan variabel maju *(forward).*

4. *Learning* (pembelajaran)

Umumnya, masalah pembelajaran adalah bagaimana menyesuaikan parameter HMM yang ada. Diberikan sebuah barisan pengamatan O dimana masalah estimasi termasuk untuk menemukan parameter model yang tepat yang menentukan model yang paling optimal. Terdapat dua kriteria optimasi yang ditemukan dalam literatur ASR, pertama *maximum likelihood* (ML) dan *Maxomum Mutual Information* (MMI).

Maximum *likelihood* (ML) merupakan algoritma untuk mencari probabilitas maksimum dari barisan pengamatan. Probabilitas ini adalah total *likelihood* (kemiripan) dari obeservasi dan dapat diekpresikan secara matematis sebagai L tot = { O | λ }. Namun, parameter model yang memiliki nilai maksimum lokal dapat dipilih mengunakan prosedur iteratif. Seperti *Baum-Welch* atau Metode berbasis g*radien*.

5. Pemodelan Unit Wicara

HMM dapat digunakan untuk merepresentasikan berbagai unit suara yang mana setiap model memiliki kelebihan dan kekurangan masing-masing sesuai

dengan penggunaannya. Jenis-jenis unit suara (Hwang, 1993) yang digunakan sebagai model antara lain :

a. Kata

Model ini mampu mengenali satuan bunyi yang bervariasi karena bunyi yang sama dapat dikenali dengan model yang berbeda kata bunyi tersebut dapat terkandung dalam kata yang berbeda. Model berbasis kata ini biasanya digunakan dalam pengenalan suara yang memiliki kosa kata dengan jumlah yang kecil. Sedangkan untuk kosa kata berukuran besar, model ini tidak dapat menghasilkan pengenalan 27 yang baik karena diperlukan perulangan yang sangat banyak untuk melakukan pelatihan data.

b. Phone (fonem)

Salah satu pendekatan yang baik dalam pengenalan suara dengan kosa kata cukup besar adalah dengan menggunakan model sub kata seperti *fonem*. Dengan menggunakan model ini, proses pelatihan akan lebih mudah dilakukan karena objek pelatihan merupakan *fonem-fonem* yang terdapat dalam sebuah bahasa. Sehingga proses komputasi lebih ringan. Namun, sedikit kekurangan pada model ini adalah masalah pengucapan sebuah kata dimana bunyi f*onem* akan selalu dipengaruhi oleh *fonem* lainnya.

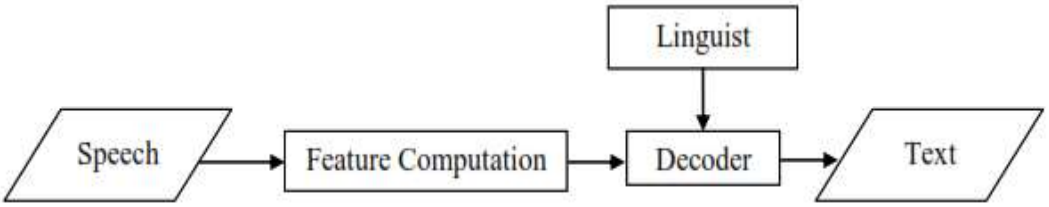
c. Multi-phone

Pengembangan dari model fonem adalah *multi-phone* yang terdiri atas *sillabel*, demi *sillabel*, dan yang lainnya.

D. Langkah-langkah konversi suara ke teks

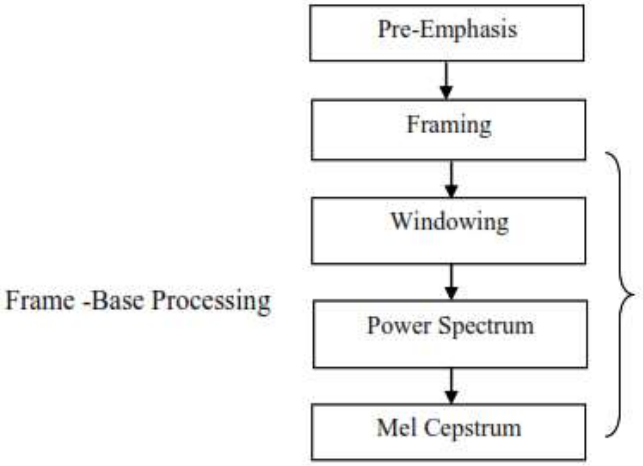
Pada aplikasi ini dapat dilihat bahwa inputan berupa suara pengguna. Langkah-langkah untuk menggunakan aplikasi ini yakni pertama-tama user harus menekan tombol yang ada pada menu utama. Kemudian pengguna mengucapkan kata dengan menggunakan bahasa indonesia.. Kemudian *block recognizer* akan melakukan proses *recognition* dan mengubah suara yang masuk menjadi text. Hasil keluaran berupa text bahasa Indonesia yang di ucapkan.

Pada *blokrecognizer* terdiri dari 3 komponen utama yakni *Front-End*, *Decoder* dan bagian *Linguist*. Di bawah ini block diagram dari *recognizer* itu sendiri :



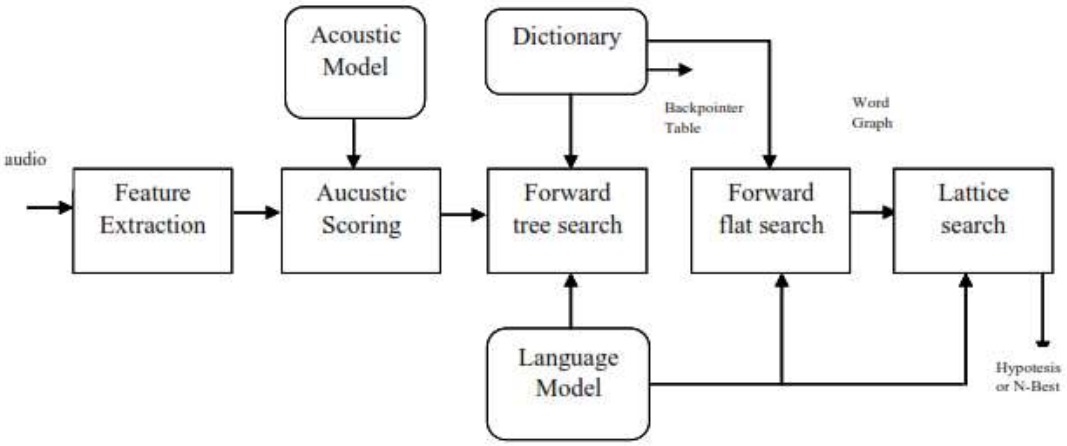
1. Feature Computation

Bagian *feature computation* merupakan masukan dari *decoder* itu sendiri. Bagian ini sendiri merupakan bagian yang berperan dalam merubahan atau tranformasi bentuk gelombang suara menjadi ciri-ciri yang unik dimana nantinya akan digunakan dalam proses pengenalan ucapan menggunakan *melfrequency cepstral coefficien*t (MFCC).



2. *Decoder*

Decoder merupakan bagian inti dari pengenal suara. *Decoder* sendiri terbagi menjadi tiga modul utama yakni :*acoustic modeling*, *forward search*, *and graph search*.



E. Detail Aplikasi

1. *Form* Aplikasi

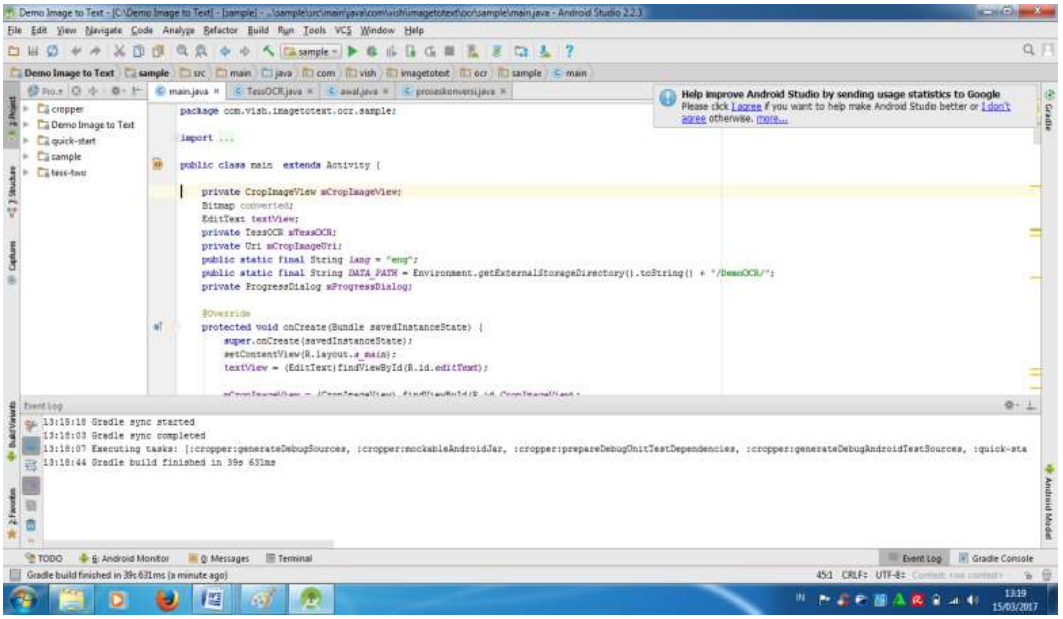


Gambar 4.8. *Form* Aplikasi

Gambar di atas adalah *Form* Aplikasi ketika *user* memulai aplikasi. nantinya *user* akan mengeluarkan suara yang akan di konversi kemudian aplikasi akan memproses suara tersebut dan menampilkan hasil konversi menjadi teks

F. Rancangan *Input* dan *Output*

1. Rancangan *input*



Gambar 4.9. *input* data Aplikasi

Gambar diatas merupakan tempat informasi di masukkan atau di *input* untuk selanjutnya di tampilkan berdasarkan suara yang dikompersi .

2. Rancangan *Output*

   a. Tampilan *output*



Gambar 4.10. tampilan aplikasi

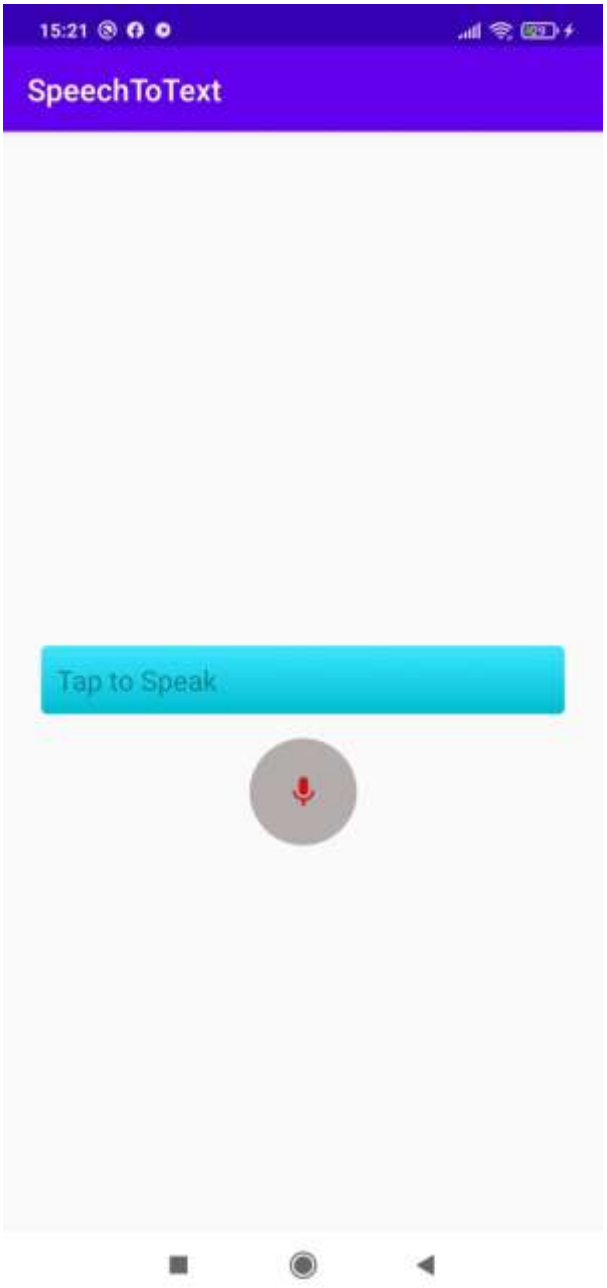
Gambar di atas adalah tampilan aplikasi ketika pertama kali di buka.

```
public void startListening(){
mSpeechRecognizer =
SpeechRecognizer.createSpeechRecognizer(mcoContext);
mSpeechRecognizerIntent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MOD
EL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_CALLING_PACK
AGE
, mcoContext.getPackageName());
mSpeechRecognizer.setRecognitionListener(new
SpeechRecognitionListener());
mSpeechRecognizer.startListening(mSpeechRecognizerIntent);
}
public void stopListening(){
if(mSpeechRecognizer!=null)
mSpeechRecognizer.destroy();
}
```

Gambar 4.11. Potongan *pseudocode speech recognition*

Gambar 4.11 merupakan potongan kode untuk memanggil fungsi *speech recognition* yang sudah ada pada *smartphone android*. Pada kode tersebut juga terdapat fungsi untuk mendengarkan inputan suara yang dilakukan oleh *user* dan apabila *input* dari *user* telah selesai maka fungsi tersebut akan dihentikan.

a. Tampilan mulai aplikasi



Gambar 4.12. Proses *speak*

Gambar diatas adalah hasil *output* proses ketika *user* berbicara yang nanti akan di ubah menjadi teks .

b. Tampilan *output* teks



Gambar 4.13. proses konversi suara ke teks

Gambar dia atas adalah hasil *output* proses konversi suara ke teks

```
class bgStuff extends AsyncTask<Void, Void, Void> {

public String text;

   String translatedText = "";

public bgStuff(String text){

this.text = text; }

@Override

protected void onPreExecute() {

// TODO Auto-generated method stub

super.onPreExecute();}

@Override

protected Void doInBackground(Void... params) {

// TODO Auto-generated method stub

try { translatedText = translateText(text);

      } catch (Exception e) {

// TODO Auto-generated catch block

translatedText = "Gagal Untuk Terjemah";

      }return null;

}

}

publicStringtranslateText(String text) throws Exception{

Translate.setClientId("Translator_text");

Translate.setClientSecret("zLvoNz/61QimQefH4ToUKcU3CsnWC0CE5LDR+
g65OTM=");

String translatedText = Translate.execute(text,
Language.ENGLISH);

return translatedText;

}
```

G. Pengujian Sistem

1. Implementasi

Implementasi sistem ini merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab - bab hasil analisis sebelumnya ke dalam suatu bahasa pemrograman.

Bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah *JAVA* dan *XML*.

2. Kebutuhan *Hardware*

Spesifikasi minimum *hardware* pengguna aplikasi ini adalah sebagai berikut :

Tabel 4.2. Spesifikasi *Hardware*

Jenis	Spesifikasi
Laptop	Semua Merk
Processor	Intel(R)Celeron(R)CPU N2840
Memory	2 GB
Hard Disk	500 GB

3. Kebutuhan *Software*

Spesifikasi minimum *software* pengguna aplikasi ini adalah sebagai berikut :

Tabel 4.3. Spesifikasi *Software*

Jenis	Spesifikasi
Sistem Operasi	Windows 10
ANDROID STUDIO	*ANDROID STUDIO*

4. Pengujian *Black Box*

Pengujian ini pengujian yang berfokus pada persyaratan fungsional perangkat lunak:

1). *Black box* tombol kamera dan tombol memulai aplikasi.

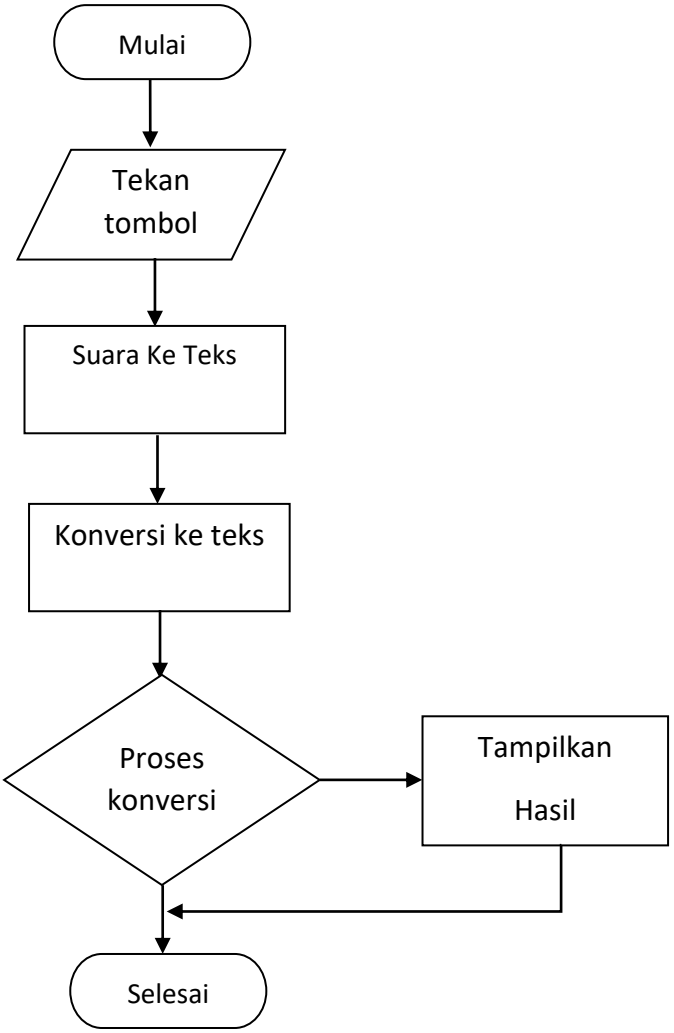
Tabel 4.4. *Black Box* Melakukan proses pada pengambilan suara.

Test Factor	Hasil	Kesimpulan
Jika tombol mikrofon di tekan akan menampilkan memulai aplikasi dan *user* akan mengeluarkan suara untuk ditampilkan ke teks	✔	Berhasil, karna teks muncul
Screenshot		


SpeechToText
Tap to Speak
SpeechToText
teknik informatika

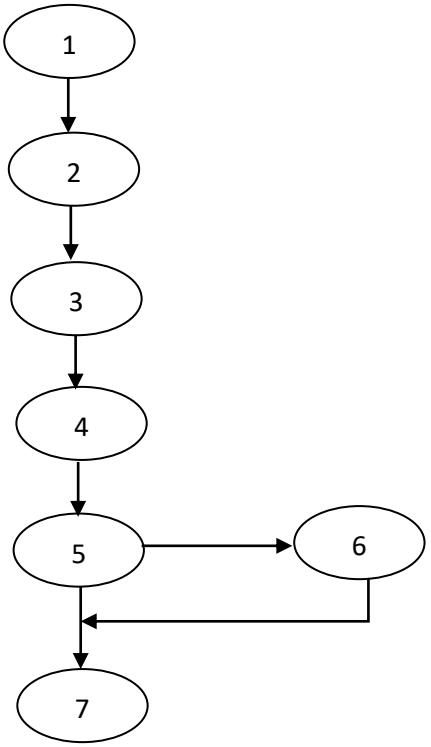
5. Pengujian *White box*

a. *Flowchart* Aplikasi



Gambar 4.14. *Flowchart* Aplikasi

b. *Flowgraph* Aplikasi



Gambar 4.15. *Flowgraph* Aplikasi

1) Proses Perhitungan

Dari Gambar *FlowGraph* di atas dapat di lakukan proses perhitungan sebagai berikut :

a. Menghitung *Cyclomatic Complexcity* V (G) = E- N + 2

N (node) = 7

E (edge) = 7

P (*predikat node)* = 1

Penyelesaian : V(G) = E – N + 2

= 7-7 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region = 2*

c. *Independent Independent Path* pada *flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 - 7

Path 2 = 1 – 2 – 3 – 4 – 5 – 7

d. Grafik Matriks

Tabel 4.5. Grafik Matriks

	1	2	3	4	5	6	7	E - 1
1		1						1-1 = 0
2			1					1-1 = 0
3				1				1-1 = 0
4					1			1-1 = 0
5						1	1	2-1 = 1
6							1	1-1 = 0
7								
Zum (E+1)								1+1 = 2

6. **Pengujian Akurasi**

a. Pengujian Fungsi Sistem

Sesuai dengan kebutuhan utama fungsi sistem aplikasi *voice translator*, yaitu mengenai konversi utama menjadi teks, terjemah hasil konversi teks kedalam bahasa inggris, membagikan hasil tejemahan pada aplikasi yang diharapkan, dapat membuka aplikasi yang di harapkan dengan perintah suara, dan dapat mengatur perintah suara sesuai dengan inpuan perintah suara yang baru oleh *user*. Seperti yang di tunjukan pada table berikut :

Tabel 4.6. Pengujian fungsi sistem

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Konversi suara menjadi teks	Dapat melakukan konversi dengan masukan suara menjadi teks	Dapat melakukan konversi dengan masukan suara menjadi teks	[✓] diterima [*x*] ditolak
Translate	Dapat Menerjemahkan teks Kedalam Bahasa Indonesia	Dapat menerjemahkan teks kedalam Bahasa Indonesia	[✓] diterima [*x*] ditolak
Share text	Dapat mengirimkan teks yang telah di terjemahkan pada aplikasi yang di harapkan	Dapat mengirimkan teks yang telah di terjemahkan pada aplikasi yang di harapkan	[✓] diterima [*x*] ditolak
Voice command	Dapat melakukan Perintah dengan Masukan suara	Dapat melakukan Perintah dengan Masukan suara	[✓] diterima [*x*] ditolak

Berdasarkan Tabel 4.6 dapat di lihat bahwa sistem telah sesuai dengan perancangan yang dilakukan sebelumnya dan fungsi-fungsi dari aplikasi *voice translator* berjalan sesuai apa yang di rancang dan di harapkan.

b. Pengujian pengenalan suara

Pada tahap uji pengenalan suara dilakukan dengan pengujian sistem dengan kata, pengujian sistem dengan frase dan kemudian pengujian sistem dengan kalimat. Berikut merupakan hasil uji sistem dengan kata :

Tabel 4.7 Pengujian sistem dengan kata

No	KATA	Jarak	Desibel	Proses	KET
1	Masukan	30 cm	80	✓	✓=terdeteksi
2	Abjad	1 m	80	✓	×= tidak terdeteksi
3	Kualitas	1,5 m	80	✓	
4	Lembab	2 m	80	✓	
5	Buku	2,5 m	80	✓	
6	Kuitansi	3 m	80	✓	
7	Efektivitas	3,5 m	80	✓	
8	Ijasah	4 m	80	✓	
9	Ayah	4,5 m	80	✓	
10	Ibu	5 m	80	✓	
11	Adik	5,5 m	80	✓	
12	Tidur	6 m	80	✓	
13	Nonton	6,5 m	80	✓	
14	Kerja	7 m	80	✓	
15	Jalan	7,5 m	80	✓	
16	Rumah	8	80	✓	
17	Manusia	8,5	80	✓	
18	Kertas	9	80	×	
19	Motor	9,5	80	×	
20	Mobil	10	80	×	

No	Kata Masukan	Kata Keluaran *(Translate)*	Bahasa Indonesia	Bahasa Inggris
1	Abjad	Abjad *(The Alphabet)*	✓	✓
2	Kualitas	Kualitas *(The Quality)*	✓	✓
3	Lembab	Lembab *(Moist)*	✓	✓
4	Makhluk	Makhluk *(Creature)*	✓	✓
5	Kwitansi	Kwitansi *(The Receipt)*	✓	✗

No	Kata Masukan	Kata Keluaran *(Translate)*	Bahasa Indonesia	Bahasa Inggris
6	Efektivitas	Efektivitas *(The Effectiveness)*	✓	✗
7	Ijasah	Ijazah *(Diploma)*	✓	✓
8	Efektif	Efektif *(Efektive)*	✓	✗
9	Ayah	Ayah *(Father)*	✓	✓
10	Ibu	Ibu *(Mother)*	✓	✓
11	Adik	Adik *(Young Brother)*	✓	✓
12	Tidur	Tidur *(Sleep)*	✓	✓
13	Nonton	Nonton *(Watch)*	✓	✓
14	Kerja	Kerja *(Work)*	✓	✓
15	Jalan	Jalan *(Street)*	✓	✓
16	Celana	Celana *(Pants)*	✓	✓
17	Seragam	Seragam *(Uniform)*	✓	✓
18	Buku	Buku *(Book)*	✓	✓
19	Sabun	Sabun *(Soap)*	✓	✗
20	Pondok	Pondok *(Cottage)*	✓	✓
21	Sekolah	Sekolah *(School)*	✓	✓
22	Kampus	Kampus *(Campus)*	✓	✓
23	Hafal	Hafal *(Memoryze)*	✓	✓
24	Ujian	Ujian *(Exam)*	✓	✓
25	Daftar	Daftar *(List)*	✓	✗

Tabel 4.7 merupakan tabel ujicoba sistem menggunakan 25 kata yang berbeda dan dari tabel uji coba 25 kata tersebut dalam bahasa indonesia sistem dapat mengenali 25 kata yang sesuai dengan inputan, jadi sistem dapat mencapai 100% kecocokan antara masukan kata dan keluaran kata. Sedangkan dalam bahasa inggris sistem dapat mengenali 20 kata yang sesuai dengan inputan dan 5 kata tidak sesuai dengan inputan, jadi sistem dapat mencapai 80% kecocokan antara masukan kata dan keluaran kata. Dan berikut hasil uji sistem dengan inputan *frase* :

Tabel 4.8 Pengujian sistem dengan *frase*

No	***Frase* Masukan**	***Frase* Keluaran *(Translate)***	**Bahasa Indonesia**	**Bahasa Inggris**
1	Nasi Goreng	Nasi Goreng *(Fried Rice)*	✓	✗
2	Rumah Kayu	Rumah Kayu *(Wooden House)*	✓	✓
3	Sepatu Kaca	Sepatu Kaca *(Glass Shoes)*	✓	✗
4	Buku Gambar	Buku Gambar *(The Book Picture)*	✓	✓
5	Lemari Besi	Lemari Besi *(Vault)*	✓	✗
6	Sangat Baik	Sangat Baik *(Very Good)*	✓	✓
7	Cukup Hebat	Cukup Hebat *(Simply Fabulous)*	✓	✗
8	Sangat Cepat	Sangat Cepat *(Very Fast)*	✓	✓
9	Sangat Mahal	Sangat Mahal *(Very Expensive)*	✓	✓
10	Sedang Tidur	Sedang Tidur *(Sleeping)*	✓	✓
11	Aku Lapar	Aku lapar *(I'm Hungry)*	✓	✓
12	Aku Kerja	Aku Kerja *(I'm Work)*	✓	✓
13	Tidak Makan	Tidak Makan *(Do not Eat)*	✓	✗

No	*Frase* Masukan	*Frase* Keluaran *(Translate)*	Bahasa Indonesia	Bahasa Inggris
14	Di Rumah	Di Rumah *(At Home)*	✓	✓
15	Ke Pasar	Ke Pasar *(To The Market)*	✓	✓
16	Di Malang	Di Malang (*In* Malang)	✓	✓
17	Provinsi Sulawesi Selatan	Provinsi Sulawesi Selatan (*South* Sulawesi *Provience*)	✓	✗
18	Bulan Pebruari	Bulan Februari *(The Month Of February)*	✓	✓
19	Cinta sejati	Cinta Sejati *(True love)*	✓	✓
20	Apa kabar	Apa kabar *(How Are You)*	✓	✓
21	Keadilan Sosial	Keadilan Sosial *(Social Justice)*	✓	✓
22	Sedang Membaca	Sedang Membaca *(Was Reading)*	✓	✓
23	Sangat Besar	Sangat Besar *(Very Large)*	✓	✓
24	Bulan Depan	Bulan Depan *(Next Month)*	✓	✗
25	Di Masjid	Di Masjid *(At The Mosque)*	✓	✗

Tabel 4.8 merupakan tabel uji coba sistem menggunakan 25 *frase* yang berbeda dan dari tabel uji coba 25 *frase* dalam bahasa indonesia sistem dapat mengenali 25 *frase* yang sesuai dengan inputan, jadi dalam uji coba menggunakan *frase* sistem juga dapat mencapai 100% kecocokan antara masukan *frase* dan keluaran *frase*. Sedangkan uji coba 25 *frase* dalam bahasa inggris sistem dapat mengenali 17 *frase* yang sesuai dengan inputan dan 8 *frase* tidak sesuai dengan inputan, jadi dalam uji coba menggunakan *frase* sistem juga dapat mencapai 68 % kecocokan antara masukan *frase* dan keluaran *frase*.

Uji coba yang terakhir dilakukan dengan menggunakan 25 kalimat yang berbeda dan didapatkan hasil 100% *output* kalimat dalam bahasa indonesia dan dalam bahasa inggris didapatkan hasil 48% yang sesuai dengan masukan kalimat. Hasil pengujian pada Tabel 4.9.

Tabel 4.9 Pengujian sistem dengan kalimat

No	**Kalimat Masukan**	**Kalimat Keluaran *(Translate)***	**Bahasa Indonesia**	**Bahasa Inggris**
1	Saya mulai dengan nama allah	Saya mulai dengan nama allah *(I started with the name of allah)*	✓	✓
2	Dimana Kamu Sekarang	Dimana Kamu Sekarang *(Where You Now)*	✓	✓
3	Hari Sabtu ke Kampus Jam 8	Hari Sabtu ke Kampus Jam 08.00 *(Saturday 08:00 to Campus)*	✓	✗
4	Revisi bab 3 halaman 30	Revisi bab 3 halaman 30 *(The revision of Chapter 3 page 30)*	✓	✗
5	Kumpulkan Laporan Pengolahan Suara pada Hari Rabu	Kumpulkan laporan pada hari Rabu *(Collect voice processing reports on Wednesday)*	✓	✓
6	Saya Tunggu di Gedung	Saya tunggu di Gedung f *(I Wait in the building f)*	✓	✓
7	Saya Sekarang di Depan Fakultas Teknik	Saya Sekarang di Depan Fakultas Teknik *(I am now in front of the Faculty of teknik)*	✓	✓
8	Saya Mahasiswa Muhammadiyah Parepare	Saya Mahasiswa Universitas Muhammadiyah Parepare *(I Am Student Of Muhammadiyah State University Of Parepare)*	✓	✓

No	Kalimat Masukan	Kalimat Keluaran *(Translate)*	Bahasa Indonesia	Bahasa Inggris
9	Teknik In*form*atika adalah Jurusan ku	Teknik InFormatika adalah Jurusan ku *(Engineering InFormatics is my Department)*	✓	x
10	Saya Tersesat dan Tidak Tahu Jalan Pulang	Saya Tersesat dan Tidak Tahu Jalan Pulang *(I was lost and didn't know the way home)*	✓	✓
11	Ujian dilaksanakan di ruang sidang besok	Ujian dilaksanakan di ruang sidang besok *(The test was carried out in the courtroom tomorrow)*	✓	x
12	Penelitian membutuhkan ketekunan yang Tinggi	Penelitian membutuhkan ketekunan yang Tinggi *(Research requires a high persistence)*	✓	x
13	Kesabaran adalah obat terbaik dalam kesulitan	Kesabaran adalah obat terbaik dalam kesulitan *(Patience is the best remedy in trouble)*	✓	x
14	Emosi tidak dapat menyelesaikan masalah	Emosi tidak dapat menyelesaikan masalah *(Emotions can not solve the problem)*	✓	x
15	Prestasi tidak dapat diraih tanpa semangat	Prestasi tidak dapat diraih tanpa semangat *(The achievement could not be achieved without the spirit)*	✓	x
16	Tugas Akhir di kumpulakan di kelas	Tugas Akhir di kumpulakan di kelas *(The final task in kumpulakan in class)*	✓	x
17	Sekolah di parepare libur pada hari sabtu	Sekolah di parepare libur pada hari sabtu *(School holiday in Surabaya on Saturday)*	✓	✓
18	Sepak bola Indonesia berkembang dengan pesat	Sepak bola Indonesia berkembang dengan pesat *(Football Indonesia grown rapidly)*	✓	✓

No	Kalimat Masukan	Kalimat Keluaran *(Translate)*	Bahasa Indonesia	Bahasa Inggris
19	Liburan bersama keluarga ke pantai	Liburan bersama keluarga ke pantai *(Vacation with the family to the beach)*	✓	✓
20	Secangkir kopi memberiku inspirasi	Secangkir kopi memberiku inspirasi *(A cup of coffee gave me inspiration)*	✓	✗
21	Fokus terhadap tujuan untuk medapatkan harapan terbaik	Fokus terhadap tujuan untuk medapatkan harapan terbaik *(Focus on the purpose to get the best hope)*	✓	✓
22	Kuliah di pindah di gedung fakultas	Kuliah di pindah di gedung fakultas *(Lectures on the moved in the building of the Faculty)*	✓	✗
23	Sejarah adalah milik para penguasa	Sejarah adalah milik para penguasa *(History belongs to the authorities)*	✓	✗
24	Rancang bangun aplikasi berbasis android	Rancang bangun aplikasi berbasis android *(Android based application architecture)*	✓	✗
25	Saya bersyukur kepada Allah	Saya bersyukur kepada allah *(I thanks to Allah)*	✓	✓

Tabel 4.10 Uji coba aplikasi dengan kalimat paragraf dengan hasil yang muncul di HP, berapa persen akurasinya

No	Kalimat	Akurasi	Proses	KET
1	Simulasi atau pemodelan merupakan sebuah pendekatan yang efektif untuk membantu menganalisis pelayanan yang relatif mendekati kondisi sebenarnya. Tujuan dari dibangunnya suatu model adalah agar dapat mempelajari suatu sistem dengan memanfaatkan alat bantu misalnya komputer dan aplikasinya untuk meniru perilaku sistem dengan sudut pandang tertentu, dimana setelah model dibangun maka dilakukan pengujian kalibrasi sampai model tersebut mendekati dengan keadaan sesungguhnya	100%	✓	✓=terdeteksi ✗= tidak terdeteksi
2	Berfungsinya alat pendukung proses perpindahan ini sesuai dengan yang diinginkan, tidaklah terlepas dari kehadiran seluruh subsistem tersebut di atas secara serentak. Masing-masing unsur itu tidak bisa hadir dan beroperasi sendiri-sendiri, kesemuanya harus terintegrasi secara serentak. Seandainya ada salah satu saja komponen yang tidak hadir maka tidak akan berfungsi	100%	✓	
3	Kapasitas jalan adalah kemampuan ruas jalan untuk menampung arus atau volume lalu lintas yang ideal dalam satuan waktu tertentu, dinyatakan dalam jumlah kendaraan yang melewati potongan jalan tertentu dalam satu jam atau dengan mempertimbangkan berbagai jenis kendaraan yang melalui suatu jalan digunakan satuan mobil penumpang sebagai satuan kendaraan dalam perhitungan kapasitas maka kapasitas habis	95%	✓	
4	Pada saat arus rendah kecepatan lalu lintas kendaraan bebas tidak ada gangguan dari kendaraan lain, semakin banyak kendaraan yang melewati ruas jalan, kecepatan akan semakin turun sampai suatu saat tidak bisa lagi arus atau volume lalu lintas bertambah, di sinilah kapasitas terjadi. Setelah itu arus akan berkurang terus dalam kondisi arus yang dipaksakan sampai suatu saat kondisi macet total, arus tidak bergerak dan kepadatan tinggi	100%	✓	

Tabel 4.11 Uji coba aplikasi dengan kalimat gabungan bahasa Indonesia dengan bahasa asing

No	Kalimat	Akurasi	Proses	KET
1	Seperti *voice command*, akses kontrol sistem berbasis suara, dan identifikasi suara untuk keamanan sistem	100%	✓	✓=terdeteksi ×= tidak terdeteksi
2	Perkembangan *speech recognition (speech to text)* berjalan cukup pesat pada saat ini dilihat dari banyaknya jurnal yang membahas mengenai *speech to text*.	100%	✓	
3	Sistem pembaca pesan text to speech akan menerima pesan instan dari Telegram, *Line* dan *WhatsApp* menjadi ucapan dan bekerja di belakang layar serta mengkonversi teks pesan yang diterima jika pesan mengandung singkatan.	95%	✓	
4	Kata-kata yang ditangkap dan dikenali oleh komputer bisa jadi hasil akhir, untuk sebuah aplikasi seperti *command* dan *control*, penginputan data, dan persiapan dokumen.	90%	✓	
5	Kosakata ukuran sedang *medium vocabulary* yang terdiri dari ratusan kata	100%	✓	
6	Pengertian suara *speech understanding* dari suara ke representasi makna suara	100%	✓	
7	*Hidden Markov Model* adalah sebuah sistem yang diasumsikan sebuah proses *Markov* dengan parameter yang tak diketahui	90%	✓	
8	*Hello.class* merupakan hasil *file java compiler* yang asalnya berupa *file extensi java*	90%	✓	
9	Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka *Library Research*	95%	✓	
10	Contoh pengujian *white box testing* ini merupakan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada user mengenai kesalahan yang di lakukan	100%	✓	

Tabel 4.12 Uji coba aplikasi dengan suara kendaraan yang di samping apakah terdeteksi

No	KATA	Jarak	Akurasi	Proses	KET
1	Saya Sekarang di Depan Fakultas Teknik	1 m	15%	✓	✓=terdeteksi ✗= tidak terdeteksi
2	Saya Mahasiswa Muhammadiyah Parepare	1,5 m	25%	✓	
3	Ujian dilaksanakan di ruang sidang besok	2 m	40%	✓	
4	Penelitian membutuhkan ketekunan yang Tinggi	2,5 m	40%	✓	
5	Kesabaran adalah obat terbaik dalam kesulitan	3 m	50%	✓	
6	Fokus terhadap tujuan untuk medapatkan harapan terbaik	3,5 m	60%	✓	
7	Kuliah di pindah di gedung fakultas	4 m	75%	✓	
8	Sejarah adalah milik para penguasa	4,5 m	85%	✓	
9	Rancang bangun aplikasi berbasis android	5 m	100%	✓	
10	Saya bersyukur kepada Allah	5,5 m	100%	✓	
11	Sekolah di parepare libur pada hari sabtu	6 m	100%	✓	
12	Sepak bola Indonesia berkembang dengan pesat	6,5 m	100%	✓	
13	Liburan bersama keluarga ke pantai	7 m	100%	✓	

No	KATA	Jarak	Akurasi	Proses	KET
14	Secangkir kopi memberiku inspirasi	7,5 m	100%	✓	✓=terdeteksi ✗= tidak terdeteksi
15	Saya Tersesat dan Tidak Tahu Jalan Pulang	8 m	100%	✓	

Tabel 4.13 Uji coba aplikasi dengan *desibel* suara yang sama dengan suara orang yang di samping

No	KATA	Jarak	Akurasi	Proses	KET
1	Saya Sekarang di Depan Fakultas Teknik	1 m	80%	✓	✓=terdeteksi ✗= tidak terdeteksi
2	Saya Mahasiswa Muhammadiyah Parepare	1,5 m	80%	✓	
3	Ujian dilaksanakan di ruang sidang besok	2 m	75%	✓	
4	Penelitian membutuhkan ketekunan yang Tinggi	2,5 m	75%	✓	
5	Kesabaran adalah obat terbaik dalam kesulitan	3 m	60%	✓	
6	Fokus terhadap tujuan untuk medapatkan harapan terbaik	3,5 m	50%	✓	
7	Kuliah di pindah di gedung fakultas	4 m	35%	✓	
8	Sejarah adalah milik para penguasa	4,5 m	25%	✓	

No	KATA	Jarak	Akurasi	Proses	KET
9	Rancang bangun aplikasi berbasis android	5 m	15%	✓	✓=terdeteksi ✗= tidak terdeteksi
10	Saya bersyukur kepada Allah	5,5 m	10%	✓	
11	Sekolah di parepare libur pada hari sabtu	6 m	5%	✓	
12	Sepak bola Indonesia berkembang dengan pesat	7 m	0%	✓	
13	Liburan bersama keluarga ke pantai	7,5 m	0%	✓	
14	Secangkir kopi memberiku inspirasi	8 m	0%	✓	
15	Saya Tersesat dan Tidak Tahu Jalan Pulang	9 m	0%	✓	

Tabel 4.14 Uji coba aplikasi suara lebih cepat dan suara lebih pelan berapa persen akurasinya

No	KATA	Jarak	Pelan	Cepat	Proses	KET
1	Saya Sekarang di Depan Fakultas Teknik	50 cm	100%	100%	✓	✓=terdeteksi ✗= tidak terdeteksi
2	Saya Mahasiswa Muhammadiyah Parepare	1 m	100%	100%	✓	
3	Ujian dilaksanakan di ruang sidang besok	2 m	100%	100%	✓	
4	Penelitian membutuhkan ketekunan yang Tinggi	2,5 m	100%	100%	✓	
5	Kesabaran adalah obat terbaik	3 m	100%	100%	✓	

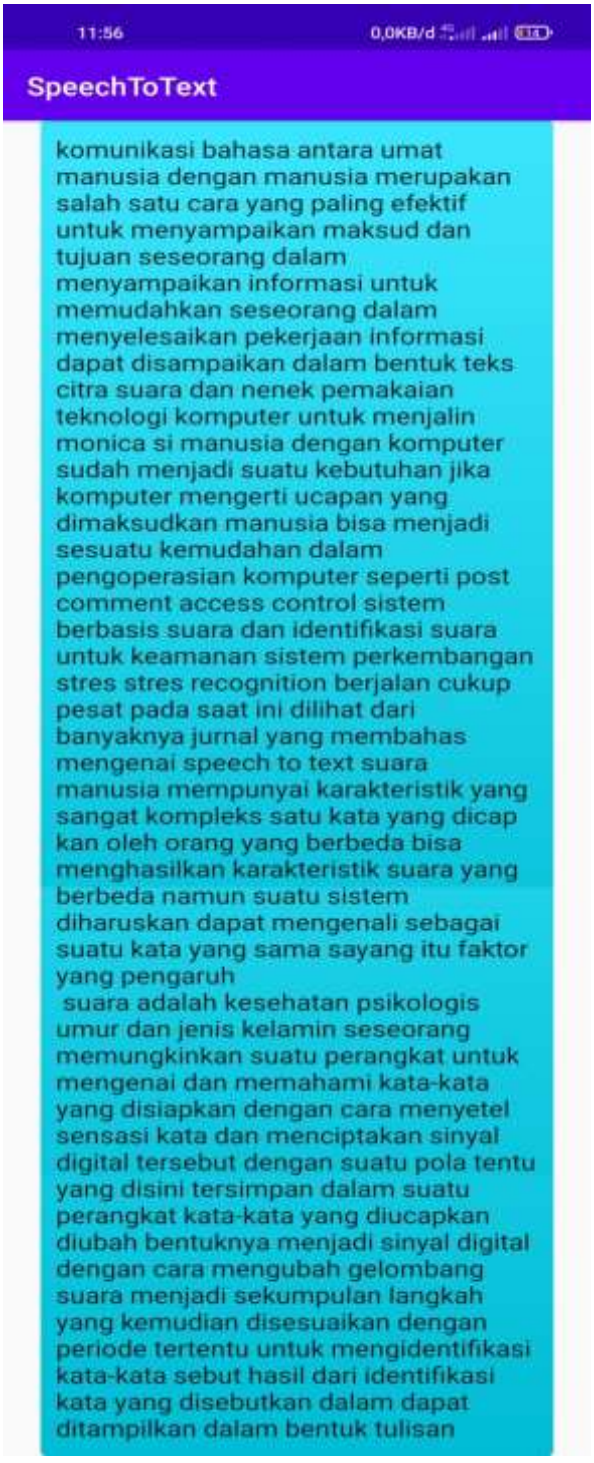
No	KATA	Jarak	Pelan	Cepat	Proses	KET
6	Fokus terhadap tujuan untuk medapatkan harapan terbaik	4 m	100%	100%	✓	✓=terdeteksi ×=tidak terdeteksi
7	Kuliah di pindah di gedung fakultas	4,5 m	100%	100%	✓	
8	Sejarah adalah milik para penguasa	5 m	100%	100%	✓	
9	Rancang bangun aplikasi berbasis android	5,5 m	100%	100%	✓	
10	Saya bersyukur kepada Allah	6 m	100%	100%	✓	
11	Sekolah di parepare libur pada hari sabtu	6,5 m	100%	100%	✓	
12	Sepak bola Indonesia berkembang dengan pesat	7 m	85%	85%	✓	
13	Liburan bersama keluarga ke pantai	7,5 m	70%	70%	✓	
14	Secangkir kopi memberiku inspirasi	8 m	50%	50%	✓	
15	Saya Tersesat dan Tidak Tahu Jalan Pulang	9 m	0%	0%	×	

Tabel 4.15 Uji coba aplikasi dengan suara *speaker* apakah terdeteksi

No	KATA	Jarak	Akurasi	Proses	KET
1	Saya Sekarang di Depan Fakultas Teknik	1 m	10%	✓	✓=terdeteksi ×= tidak terdeteksi
2	Saya Mahasiswa Muhammadiyah Parepare	1,5 m	10%	✓	
3	Ujian dilaksanakan di ruang sidang besok	2 m	20%	✓	

No	KATA	Jarak	Akurasi	Proses	KET
4	Penelitian membutuhkan ketekunan yang Tinggi	2,5 m	25%	✓	✓=terdeteksi ✗= tidak terdeteksi
5	Kesabaran adalah obat terbaik dalam kesulitan	3 m	35%	✓	
6	Fokus terhadap tujuan untuk medapatkan harapan terbaik	3,5 m	50%	✓	
7	Kuliah di pindah di gedung fakultas	4 m	65%	✓	
8	Sejarah adalah milik para penguasa	4,5 m	75%	✓	
9	Rancang bangun aplikasi berbasis android	5 m	80%	✓	
10	Saya bersyukur kepada Allah	5,5 m	95%	✓	
11	Sekolah di parepare libur pada hari sabtu	6 m	100%	✓	
12	Sepak bola Indonesia berkembang dengan pesat	6,5 m	100%	✓	
13	Liburan bersama keluarga ke pantai	7 m	100%	✓	
14	Secangkir kopi memberiku inspirasi	7,5 m	100%	✓	
15	Saya Tersesat dan Tidak Tahu Jalan Pulang	8 m	100%	✓	

Dari hasil pengujian pada tabel 4.15 yang merupakan uji coba menggunakan *speaker* dengan jarak tertentu, didapatkan hasil bahwa aplikasi mampu mendeteksi dengan tingkat akurasi yang berbeda-beda.



Gambar 4.16 pengujian jumlah maksimal kata

Tidak ada batasan jumlah kata, selama suara terdeteksi maka hasil *(output)* akan muncul pada layar

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba perangkat lunak yang dilakukan, penulis mengambil kesimpulan dari uji coba aplikasi *voice to text* yang dibangun antara lain :

1. Aplikasi *voice to text* ini bekerja dengan memanfaatkan suara sebagai inputan yang akan diproses sehingga *user* tidak perlu mengetikkan kata yang akan dikirimkan.
2. Aplikasi *voice to text* juga akan menerjemehkan hasil konversi suara ke teks
3. Setelah dilakukan uji coba dengan menggunakan 15 kata dengan desibel 80 dan jarak beragam mulai dari 30 cm sampai dengan 10 m didapatkan hasil *output* yaitu pada jarak pembacaan 9 m suara tidak dapat terdeteksi lagi.
4. Setelah dilakukan uji coba dengan menggunakan 25 kata yang berbeda didapatkan hasil *output* yang sama dengan persentase 90%.
5. Uji coba sistem dengan menggunakan 25 *frase* yang berbeda didapatkan *output*yang sesuai dengan persentase 90%.
6. Uji coba sistem dengan menggunakan 25 kalimat yang berbeda didapatkan *output* yang sesuai dengan persentase 90%.
7. Uji coba menggunakan *speaker* dengan jarak tertentu, didapatkan hasil bahwa aplikasi mampu mendeteksi suara dengan tingkat akurasi yang berbeda-beda.

B. Saran

Pada penelitian ini peulis menyadari bahwa masih ada keterangan kekurangan yang sangat di perlukan perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Perlu diadakan penelitian lanjutan untuk mendapatkan tingkat akurasi yang lebih baik
2. Kedepannya penulis berharap agar sistem yang dihasilkan pada penelitian ini dapat dikembangkan lagi.

DAFTAR PUSTAKA

Adi nugroho 2017 dalam buku “Rekayasa Perangkat Lunak Beriorentasi Objek” Penerbit ANDI

Eko Widayanto, Aplikasi Speech To Text Berbahasa Indonesia Menggunakan Mel-Frequency Cepstral Coefficient Dan Hidden Markov Model, Universitas Diponegoro Semarang 2020

G.P.S. Prasanthi, K. Sirisha, G. Ramya3,B. Padma, 2016. Speech to Text Coersion Using

HMM.Internasional Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 5, Issue 3, March 2016

Hossein Amerkashi,2015, Absolute App Inventor 2: Android Programming for all ages Kindle Edition,Kindle Store

Irawan. 2018. *Membuat Alikasi Android untuk Orang Awam*. Palembang: Maxkom.

Jay A. Kreibich 2018 Pengenalan SQLite Penerbit ANDI

Jugianto, 2018. Aplikasi Pada Sebuah Komputer. Penerbit. Andi

Kamriani,F. & Roy,K.,2015,App Inentor 2 Essentials A Step- By-Step Introductory Guide to mobile app development with app inventor 2,Birmingham:Packt

Nurpadila Muslimin, 2019, Sistem Pembaca Pesan Text To Speech Berbasis Android Menggunakan Framework Google Api, Universitas Hasanuddin Makassar

Shalahuddin 2008, *Pengenalan Pemrograman java* Penerbit Andi

Sharon, Jacob, Israel. 2019. *Speech Processing In Modern Communication*. Germany: Springer. Ebook.

Shivam Sharma, 2015. Speech Recognition with Hidden Markov Model: A Review.international Journal of Scientific & Engineering Research, Volume 6, Issue 11, November-2015.

Uswatun Hasanah, Aplikasi Dongeng Berbasis Text To Speech Untuk Platform Android, , Politeknik Negeri Bengkalis 2018