PENGEMBANGAN APLIKASI *VOICE CALL* DENGAN *MANAGEMENT* KONTAK BERBASIS SUARA BAGI PENGGUNA TUNA NETRA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Guidita Suryani
NIM: 155150201111233

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

PENGEMBANGAN APLIKASI VOICE CALL DENGAN MANAGEMENT KONTAK BERBASIS SUARA BAGI PENGGUNA TUNA NETRA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Guidita Suryani
NIM: 155150201111233

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing 2
Dr. Eng. Herman Tolle, S.T, M.T	Tri Afirianto, S.T, M.T
NIP: 19740823 200012 1 001	NIK: 201309 851213 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Januari 2019

Guidita Suryani

NIM: 155150201111233

PRAKATA

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah kepada penulis sehingga penelitian yang berjudul "Pengembangan Aplikasi *Voice Call* Dengan *Management* Kontak Berbasis Suara Bagi Pengguna Tuna Netra" dapat terselesaikan. Tanpa adanya bantuan dari beberapa pihak laporan ini tidak akan mampu terselesaikan. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terimakasih sebesar-besarnya kepada:

1. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku Pembimbing I yang telah dengan sabar membimbing dan mengarahkan penulis sehingga laporan ini dapat terselesaikan.
2. Bapak Tri Afirianto, S.T, M.T selaku Pembimbing II yang telah dengan sabar membimbing dan mengarahkan penulis sehingga laporan ini dapat terselesaikan.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika.
4. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika.
5. Bapak Nanang Yudi Setiawan, ST., M.Kom. selaku dosen Penasihat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
6. Seluruh keluarga besar penulis atas dukungan dan doa yang tak henti-hentinya diberikan sehingga penulis mampu menyelesaikan laporan ini.
7. Seluruh civitas academica Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penulisan laporan ini masih jauh dari kata sempurna dan masih banyak kekurangan, sehingga saran dan kritik yang diberikan sangat penulis harapkan. Akhir kata penulis berharap agar laporan ini juga dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 3 Januari 2019

Penulis

ABSTRAK

Guidita Suryani, Pengembangan Aplikasi *Voice Call* dengan *Management* Kontak Berbasis Suara Bagi Pengguna Tuna Netra

Pembimbing: Dr.Eng.HermanTolle, S.T, M.T dan Tri Afirianto, S.T. M.T

Dengan adanya *smartphone* memungkinkan setiap orang untuk berkomunikasi baik verbal maupun non-verbal, dengan sangat mudah dan dapat dilakukan kapan pun dan di mana pun. Namun, bagi penyandang tuna netra yang memiliki keterbatasan dalam konteks visual merasa sulit untuk mengoperasikan *smartphone* yang kebanyakan menggunakan teknologi layar sentuh dengan layar datar. Untuk mengatasi masalah yang dialami oleh penyandang tunanetra dalam berkomunikasi dengan menggunakan *smartphone*, maka melalui penelitian ini dilakukan sebuah pengembangan aplikasi komunikasi verbal dalam bentuk panggilan telepon/*voice call* yang juga mampu menangani manajemen kontak telepon seperti menambah, mengubah data kontak, dan menghapus kontak.

Aplikasi ini juga dirancang secara khusus dengan menggunakan *voice interaction* yang memanfaatkan Google API, sehingga pengguna dapat melakukan kontrol terhadap aplikasi dengan hanya menggunakan suara dan umpan balik yang diterima juga dalam bentuk suara. Dalam perancangan aplikasi ini juga menggunakan pendekatan *Human Centered Design* (HCD), yang melibatkan calon pengguna dalam setiap proses perancangan sehingga hasil akhir dari aplikasi ini sudah memenuhi kebutuhan pengguna. Berdasarkan pengujian *usability* menggunakan *System Usability Scale* (SUS) yang memperoleh rata-rata nilai sebesar 86 yang menunjukkan kualitas aplikasi sudah berada pada *rating* "*excelent*" dan sudah dirasa mudah untuk digunakan oleh pengguna. Selain itu kualitas dari layanan *voice interaction* yang disediakan pada aplikasi juga sudah teruji memiliki kualitas yang baik (*Good*), yang ditunjukkan berdasarkan nilai rata-rata 4,2 yang diperoleh dari pengujian Mean Opinion Score (MOS) *testing*.

Kata kunci: tuna netra, *voice call, voice interaction, assistive technology, Human Centered Design* (HCD)*, System Usability Scale* (SUS)*, Mean Opinion Score* (MOS) *testing*

ABSTRACT

Guidita Suryani, ***The development of voice call applications with voice-based contact management for people with visual impairment***

***Supervisors*: Dr.Eng.HermanTolle, S.T, M.T and Tri Afirianto, S.T. M.T**

With the existence of a smartphone allows everyone to communicate both verbal and non-verbal easier and can be done anytime and anywhere. However, people with visual impairments who have problems in a visual context feels difficult to operate a smartphone with a touch and flat screen. To overcome the problems experienced by visual impairments people in communicating using a smartphone, this research carried out a development of verbal communication applications for a voice calls that are also able to handle telephone contact management such as adding, changing contact data, and deleting a contact.

This application is also designed specifically by using voice interaction that utilizes the Google API and allowing users to control the application by using voice and the feedback is also by voice. This application is design by using Human Centered Design (HCD) approach, which always involved the user in the development process. Based on usability testing using System Usability Scale (SUS), obtaining an average value of 86 which indicates the quality of the application is already at an "excellent" rating and is already considered easy to use by users. The quality of the voice interaction service provided in the application has also been tested to have good quality, which is shown based on the average value of 4.2 obtained from the Mean Opinion Score (MOS) testing.

Keywords: *visually impaired*, *voice call, voice interaction, assistive technology, Human Centered Design* (HCD)*, System Usability Scale* (SUS)*, Mean Opinion Score* (MOS) *testing*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar belakang

Penyandang tunanetra merupakan seseorang yang kehilangan penglihatan secara signifikan di mana dapat diperbaiki menggunakan alat bantu penglihatan seperti kacamata, lensa kontak, atau bedah (Martz, 2018). Ada tiga kategori penderita tunanetra berdasarkan kemampuan data penglihatan yang mereka miliki yaitu tunanetra berat (*totally blind*), tunanetra sedang (*partially sighted*), dan tunanetra ringan (*low vision*) (Colenbrander, 2007). Penderita tunanetra berat adalah mereka yang kehilangan seluruh kemampuannya untuk melihat dan membaca, sedangkan penderita tunanetra sedang dan ringan adalah mereka yang masih mampu melihat dan membaca meskipun terdapat banyak hambatan sehingga membutuhkan alat-alat khusus. Berdasarkan data perhitungan yang dilakukan oleh World Health Organization (WHO) diestimasikan ada sekitar 253 juta orang dengan keterbatasan penglihatan, di antaranya ada 36 juta orang yang buta (tunanetra) dan 217 juta orang yang memiliki keterbatasan penglihatan di mana 81% di antaranya adalah mereka yang berusia 50 tahun keatas (WHO, 2017), sedangkan di Indonesia sendiri penderita Tuna Netra mencapai 3.750.000 orang atau sekitar 1,5 persen dari jumlah total penduduk 250 juta jiwa. Jumlah tersebut meliputi buta maupun lemah penglihatan (Indrawati, 2016).

Komunikasi menjadi suatu hal yang sangat penting bagi kehidupan manusia, selain itu kodrat manusia adalah sebagai makhluk sosial yang artinya tidak dapat hidup tanpa berhubungan dengan sesama. Hal ini berlaku juga bagi penyandang disabilitas terutama tuna netra, keterbatasan yang mereka alami adalah dari segi penglihatan sehingga tidak menjadi hambatan bagi mereka untuk berkomunikasi secara verbal atau lisan seperti orang normal pada umumnya. Bagi penyandang tunanetra, komunikasi dapat memberi mereka wawasan tentang dunia sekitar, meskipun mereka tidak dapat melihat. Dengan berbicara dengan orang lain, mereka dapat belajar banyak dari pengalaman mereka (Garibay, et al., 2014).

Seiring dengan perkembangan teknologi, cara komunikasi yang dimiliki oleh manusia juga ikut mengalami perkembangan. Jika pada awalnya proses komunikasi yang dilakukan terbatas pada ruang dan waktu, maka manusia mulai menciptakan dan mengebangkan suatu alat komunikasi mulai dari telepon umum hinga alat komunikasi modern saat ini yaitu *smartphone. Smartphone* telah menjadi sarana bagi individu untuk berkomunikasi dengan cepat (Gladden, 2018). Bahkan menurut data terakhir dari Badan Pusat Statistik (BPS) pada tahun 2016 persentase penduduk yang memiliki atau menguasai Telepon Seluler di Indonesia mencapai 58,30% dari total seluruh penduduk (BPS, 2018). Lembaga riset *digital marketing* Emarketer telah memperkirakan pada tahun 2018 jumlah pengguna *smartphone* di Indonesia lebih dari 100 juta orang. Hal ini menjadikan Indonesia menjadi negara pengguna aktif *smartphone* terbesar keempat di dunia setelah Cina, India, dan Amerika (Kominfo, 2015).

Namun, sangat disayangkan kemajuan teknologi pada *smartphone* tidak terlalu mempertimbangkan fasilitas bagi orang-orang yang memiliki keterbatasan visual. Kebanyakan *smartphone* dikembangkan dengan permukaan layar sentuh yang datar dan untuk mengaksesnya sangat dibutuhkan kemampuan visual. Studi tentang aksesibilitas pada ponsel untuk orang tua dan penyandang cacat menyimpulkan bahwa ponsel memang bukan dirancang secara khusus untuk mereka (Johnsen, et al., 2010).

Bagi mereka yang memiliki gangguan penglihatan, sangat sulit untuk melakukan panggilan dan mencari kontak serta menambahkan kontak baru pada buku kontak bawaan dari *smartphone* mereka. Penelitian yang berjudul "*Audio phonebook for the blind people*" yang dilakukan oleh (Popović & Pale, 2016) mengemukakan sebuah ide membuat sebuah *hardware* sebagai pengganti sebuah kontak telepon (*phonebook*) yang dapat terhubung dengan perangkat bergerak dan didapatkan hasil bahwa perangkat tersebut telah memecahkan masalah utama yang dimiliki penderita tunanetra ketika melakukan panggilan telepon seperti menyimpan kontak dalam jumlah yang besar, dan menghilangkan resiko pemanggilan nomor yang salah. Yang membedakan dengan penelitian pada skripsi ini adalah tidak memakai *hardware* tambahan. Selain itu, terdapat pula penelitian yang berjudul "*A new Android application for blind and visually impaired people*" yang dilakukan oleh (Kardyś, et al., 2016) mengemukakan sebuah ide membuat sebuah aplikasi berbasis Android dalam penggunaan *smartphone* dalam melakukan telepon, mengirim dan menerima pesan teks serta opsi tambahan seperti pemosisian atau pemantauan baterai melalui perintah suara, dan telah didapatkan hasil aplikasi tersebut dapat digunakan dengan mudah bagi penderita tunanetra karena perintah untuk menjalankan fiturnya hanya berdasarkan suara. Namun, yang membedakan dengan penelitian pada skripsi ini adalah fokusnya pada komunikasi telepon tidak hanya melakukan panggilan namun juga dapat melakukan manajemen kontak dan pengimplementasiannya secara *hybrid* sehingga lebih mudah ketika ingin dikembangkan pada lintas *platform.*

Berdasarkan masalah yang telah dipaparkan sebelumnya maka dalam penelitian pada skripsi ini memiliki tujuan untuk merancang dan mengimplementasi aplikasi *Mobile* yang berguna sebagai kontak telepon yang lebih *User Friendly* bagi pengguna dengan keterbatasan penglihatan (tunanetra) dengan menggunakan *Speech to Text* untuk melakukan panggilan, pencarian, dan untuk melakukan penambahan kontak dengan menginputkan suara, dan *Text to Speech* berupa *output* dalam bentuk suara agar pengguna mengetahui aksi yang mereka lakukan benar, dalam mendesain menggunakan pendekatan metode *Human Centered Design* (HCD) yang melibatkan Manusia/Pengguna dalam proses perancangannya sehingga pada akhirnya akan dihasilkan suatu produk khusus untuk memenuhi kebutuhan tunanetra, pengimplementasiannya pada perangkat bergerak secara *hybrid* yang nantinya akan diuji pada Android yang saat ini memiliki bangsa pasa nomor satu di indonesia yaitu sebesar 52,59% (GlobalStats, 2018). Pengimplementasian menggunakan kerangka kerja Ionic yang merupakan *platform* yang jauh lebih baik untuk digunakan dalam membangun aplikasi secara lintas *platform* tanpa perlu mengeluarkan biaya lebih dalam pengembangannya.

Nantinya sistem juga akan diuji usabilitasnya menggunakan *System Usability Scale* (SUS) dan kualitas dari interaksi suara akan diuji menggunakan *Mean Opinion Score* (MOS) untuk mengetahui *Quality-of-Experience* (QoE) dari aplikasi yang dikembangkan. Diharapkan dengan adanya aplikasi ini dapat membantu pengguna tunanetra dalam melakukan panggilan dan menyimpan kontak dengan lebih mudah bahkan tanpa bantuan orang lain.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dalam penelitian ini dirumuskan permasalahan-permasalahan yang dibahas dalam penelitian ini sebagai berikut.

1. Bagaimana rancangan dan implementasi sistem aplikasi *voice call* dengan manajemen kontak berbasis suara yang dapat mendukung komunikasi *voice call* bagi penyandang tuna netra?
2. Bagaimana tingkat kemudahan pengggunaan sistem aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tuna netra?
3. Bagaimana kinerja Google *Text-to-Speech* dan Google *Speech Recognition* menggunakan modul bahasa Indonesia dalam penerapannya pada aplikasi?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka tujuan dari penelitian ini sebagai berikut :

1. Merancang sistem dengan menggunakan pendekatan berorientasi objek dan mengimplementasikan sistem dengan menggunakan Google *Speech Recognition* API sebagai modul pemroses *Speech-To-Text* dan Google API sebagai modul pemroses *Text-to-Speech*.
2. Mengukur tingkat kemudahan sistem menggunakan menggunakan *usability testing*, serta memastikan sistem memiliki tingkat kemudahan penggunaan yang baik.
3. Mengukur kinerja Google *Text-to-Speech* dan Google *Speech Recognition* menggunakan modul bahasa Indonesia dalam penerapannya pada aplikasi menggunakan *Mean Opinion Score* (MOS) *testing*.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut.

1. Manfaat bagi pengembangan ilmu pengetahuan
   Penelitian ini diharapkan mampu memberikan kontribusi terhadap pengembangan aplikasi kontak telepon yang dapat memudahkan penggunaan *smartphone* bagi penyandang tunanetra dalam melakukan panggilan dan manajemen kontak, khususnya mengenai penerapan Google *Speech Recognition* API sebagai pemroses *Speech-To-Text* dan Google *Text-to-Speech* menggunakan API Google.

2. Manfaat bagi penulis
   Penelitian ini dapat bermanfaat membantu penulis memahami efisiensi dari penerapan Google *Speech Recognition* API sebagai pemroses *Speech-To-Text* dan Google *Text-to-Speech* dalam aplikasi *voice call* dengan manajemen kontak telepon ketika digunakan oleh penyandang tuna netra.
3. Manfaat bagi pengguna
   Penelitian ini dapat bermanfaat membantu pengguna khususnya pengguna yang memiliki keterbatasan penglihatan baik sebagian maupun total dalam memudahkan melakukan panggilan dan manajemen kontak telepon pada *smartphone* milik mereka.

1.5 Batasan masalah

Agar penelitian ini dapat terfokus dan mengindari pelebaran topik, maka berdasarkan rumusan masalah yang telah dipaparkan, batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Modul *speech-to-text* menggunakan Google *Speech Recognition* API Bahasa Indonesia.
2. Modul *text-to-speech* menggunakan Google *Text-to-Speech* Bahasa Indonesia.
3. Pengimplementasian dan pengujian sistem dalam penelitian ini adalah pada sistem operasi perangkat *mobile* Android.
4. Sistem tidak dapat digunakan dalam keadaan *offline* atau tidak tehubung dalam jaringan internet.

1.6 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam laporan penelitian ini adalah sebagai berikut.

BAB I: Pendahuluan

Bab ini berisikan hal-hal yang mendasari latar belakang dilakukannya penelitian ini, memaparkan rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan yang digunakan dalam penulisan skripsi ini.

BAB II: Landasan Kepustakaan

Bab ini berisikan penjelasan mengenai dasar teori tentang teknologi yang digunakan dalam penelitian ini dan referensi secara luas serta informasi yang dibutuhkan dalam pengembangan, perancangan dan implementasi pada skripsi ini. Pembahasan pada bagian ini meliputi metode, teori, konsep yang diperoleh dari literatur ilmiah hingga penelitian-penelitian terkait sebelumnya yang didapatkan dari sumber yang terpercaya.

BAB III: Metodologi

Bab ini berisikan metode dan langkah-langkah kerja yang digunakan sebagai acuan dalam proses perancangan, analisis kebutuhan, dan implementasi yang diteliti dalam skripsi ini. Semua proses pada bagian ini nantinya akan mengarahkan penelitian ke tahap akhir untuk mencapai solusi dan jawaban dari permasalahan topik yang diangkat.

BAB IV: Analisis Kebutuhan Sistem

Bab ini berisikan analisis kebutuhan sistem yang dibangun dan juga menjelaskan mengenai alur proses, kebutuhan serta fitur-fitur apa saja yang nantinya akan ada di dalam sistem.

BAB V: Perancangan Sistem

Bab ini berisikan mengenai perancangan sistem yang menjadi sebuah jawaban atau solusi atas maslah yang diangkat dalam penelitian ini yang ditinjau dari sudut pandang pemodelannya, serta akan menjadi acuan untuk melakukan implementasi sistem.

BAB VI: Implementasi Sistem

Bab ini berisikan mengenai proses implementasi dari metode yang digunakan ke dalam sistem. Setelah dilakukan perancangan dan analisis pada bab sebelumnya maka pada bab ini akan membahas mengenai proses implementasi dari sistem yang nantinya bertujuan untuk menghasilkan aplikasi yang diharapkan mampu menjadi solusi atau jawaban dari permasalahan yang dibahas pada skripsi ini.

BAB VII: Pengujian Sistem

Bab ini membahas tentang prosedur dan metode yang nantinya digunakan untuk memastikan apakah hasil implementasi yang diperoleh dapat menjadi solusi terhadap permasalahan yang dibahas dalam penelitian ini.

Bab VIII: Kesimpulan dan Saran

Bab ini berisikan kesimpulan dan saran yang diperoleh dari penelitian yang telah dilakukan. Kesimpulan berupa penjelasan bahwa aplikasi yang dikembangkan berhasil atau tidak dalam menyelesaikan permasalahan yang telah dirumuskan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisikan uraian dan pembahasan mengenai landasan kepustakaan yang mencakup dasar teori. Pada kajian pustaka terdiri dari pembahasan mengenai penelitian yang telah dilakukan dan yang diusulkan sebagai bahan untuk penelitian berikutnya. Dasar teori membahas mengenai teori yang diperlukan untuk perancangan sistem dan teori untuk mencari keterkaitan antara objek dengan penggunaan sistem.

2.1 Kajian Pustaka

Untuk mendukung penelitian yang akan dilakukan, penulis mengumpulkan beberapa penelitian yang telah dilakukan sebelumnya mengenai kontak telepon (*phonebook*) yang dirancang khusus untuk penderita tuna netra. Penelitian tersebut antara lain:

Pada penelitian yang berjudul "*Audio phonebook for the blind people*" yang dilakukan oleh (Popović & Pale, 2016) mengemukakan sebuah ide untuk membuat sebuah *hardware* sebagai pengganti sebuah kontak telepon (*phonebook*) yang dapat terhubung dengan ponsel di mana pada *hardware* tersebut dapat menyimpan nomor telepon dengan menggunakan tiga buah tombol. Tombol-tombol tersebut akan diaktifkan secara bergilir, pertama pengguna akan menyebutkan nama kontak, lalu setelah nama kontak yang dicari telah sesuai dan ditemukan maka pengguna perlu menekan sebuah tombol *"dial"* dan sistem akan melakukan panggilan. Perangkat ini memungkinkan penderita tunanetra lebih mudah dan lebih cepat menangani panggilan, dan untuk melakukan penyimpanan nomor telepon pengguna tersebut cukup memasukkan satu kali nomor telepon dengan menggunakan sebuah tombol untuk memulai aksi dan sebuah *knob* berputar sebagai masukkan angka, dan *feedback* yang diberikan adalah dalam bentuk suara. Dengan menggunakan Arduino *device* maka pengguna akan terhubung secara paralel dengan ponsel mereka.

Kelebihan dari penelitian ini adalah dengan pembuatan *hardware* yang sederhana dan pengoperasiannya mudah dilakukan, serta telah di uji kepada orang-orang dengan keterbatasan penglihatan dan mendapat akurasi yang cukup baik sekitar 73% dari pengguna yang diuji langsung dapat mengoperasikan *audio phonebook* baik dalam melakukan panggilan, menyimpan kontak dan mengedit kontak. Kekurangan dari penelitian ini adalah ketergantungan terhadap *hardware* di mana hanya dapat dioperasikan ketika *hardware*nya tersedia dan apabila terjadi kerusakan belum ada penanggulangan lebih lanjut terhadap hal tersebut.

Persamaan penelitian ini dengan penelitian ini dengan yang akan dilakukan adalah pengembangan kontak telepon berbasis audio untuk memudahkan pengguna tuna netra dalam melakukan panggilan telepon dan menyimpan kontak telepon. Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah mengembangkan aplikasi bukan *hardware* sehingga dapat langsung diimplementasikan pada *smartphone*.

Dalam penelitian yang berjudul "*A new Android application for blind and visually impaired people*" yang dilakukan oleh (Kardyś, et al., 2016) mengemukakan sebuah ide membuat sebuah aplikasi berbasis Android untuk membantu penderita gangguan penglihatan penuh (tuna netra) maupun penderita gangguan penglihatan sebagian dalam penggunaan *smartphone* dalam melakukan telepon, mengirim dan menerima pesan teks dengan memanfaatkan "*Phone book*" pada *smartphone* serta penambahan opsi lain seperti pemosisian atau pemantauan baterai melalui perintah suara. Pertama-tama pengguna akan menyebutkan perintahnya melalui suara, pada aplikasi ini menggunakan modul *Google Speech Recognition* sebagai pemrosesan masukkan suaranya. Modul pengenalan suara mengembalikan aliran karakter yang sesuai dengan kata yang dikenal, dan mengembalikannya ke smartphone dengan cara yang sama. Setelah itu, program akan memeriksa apakah kata-kata yang diterima merupakan salah satu dari perintah yang tersedia. Jika tidak, maka pengguna akan mendengar kembalian berupa suara "*There is not such command*" atau apabila menerima perintah yang tepat, maka fungsi akan dijalankan sesuai dengan perintah fungsi dalam aplikasi. Untuk *feedback* ke pengguna berupa suara menggunakan modul *Text to Speech* dengan *Ivona Speech Synthesizer.*

Kelebihan dari penelitian ini adalah berdasarkan pengujian yang dilakukan pada Pusat Pendidikan Khusus untuk Anak Tunanetra di Owińska aplikasi ini sudah dapat membantu mereka dalam dengan banyak perintah suara yang telah ditentukan, banyak kegiatan dapat dilakukan termasuk membuat panggilan, mengirim dan menerima pesan teks, menggunakan buku telepon dengan mudah, menentukan posisi pengguna, memperoleh informasi tentang waktu sekarang, dan mengetahui tingkat baterai. Kekurangan dari penelitian ini adalah untuk melakukan panggilan dan mengirim pesan teks hanya dapat dilakukan pada kontak yang sudah terdaftar pada *smartphone* milik pengguna secara manual.

Persamaan penelitian ini dengan yang dilakukan adalah dalam penggunaan *Speech Recognition* dalam penerimaan masukkan berupa suara dari pengguna, dan *feedback* menggunakan *Text to Speech* berupa suara, dan juga peggunaan aplikasi dapat digunakan untuk melakukan panggilan telepon berdasarkan *contact list* yang sudah ada pada aplikasi tersebut. Perbedaan penelitian ini dengan yang akan dilakukan adalah dalam penelitian ini aplikasi tersebut fungsionalitasnya selain dapat melakukan panggilan, juga dapat melakukan pengiriman pesan singkat, dan mengetahui posisi pengguna serta mengetahui status baterai dari ponsel pengguna, sedangkan penelitian yang akan dilakukan terbatas pada pengoperasian panggilan dan manajemen kontak seperti menambahkan kontak, mencari kontak dan mengedit kontak.

Selain itu terdapat pula penelitian sejenis yang juga memanfaatkan *voice intercation* yang digunakan untuk melakukan pengiriman pesan teks (*messaging*) yang berjudul "Rancang Bangun Aplikasi *Messaging* Berbasis *Voice Interaction* Bagi Penderita Tunanetra Pada Sistem Operasi Android" yang dilakukan oleh (Justicia, 2017). Yang mana pada penelitian ini memanfaatkan Android *Speech Recognizer* sebagai pemrosesan *speech recognition*. Pada penelitian ini membuat

sebuah aplikasi *messaging* bagi pengguna tuna netra yang dapat mengirimkan pesan baru, membaca pesan yang diterima dan menghapus pesan. Semua data pesan disimpan pada sebuah database NoSQL menggunakan Firebase. Berdasarkan pengujian Mean Opinion Score (MOS) *testing* yang dilakukan terhadap 10 responden didapatkan hasil sebesar 3,7 yang menandakan bawa rata-rata pengguna menyatakan interaksi suara yang diberikan aplikasi ini memiliki kualitas yang baik.

Persamaan penelitian ini dengan penelitian yang akan dilakukan adalah penggunaan *voice interaction* dan target penelitian adalah pada pengguna tuna netra. Sedangkan perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah pada penelitian ini berfokus dalam membuat aplikasi pengiriman pesan teks (*messaging*) sedangkan pada penelitian yang akan dilakukan berfokus pada melakukan panggilan suara (*voice call*) berdasarkan kontak telepon. Selain itu pada penelitian ini juga menggunakan basis data terpisah untuk menyimpan data akun dan data pesan, sedangkan pada penelitian yang akan dilakukan tidak menggunakan basis data terpisah melainkan langsung menggunakan basis data lokal untuk menyimpan dan mengakses kontak telepon.

Tabel 2.1 Perbedaan dan Persamaan Penelitian Dengan Penelitian Sebelumnya

No	Judul	Penulis dan Tahun	Perbandingan	
			Persamaan	Perbedaan
1	*Audio phonebook for the blind people*	Goran Popović, Una Pale (2016)	Pengembangan kontak telepon berbasis audio untuk memudahkan pengguna tuna netra dalam melakukan panggilan telepon dan menyimpan kontak telepon.	Pada penelitian ini dikembangkan aplikasi berbasis Android bukan *hardware* sehingga dapat langsung diimplementasikan pada *smartphone*.
2	*A new Android application for blind and visually impaired people*	Piotr Kardyś, Adam Dąbrowski, Marcin Iwanowski, Damian Huderek (2016)	Penggunaan *Speech Recognition* dalam penerimaan masukkan berupa suara dari pengguna, dan feedback menggunakan *Text to Speech* berupa suara, dan juga peggunaan aplikasi	Pada penelitian ini aplikasi tersebut fungsionalitasnya selain dapat melakukan panggilan, juga dapat melakukan pengiriman pesan singkat, dan mengetahui posisi pengguna serta mengetahui status

			dapat digunakan untuk melakukan panggilan telepon berdasarkan *contact list* yang sudah ada pada aplikasi.	baterai dari ponsel pengguna, serta pengimplementasian aplikasi ini adalah *native* sedangkan penelitian yang akan dilakukan terbatas pada pengoperasian panggilan dan manajemen kontak seperti menambahkan kontak, dan mencari kontak dan pengimplementasian aplikasi ini *hybrid*. Serta peneliti menggunakan modul TTS Google API, sedangkan pada penelitian ini menggunakan modul *Ivona Speech Synthesizer.*
3	Rancang Bangun Aplikasi *Messaging* Berbasis *Voice Interaction* Bagi Penderita Tunanetra Pada Sistem Operasi Android	Leo Tiofan Justicia (2017)	Persamaan penelitian ini dengan penelitian yang akan dilakukan adalah penggunaan *voice interaction* dan target penelitian adalah pada pengguna tuna netra	Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah fokus pengembangan aplikasi di mana pada penelitian ini mengembangkan aplikasi *messaging* sedangkan pada penelitian yang akan dilakukan yaitu mengembangkan

				aplikasi voice call, selain itu juga perbedaan penggunaan basis data, di mana pada penelitian ini menggunakan basis data Firebase, sedangkan penelitian yang akan dilakukan tidak menggunakan basis data terpisah.

Pada Tabel 2.1 menjelaskan secara ringkas mengenai perbandingan dari persamaan dan perbedaan dari penelitian yang sudah dilakukan sebelumnya dengan penelitian yang dilakukan pada skripsi ini.

2.2 Aksesibilitas pada Perangkat *Mobile*

Bagi orang-orang yang memiliki keterbatasan dalam penglihatan sepeti penderita tuna netra merasa terbatas dalam penggunaan *smartphone* dikarenakan mereka tidak dapat melihat informasi yang disajikan dalam bentuk teks. Kebanyakan *developer* selain fokus terhadap fungsionalitas juga fokus pada tampilan *User Interface* yang menarik, namun hal ini tidak diperlukan bagi mereka yang memiliki keterbatasan dalam penglihatan. Oleh sebab itu, untuk mendukung *software* yang mendukung pengguna yang memiliki keterbatasan penglihatan, *developer* harus mempertimbangkan aksesibilitas dari fungsionalitas yang dibangun pada Sistem Operasi *Mobile*. Pada Android sendiri telah menyediakan aksesibilitas bagi penderita tuna netra untuk menerima output dalam bentu suara yang diberi nama Talkback. Menurut W3C ada 3 prinsip dalam membangun aksesibilitas (W3C, 2015) yaitu:

1. *Perceivable* (mudah dipahami) dalam kasus aplikasi bagi mereka yang memiliki keterbatasan penglihatan informasi dan komponen antarmuka harus ditampilkan kepada pengguna dengan cara yang dapat mereka rasakan. *Feedback* yang diberikan sebaiknya berupa suara agar lebih mudah dipahami.
2. *Operable* (dapat dioperasikan) bagi mereka yang memiliki keterbatasan penglihatan akan mengalami kesulitan dalam mengakses operasi yang akan mereka lakukan dengan tepat pada *touchscreen* sehingga *developers* sebaiknya menyediakan alternatif dari touch *user interface.*
3. *Understandable* (dapat dimengerti) informasi yang disediakan oleh aplikasi seluler dan pengoperasiannya harus dapat dimengerti oleh penggunanya, apabila pengguna tidak memahami bagaimana isinya maka pengguna tidak

dapat mengendalikan aplikasi tersebut. Sayangnya, banyak aplikasi seluler yang tidak dapat dipahami oleh pengguna tuna netra baik karena sedikit informasi maupun penyajian informasi yang tidak jelas.

4. *Robust* (kuat) ini menyangkut kekuatan konten, maksudnya harus dapat ditafsirkan dengan benar pada berbagai perangkat seluler termasuk teknologi bantu bawaan yang diperlukan oleh aplikasi seperti sensor.

2.3 Google API

Google API merupakan bagian dari *Framework* (kerangka kerja) Google. Pada Google disediakan berbagai API (Application Programming Interface) yang berguna terutama bagi pengembang (*developer*) aplikasi baik web, mobile, maupun desktop dalam memanfaatkan berbagai fitur yang disediakan oleh Google seperti*: Search Engine*, Youtube, *AdSense*, *Translation* dan lain sebagainya. Secara sederhana API dapat diartikan sebagai sebuah kode program yang menjadi antarmuka atau penghubung antar aplikasi atau web yang dibuat dengan seluruh fungsi-fungsi yang dibuat.

Jadi, Google API merupakan kode program yang dapat ditambahkan pada aplikasi untuk memanfaatkan fitur-fitur yang disediakan Google pada aplikasi yang sedang dibangun. Google API dapat dipelajari secara mudah melalui Google Code, developer dapat memilihi layanan yang disediakan oleh google sesuai dengan yang mereka butuhkan pada pengembangan aplikasi mereka. Ada banyak jenis API yang disediakan oleh Google, berikut beberapa di antara API yang disediakan oleh Google.

1. Language API: memanfaatkan fitur translation milik Google
2. Earth API: memanfatkan fitur pada Google Earth
3. Maps API: memanfaatkan fitur pada Google Maps
4. Search API: memanfaatkan fitur pencarian pada Google Search
5. Visualization API: membuat grafik maupun chart dengan Google API
6. YouTube API: memanfaatkan fitur youtube

2.3.1 Google Speech Recognition API

Pada penelitian yang dilakukan oleh Veton Këpuska dan Gamal Bohouta yang berjudul "*Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)*" membandingkan 3 API yang cukup terkemuka untuk *Speech Recognition* dengan menggunakan beberapa rekaman audio yang dipilih dari banyak tempat didapatkan hasil seperti pada Tabel 2.2 :

Tabel 2.2 Perbandingan *Library* Sphinx4, Google API dan Microsoft API

Sumber: (Këpuska & Bohouta, 2017)

File	Sphinx4		Google API		Microsoft API	
	WA	WER	WA	WER	WA	WER
TSX223	0.88	0.25	1.0	0.0	0.88	0.13
TSX293	0.64	0.36	0.82	0.18	1.0	0.0
TSI1894	0.78	0.22	1.0	0.0	0.78	0.22
TSI1400	0.79	0.29	0.93	0.07	1.0	0.0
TSX188	0.33	0.67	1.0	0.0	0.0	0.1
TSI1628	0.42	0.58	0.83	0.17	0.17	0.83
TSX314	0.67	0.33	1.0	0.0	1.0	0.0
DIG001	0.93	0.07	1.0	0.0	1.0	0.0
TSX216	0.89	0.11	1.0	0.0	1.0	0.0
TSX209	0.71	0.29	1.0	0.0	0.71	0.29
TSI1584	0.38	0.62	0.46	0.54	0.46	0.54
TSX371	0.45	0.55	1.0	0.0	0.91	0.09
TSI1373	0.29	0.71	1.0	0.0	0.57	0.43
TSX233	0.71	0.43	0.71	0.14	0.71	0.29
OSE003	0.63	0.5	0.63	0.38	0.63	0.38
AENGM8	0.89	0.11	1.0	0.0	1.0	0.0
AENGF8	0.33	0.67	0.78	0.22	1.0	0.0
AENGF7	0.83	0.17	1.0	0.0	1.0	0.0
AENGM2	0.86	0.14	1.0	0.0	0.86	0.14
Mean	WER: 0.37		WER: 0.09		WER:|0.18	

Dapat dilihat dari Tabel 2.2 dapat disimpulkan bahwa pada Sphinx-4 mencapai 37% WER (*Word Error Rate),* Microsoft API mencapai 18% WER dan Google API mencapai 9% WER. Sehingga dapat disimpulkan bahwa Google API lebih unggul (Këpuska & Bohouta, 2017).

Dengan menggunakan *Google Speech Recognition* API *developer* dapat mengkonversi dalam bentuk ucapan (*speech*) ke dalam teks. Agar dapat menggunakan fitur tersebut, maka *developer* Android dapat menggunakan *interface* dan *class* yang telah disediakan oleh Google API *(package android.speech)* (Sianturi, 2014).

1. *Interfaces*
   RecognitionListener digunakan untuk menerima pemberitahuan dari *SpeechRecognizer* ketika peristiwa pengenalan *(recognition)* terkait terjadi.
2. *Classes*
   Terdapat beberapa *class* yang terdapat pada *package android.speech,* diantaranya:
   a) *RecognitionService*, kelas ini menyediakan kelas dasar untuk implementasi layanan pengenalan.
   b) *RecognitionService.Callback,* kelas ini menerima *callback* dari layanan pengenalan suara dan mengirimkannya ke pengguna.

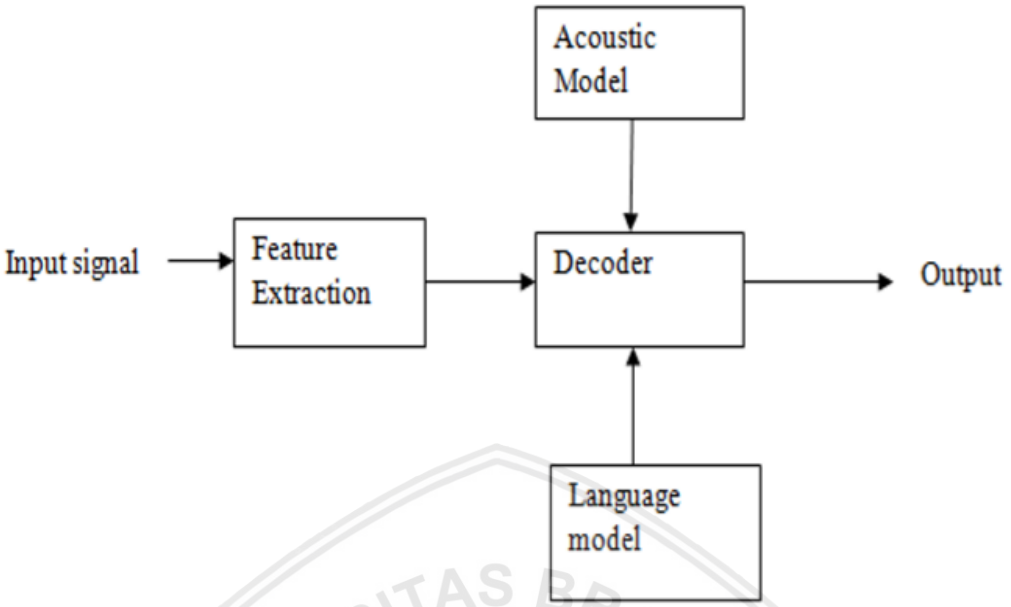
c) *RecognizerIntent,* konstanta untuk mendukung pengenalan suara melalui memulai *Intent.*
d) *RecognizerResultsIntent,* konstanta untuk *intent* yang berkaitan dengan menunjukkan hasil pengenalan suara.

2.3.2 Speech Recognition

Speech Recognition adalah sebuah mesin yang kerjanya berdasarkan pengucapan kata yang diterima lalu dilakukan identifikasi serta memberikan respons untuk reaksi selanjutnya. Pada *Speech Recognition* mampu mengubah sinyal suara yang diterima menjadi sebuah teks atau perintah melalui proses identifikasi dan penalaran. Teknologi ini membutuhkan sebuah komputasi matematika saat mengenali suara yang diucapkan oleh pengguna, sehingga sangat dibutuhkan pengaksesan yang cepat dan langsung *(realtime)* pada pengembangan teknologi ini.

Perkembangan *Speech Recogition* meningkat secara bertahap hingga ditemukannya Hiden Markov Model (HMM) pada awal tahun 1970. HMM merupakan sebuah pendekatan statistik yang paling dominan digunakan selama bertahun-tahun. Teori dasar HMM dikemukakan dalam serangkaian makalah klasik oleh Baum dan teman-temannya yang kemudian diimplementasikan untuk aplikasi pengenalan suara oleh Baker di Carnegie Mellon University (CMU) oleh Jelinek dan teman-temannya di IBM pada tahun 1970-an. Munculnya metode HMM menjadi salah satu pemicu dari pembaruan teknik pengolahan *speech* (ucapan/suara) yang dapat menghasilkan tingkat kesalahan yang seminimal mungkin serta meningkatkan pengklasifikasian pola pada berbagai kondisi yang kurang baik. Penerapan HMM pada pengenalan suara sering digunakan untuk memprediksi suara yang diucapkan berdasarkan berdasarkan pada *dictionary file*,*language model,* dan *acoustic model.* HMM akan memberikan probabilitas berdasarkan kosakata yang akan diucapkan. Probabilitas kemudian dihitung berdasarkan pembobotannya sehingga mempunyai nilai terbaik yang nantinya memungkinkan kata tersebut dikeluarkan berdasarkan persamaan suara. Pencarian probabilitas terbaik dari masing-masing suku kata diselesaikan dengan Algoritma Viterbi (Alauddin, et al., 2017).

Gambar 2.1 menjelaskan mengenai pemodelan pada *speech recognation: Speech-to-text* di mana maksud dari gambar tersebut adalah : sinyal input yang didapat adalah dari masukkan suara (*voice input*) yang dimasukkan oleh user. Selanjutnya, *Fitur Extraction* harus mempertahankan informasi yang penting yang didapatkan dari sinyal input, kemudian mengurangi informasi yang tidak diperlukan pada sebuah *speech*. *Acoustic Model* berisi tentang representasi statistik dari setiap suara berbeda yang membentuk sebuah kata. *Languange Model* memberikan kemungkinan urutan kata dengan cara distribusi probabilitas. *Output* yang diberikan ke komputer adalah dalam bentuk teks.



Gambar 2.1 Pemodelan pada *Speech Recognation : Speech-to-Text*

Sumber: (Hansen, et al., 2010)

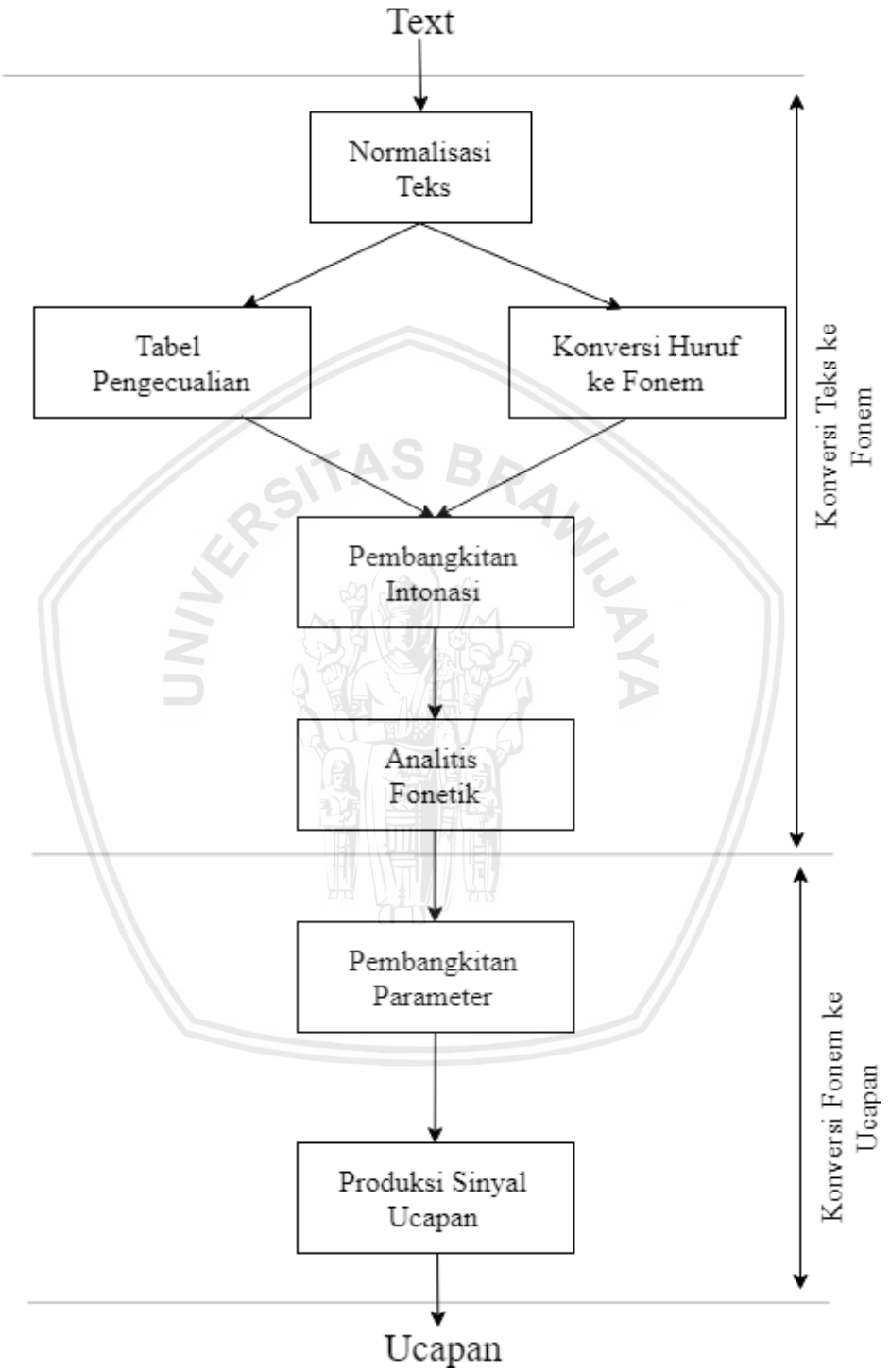
2.3.3 *Text to Speech* (TTS)

Biasanya TTS diartikan sebagai sebuah sistem yang dapat mengubah teks menjadi sebuah ucapan. Ada dua bagian dari TTS yang penting yaitu teks ke fonem (text to phoneme) dan *converter* fonem dapat melakukan konversi dari kalimat dalam suatu *converter* teks menjadi kode fonem ke ucapan (*phoneme to speech)*, pada TTS dapat melakukan konversi dari kalimat dalam suatu bahasa tertentu yang sebelumnya masih dalam bentuk teks yang selanjutnya dapat dirubah ke bentuk fonem, durasi dan pitch menjadi bagian dari rangkaian kode-kode yang akan merepresntasikan pitch dan durasi yang dibuat oleh bagian *text to phoneme*. (Rachma, et al., 2011).

Gambar 2.2 menjelaskan bahwa pada Sistem TTS terdapat dua subsistem dasar:

1. Subsistem yang berfungsi sebagai konverter dari sebuah tulisan dalam bentuk teks ke fonem yang memiliki 2 fungsi utama yaitu: Pertama sebagai pengambil kalimat masukan yang ada pada suatu bahasa tertentu dalam bentuk teks dan merubahnya menjadi tulisan sesuai dengan bunyi yang seharusnya, hal ini biasa disebut dengan normalisasi teks. Kedua sebagai penentu kode fonetik untuk tiap kata serta durasi dan nadanya. Kode fonem merupakan kode yang merepresentasikan satuan dari bunyi yang diucapkan. Pengucapan kata atau kalimat secara simbolik merupakan urutan kode fonem.
2. Subsistem yang berfungsi sebagai konventer dari fonem ke ucapan yang menerima masukkan berupa kode-kode dalam bentuk fonem serta nada dan durasi yang dihasilkan pada subsistem sebelumnya. Berdasarkan dari

kode-kode tersebut maka akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang nantinya akan diucapkan.

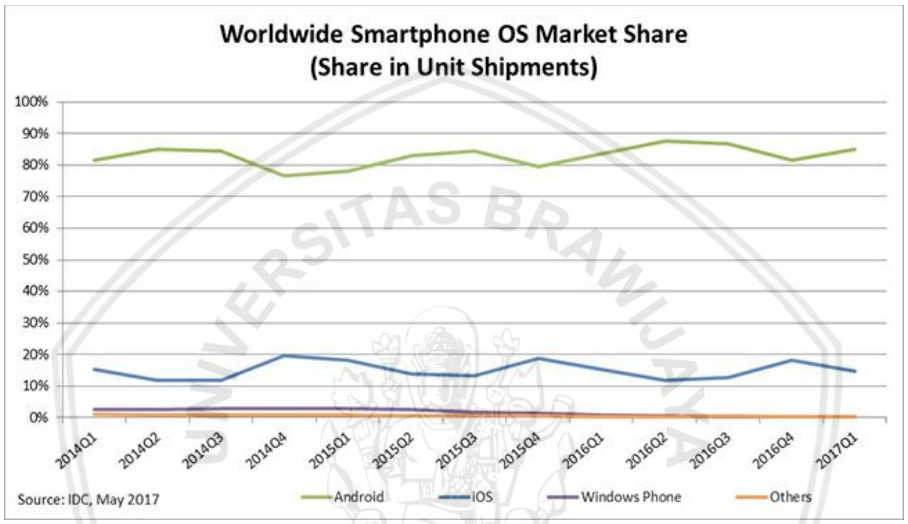


Gambar 2.2 Urutan Proses Konversi dari Teks ke Ucapan

Sumber: (Samsudin & Putra, 2016)

2.4 *Market Share* Sistem Operasi Perangkat Mobile

Sistem Operasi pada perangkat mobile merupakan sebuah *platform* pemrograman yang dikembangkan oleh perusahaan tertentu seperti Android dikembangkan oleh Google dan iOS dikembangkan oleh Apple yang digunakan pada *smartphone* dan perangkat seluler lainnya seperti tablet, smartwatch, dll. Sistem Operasi paling populer dan paling banyak digunakan oleh produsen *smartphone* dan paling diminati oleh masyarakat dapat dilihat dari grafik dibawah ini:



Gambar 2.3 *Market Share* Sistem Operasi Perangkat *Mobile*

Sumber: (IDC, 2015)

Dapat dilihat dari Gambar 2.3 bahwa menurut data yang diperoleh dari website IDC bahwa perkembangan kebutuhan pasar akan sistem operasi perangkat mobile didominasi oleh perangkat mobile android dan iOS.

2.5 Pemrograman Aplikasi Berbasis *Web*

Seiring dengan kemajuan teknologi perangkat keras dan perangkat lunak, kini kita dapat dengan mudah menjumpai peramban *web (web browser)* pada peranti bergerak yang mendukung standar teknologi *web* terkini, bahkan kemampuannya telah mendekati kemampuan peramban *web* konvensional pada komputer personal. Dengan kemampuan tersebut, para pengembang (*developers*) dapat memanfaatkan fitur-fitur modern dan canggih yang dimiliki standar *web* interaktif berbasis *web* pada peranti bergerak (Tolle, et al., 2017).

HTML5 (*Hypertext Markup Language*) versi 5, CSS3 (*Cascading Style Sheet*) versi 3, dan Javascript merupakan teknologi standar yang menjadi dasar dalam pembuatan aplikasi *mobile* berbasis *web modern*. HTML5, CSS3, dan Javascript memiliki peran masing-masing dalam menyusun aplikasi berbasis web. HTML berfungsi dalam membuat keseluruhan struktur dokumen atau konten, CSS3

memiliki fungsi dalam menambahkan konten atau aspek presentasi tampilan pada dokumen HTML seperti gaya, warna, tata letak, jenis huruf, dsb. Sedangkan Javascript berperan dalam menyajikan aspek perlaku dinamis pada dokumen HTML.

2.6 Aplikasi *Hybrid*

Aplikasi *hybrid* merupakan salah satu jenis aplikasi perangkat bergerak yang dikembangkan menggunakan pendekatan pengembangan *web* yang dikombinasikan dengan *Application Protocol Interface* (API) spesifik sehingga menghasilkan aplikasi yang memiliki kemampuan layaknya aplikasi yang dikembangkan secara *native* dan juga dapat dipasangkan pada *smarphone*. Pendekatan pengembangan aplikasi *hybrid* bertujuan untuk memanfaatkan kemudahan membuat aplikasi berbasis *web* yang dapat dijalankan *multiplatform*. Pada umumnya aplikasi *hybrid* juga dikembangkan menggunakan bahas pemrograman berbasis *web* seperti HTM, CSS, dan Javascript (Tolle *et al.*, 2017). Beberapa kelebihan yang dimiliki oleh aplikasi yang dikembangkan secara *hybrid* adalah:

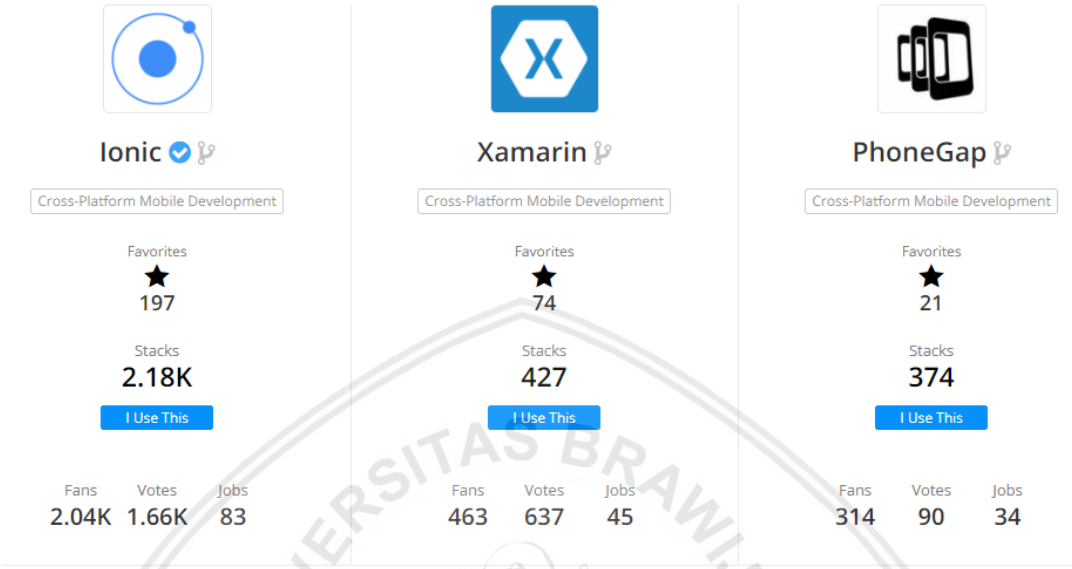
1. Dapat disebarkan (*deploy*) pada berbagai jenis platform dengan menggunakan *source code* (kode program) yang sama tanpa perlu melakukan proses *porting*.
2. Mempunyai akses yang luas terhadap API platform, meskipun tidak seluas hak akses yang dimiliki oleh aplikasi yang dikembangkan secara *native*.
3. Kinerja dalam segi kecepatan dan responsif yang lebih baik dibandingkan aplikasi *web*.

2.7 Framework Ionic

Framework Ionic merupakan sekumpulan dari kerangka kerja teknologi yang dikembangkan guna membantu membangun sebuah aplikasi berbasis mobile hybrid yang cepat, *powerful*, mudah dan juga tetap memiliki tampilan yang menarik. Pada Ionic menggunakan AngularJS sebagai *framework* dengan berbasis web serta menggunakan Cordova untuk membangun aplikasi mobile. Dengan *Framework* ini sangat membantu para *developer* karena dengan kemampuan mereka dalam mengembangkan web dengan bermodal kemampuan dalam menguasai HTML, CSS dan Javascript sudah dapat membuat aplikasi *mobile*. Selain itu *framework* ini juga menggunakan lisensi yang *opensource* sehingga penggunakannya tanpa perlu berbayar dan dapat digunakan oleh siapapun, sehingga para *developer* dapat membuat aplikasi mereka secara gratis atau komersial dengan menggunakan *framework* ionic ini. Dalam ionic memanfaatkan AngularJS untuk mengimplementasi logicnya sehingga dapat menawarkan performa serta respons yang terasa sangat cepat layaknya seperti aplikasi yang dibangun secara native (Rofiq & Putri, 2017).

Menurut situs stackshare.io yang melakukan survei terhadap para pengembang aplikasi *mobile* dilakukan survei untuk membandingkan *framework* yang cukup terkenal dalam mengembangkan aplikasi secara *cross platform* yaitu

framework Ionic, Xamarin dan PhoneGap hasil dari survei tersebut dapat dilihat dalam Gambar 2.4.



Gambar 2.4 Perbandingan *Framework* Ionic, Xamarin dan PhoneGap

Sumber: (Stackshare, 2018)

Dalam Gambar 2.4 menunjukkan perbandingan minat pengguna terhadap 3 *framework* yang cukup populer dalam mengembangkan aplikasi secara *hybrid* didapatkan hasil bahwa *framework* Ionic jauh lebih diminati oleh para pengembang aplikasi dibandingkan dengan *framework* sejenis seperti xamarin dan PhoneGap.

2.8 *Human Centered Design* (HCD)

Human Centered Design (HCD) merupakan sebuah pendekatan desain yang berpusat pada manusia. Dalam pengembangan sistem interaktif yang tujuan utamanya dapat digunakan dan berguna dengan berfokus pada aktivitas-aktivitas yang dapat dilakukan oleh pengguna (ISO, 2010). Berikut ini merupakan aktivitas-aktivitas yang ada pada ISO 9241-210 :

1. Memahami dan menentukan konteks penggunaan

   Di mana konteks pengguna yang dimaksud meliputi pengguna yang berhubungan dengan usulan pengembangan sistem. Konteks penggunaan dapat berupa apa saja yang mempengaruhi suatu sistem.

2. Menspesifikasikan persyaratan pengguna

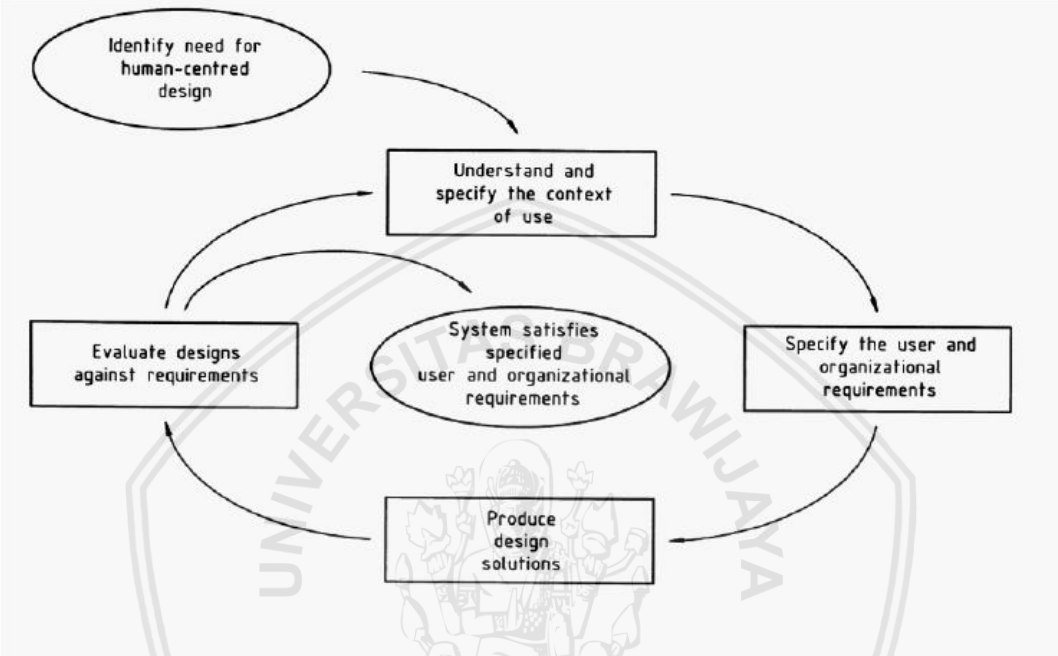
   Menentukan apa saja yang menjadi persyaratan dari pengguna berdasarkan dengan kebutuhan pengguna terhadap sistem.

3. Menghasilkan solusi desain untuk memenuhi persyaratan pengguna

Membuat suatu solusi desain berdasarkan tahap-tahap yang telah dilakukan sebelumnya sesuai dengan lingkungan pengguna dengan tujuan memenuhi *user experience* yang baik.

4. Mengevaluasi desain

   Melakukan evaluasi terhadap kesesuaian hasil rancangan dengan kebutuhan pengguna.



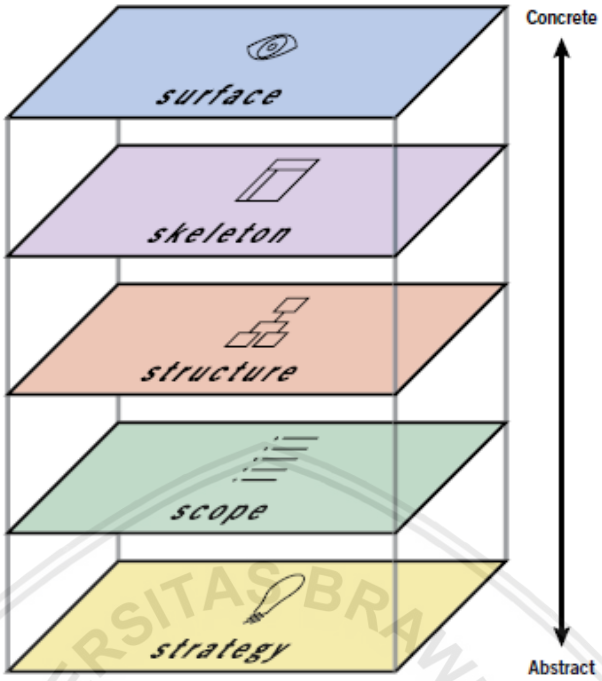
Gambar 2.5 Diagram Proses HCD Untuk Iteratif Sistem Berdasarkan Standar ISO 13407-1999

Sumber: (Pagliari, 2007)

Gambar 2.5 merupakan diagram proses HCD untuk iteratif sistem berdasarkan standar aktivitas-aktivitas yang ada pada ISO 13470 di mana tahap pertama dilakukan Identifikasi kebutuhan akan HCD, selanjutnya menentukan konteks penggunaan, selanjutnya menspesifikasikan kebutuhan user atau organisasi, selanjutnya menghasilkan solusi dari desain, selanjutnya mengevaluasi kebutuhan sampai mendapatkan kepuasan dari user. Jika tidak mendapatkan kepuasan dari user maka langkahnya diulangi dari menentukan konteks penggunaan.

2.9 Garett's 5 *Planes of User Experience*

Dalam bukunya yang berjudul "*The Elements of User Experience*" Jesse James Garrett menyebutkan bahwa dalam proses mendesain sebuah produk berdasarkan User Experience (UX) dibagi menjadi 5 elemen seperti dalam Gambar 2.5:



Gambar 2.6 Garett's 5 *Planes of User Experience*

Sumber: (Garett, 2003)

Dalam Gambar 2.6 merupakan kelima elemen menurut Garett dan dalam bukunya juga disebutkan bahka dalam membangun sebuah produk berdasarkan kelima elemen tersebut berdasarkan metode *Building from Bottom to Top* yaitu membangun mulai dari elemen paling bawah sampai paling atas dengan urutan sebagai berikut.

1. *Strategy*

   Merupakan elemen paling bawah yang menjadi tahap paling awal untuk menentukan bagaimana ingin membangun sebuah produk dengan UX yang baik. Pada elemen ini pertama-tama ditentukan terlebih dahulu tujuan dari produk yang dibangun untuk apa dan memahami apa saja yang menjadi kebutuhan pengguna.

2. *Scope*

   Dalam elemen ini dibuat untuk menentukan cakupan (*spoce*) terkait produk yang akan dibuat. Cakupan yang dimaksud merupakan translasi dari tujuan produk pada elemen sebelumnya menjadi spesifikasi dari kebutuhan fungsional yang ditawarkan ke pengguna.

3. *Structure*

   Setelah menentukan dan mendefinisikan persyaratan pengguna yang mencakup kebutuhan fungsional, maka dapat digambarkan mengenai desain interaksi dalam sistem.

4. *Skeleton*

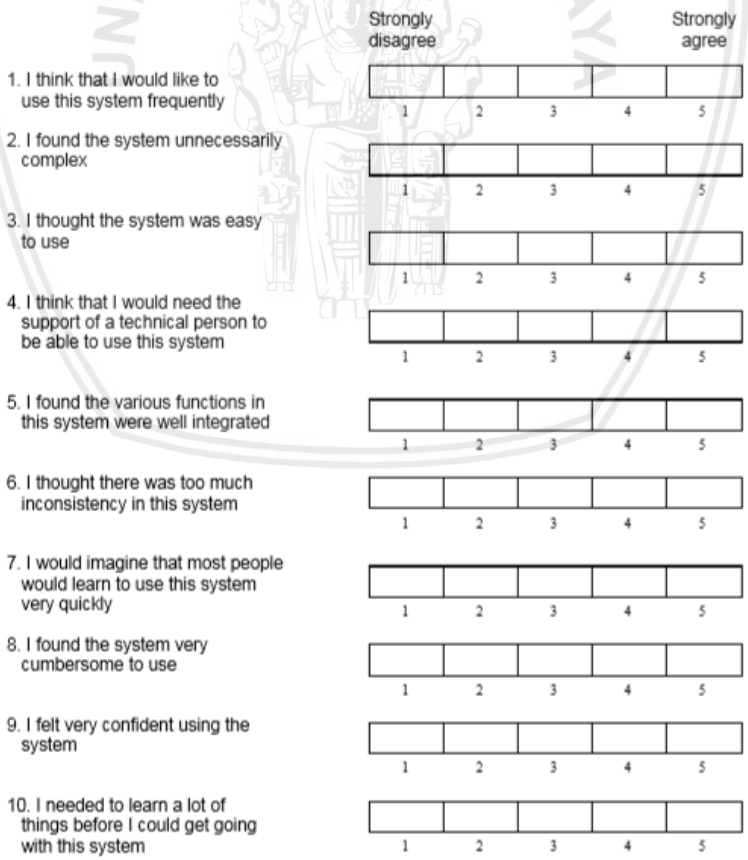
Setelah mengembangkan s*tucture* pada elemen sebelumnya, kemudian dikembangkan kembali secara lebih spesifik pada setiap aspek antarmuka dan navigasi.

5. *Surface*

Merupakan elemen paling atas yang lebih memperhatikan mengenai aspek-aspek yang secara langsung dapat dirasakan oleh pengguna. Pada lapisan ini menggabungkan seluruh konten, fungsionalitas, dan estetika dalam membuat desain akhir yang memenuhi tujuan dari elemen-elemen yang ada dibawahnya.

*2.10 System Usability Scale (*SUS*)*

SUS atau yang biasa dikenal juga dengan sebutan *likert scale* (skala likert) merupakan sebuah metode yang digunakan dalam pengujian *usability* dari sebuah sistem yang bertujuan untuk mendapatkan umpan balik dari koresponden. Dalam pengujian SUS terdapat 10 pertanyaan yang sudah mencakup pandangan luas mengenai penilaian subyektifitas dari *usability*. Dalam setiap pertanyaan dijawab dengan skala nilai 1 sampai 5 yang mengartikan jawaban dari sangat tidak setuju hingga sangat setuju seperti yang dapat dilihat dalam Gambar 2.7



Gambar 2.7 Daftar Pertanyaan Pengujian SUS

Sumber : (Brooke, 1996)

Dapat dilihat dalam Gambar 2.7 merupakan 10 daftar pertanyaan dalam pengujian SUS di mana untuk setiap pertanyaannya jawaban yang diperoleh adalah dari skala angka 1 sampai 5, penjelasan poin untuk skala jawaban dapat dilihat pada Tabel 2.3.

Tabel 2.3 Penilaian Jawaban Kuantitatif pada Pengujian SUS

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

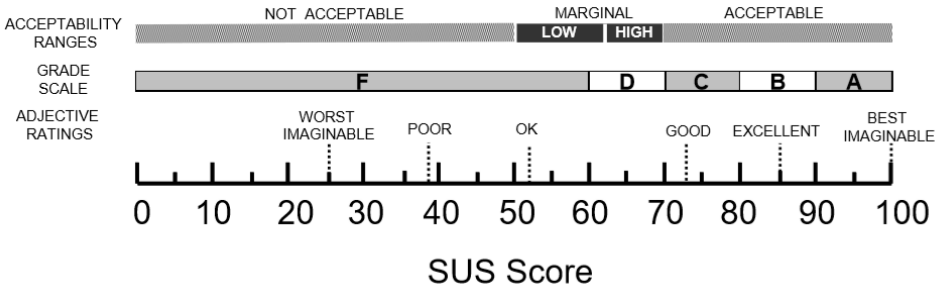
Pada Tabel 2.3 menjelaskan mengenai penilaian jawaban kuantitatif yang diperoleh dari masing-masing pertanyaan dari 10 pertanyaan yang diajukan pada pengujian SUS di mana skor atau skala nilai dari 1 sampai 5 memiliki makna tersendiri (Usabilit.gov, 2017).

Manfaat dari pengujian menggunakan metode SUS yaitu skala penilaian yang diperoleh sangat mudah dengan lingkup pengujian sampel kecil, namun hasil yang didapatkan dapat diandalkan dan juga valid untuk digunakan dalam menentukan apakah suatu sistem mudah untuk digunakan atau tidak (Usabilit.gov, 2017). SUS juga dapat memberikan hasil berupa satu angka sebagai ukuran nilai gabungan yang dapat mewakili *usability* dari sebuah sistem secara keseluruhan (Brooke, 1996).

Setelah keseluruhan data hasil dari pertanyaan sudah diperoleh, selanjutnya untuk mengkaji secara lebih dalam mengenai hasil dari SUS dapat dilakukan dengan menggunakan perhitungan *Relative Importance Indeks* (RII) (Nazrina Aziz *et al.*, 2016) seperti yang dapat dilihat menggunakan rumus yang ditunjukkan dalam Persamaan 2.1.

$$RII\ (\%) = \left(\frac{n1+2n2+3n3+4n4+5n5}{5(n1+n2+n3+n4+n5)}\right) x\ 100 \qquad (2.1)$$

Pada Persamaan 2.1 nilai dari n1, n2, n3, n4, n5 merupakan jumlah koresponden yaitu sebanyak 5 orang masing-masing dari nilai yang diperoleh dari koresponden sebesar 1 sampai 5 sesuai dengan skala skor pada SUS. Setelah diperoleh hasil akhir dari perhitungan RII maka dapat ditarik kesimpulan nilai dari skor pengujian SUS dengan interval nilai hasil yang digambarkan pada Gambar 2.8



Gambar 2.8 Hasil Skor Penilaian SUS

Sumber : (Bangor, et al., 2009)

2.11 *Mean Opinion Score* (MOS) *Testing*

Mean Opinion Score (MOS) *testing* merupakan salah satu dari jenis metode pengujian yang sifatnya subjektif. MOS digunakan untuk mengetahui tingkat kualitas dari sebuah komunikasi multimedia yang didasarkan pada pengalaman penggunanya *Quality-of-Experience* (QoE). Kebanyakan studi pada QoE menyatakan hanya MOS dan pemodelan yang bisa memetakan parameter QoE (Hoßfeld, Fiedler and Gustafsson, 2017).

Penerapan MOS *testing* pada pengujian melibatkan 6 sampai 40 orang sebagai responden berdasarkan rekomendasi dari *International Telecomunication Union* (ITU) melalui dokumen yang berjudul "*Subjective Audiovisual Quality Assessment Methods For Multimedia Applications",* jumlah ini juga sama dengan *usability testing* pada sebuah layanan (ITU, 1998).

Tabel 2.4 Opini pada MOS

Rating	Quality	Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying, but not objectionable
1	Bad	Very annoying and objectionable

Sumber: (Ribeiro, et al., 2011)

Pada Tabel 2.4 merupakan penilaian standar dari skala opini pada MOS di mana terdapat 5 jenis penilaian yaitu: *rating* 5 yang berarti kualitas dari sistem adalah *Excellent* (Sangat Bagus) dan distorsi (penyimpangan) yang ada pada sistem *Imperceptible* (Tidak Kelihatan), *rating* 4 berarti kualitas sistem adalah *Good* (Baik) dan distorsinya hanya terlihat namun tidak mengganggu, *rating* 3 berarti kualitas sistem adalah *Fair* (Cukup) dan distorsinya kelihatan dan sedikit mengganggu, *rating* 2 berarti kualitas sistem adalah *Poor* (Kurang) dan distorsinya mengganggu, tetapi tidak membuat sistem menjadi tidak disukai, dan *rating* 1 berarti *Bad*

(Buruk) dan distorsinya sangat mengganggu serta membuat sistem menjadi tidak disukai.

Secara umum pertanyaan-pertanyaan yang digunakan dalam pengujian MOS adalah mengenai kualitas layanan yang diberikan oleh suatu sistem berdasarkan pengalaman mereka dalam mengoperasikan sistem tersebut. Pertanyaan yang biasanya muncul seperti:

- Seberapa mudah bahasa yang diberikan aplikasi dapat dipahami?
- Seberapa baik kualitas suara dari output sistem?

Cara untuk mendapatkan intepretasi nilai rata-rata dari MOS terkait kalitas layanan dari *voice interaction* yang ada pada sebuah aplikasi adalah dengan menggunakan rumus (Justicia, 2017):

$$MOS = \frac{1}{N}\sum_{i=1}^{N} yi$$

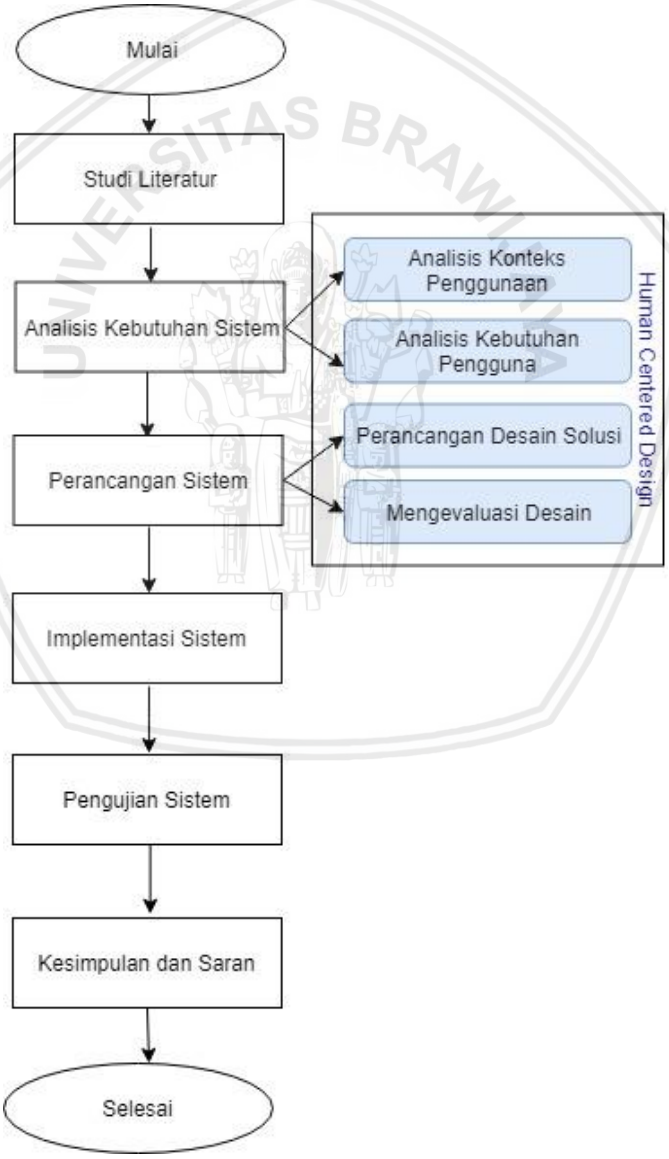
Keterangan :

N : Merupakan jumlah responden

yi : Merupakan nilai opini responden ke-i

BAB 3 METODOLOGI

Bab ini akan membahas tentang langkah-langkah yang dilakukan dalam penyelesaian masalah dan pembuatan aplikasi seperti dalam Gambar 3.1. Dalam metode penelitian dilakukan dengan menggunakan *waterfall* dan menggunakan pendekatan HCD dalam mendesain aplikasi ini. Pada metodologi penelitian ini terbagi menjadi beberapa tahapan, yaitu Studi Literatur, Analisis Kebutuhan yang didalamnya terdapat proses perancangan dengan metode HCD dan menenentukan: Analisis Konteks Penggunaan dan Analisis Kebutuhan Pengguna, Perancangan Sistem yang didalamnya terdapat proses perancangan dengan metode HCD dan melakukan: Perancangan Desain Solusi dan Mengevaluasi Desain, Implementasi Sistem dan pengambilan Kesimpulan dan Saran.



Gambar 3.1 Diagram Metodologi Penelitian

3.1 Studi Literatur

Pada Tahap pertama penelitian ini dilakukan studi literatur untuk memahami secara mendalam mengenai hal-hal yang memiliki keterkaitan dengan penelitian yang akan dilakukan baik dari segi konsep, dasar teori, maupun hal-hal yang beguna dalam membantu proses perancangan, implementasi, dan pengujian sistem.

Sumber pembelajaran yang menjadi acuan dari penelitian ini adalah mempelajari literatur dan ilmu-ilmu yang berkaitan dengan penelitian dari beberapa pustaka yang memiliki keterkaitan dengan aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tunenetra. Literatur yang digunakan sebagai bahan penunjang penelitian yang berasal dari buku, jurnal, paper, serta website terkait. Adapun yang dijadikan bahan studi literatur dan teori pustaka yang mendukung dengan penelitian ini adalah:

1. Google API yang juga memiliki modul khusus Google *Speech Recognation* API yang digunakan sebagai modul untuk pemrosesan *Speech Recognation* dan *Text to Speech* (TTS).
2. *Market Share* Sistem Operasi Perangkat Mobile.
3. Pemrograman Aplikasi Berbasis Web
4. *Framework* Ionic.
5. *Human Centered Design* (HCD).
6. Garett's 5 *Planes of User Experience.*
7. *System Usability Score* (SUS)
8. Mean Opinion Score (MOS) Testing.

3.2 Analisis Kebutuhan SIstem

Pada Tahap ini pertama-tama dilakukan tahap menganalisis kebutuhan sistem yang akan diimplementasikan pada aplikasi yang dikembangkan. Pada bagian analisis menggunakan pendekatan mendesain sistem berdasarkan pendekatan HCD. Adapun langkah-langkah analasis kebutuhan pada aplikasi ini adalah:

1. Melakukan Identifikasi Masalah yang nantinya akan diselesaikan melalui sistem yang dibangun.
2. Membuat gambaran umum mengenai sistem seperti apa yang akan dibangun dan disederhanakan dalam bentuk *storyboard* (ilustrasi gambar) sehingga dapat memudahkan dalam menjelaskan sistem yang akan dibangun.
3. Analisis konteks penggunaan dan merancang berdasarkan elemen paling dasar dalam pengembangan UX menurut Garett yaitu elemen *srategy.* Dalam menentukan konteks penggunaan dilakukan dengan metode wawancara terhadap satu pengguna lansia yang menderita tunanetra sedang (*partially sighted*) dan memiliki keterbatasan penglihatan dikarenakan faktor usia,

sehingga melalui proses ini didapatkan apa saja yang menjadi kebutuhan pengguna (*user needs*) terhadap sistem yang akan dirancang. Berdasarkan kebutuhan pengguna yang telah didapatkan maka diputuskan dalam konteks pengembangan sistem dilakukan pada perangkat *mobile* dan diimplementasikan secara *hybrid* agar bisa diimplementasikan secara lintas platform yang menjadi tujuan dari produk yang dirancang (*product objective*).

4. Analisis Kebutuhan Pengguna pengguna dalam bentuk daftar kebutuhan fungsional dan non-fungsional. Berdasarkan pengembangan UX menurut Garett dilakukan elemen *scope* dan diperoleh spesifikasi dari kebutuhan fungsional (*functional spesification*) berdasarkan kebutuhan pengguna yang diperoleh dari proses sebelumnya, dan persyaratan konten (*content requirement*) yang dibuat dalam bentuk gambar dan tabel meliputi pembuatan *Use Case Diagram* dan *Use Case Scenario*.

3.3 Perancangan Sistem

Tahap ini dilakukan sebagai panduan dalam melakukan penelitian dan pengujian. Pada proses perancangan didasari pada kebutuhan sistem yang telah didata pada tahap analisis kebutuhan. Dalam merancang sistem pada penelitian ini berhubungan dengan perancangan sistem yang dilakukan dengan menggunakan pendekatan HCD dan memvisualisasi dengan bantuan pemodelan Unified Modeling Language (UML). Tahap-tahap yang dilakukan dalam perancangan sistem:

1. Menggambarkan mengenai perancangan arsitektur sistem yang akan dibangun.
2. Memjelaskan aktivitas yang terjadi dalam sistem yang akan dibangun dalam bentuk *Actvity Diagram*
3. Menjabarkan mengenai komunikasi antar objek dengan menggunakan *Sequence Diagram*.
4. Memodelkan hubungan antar *class* dalam sistem dengan membuat *Class Diagram.*
5. Menjabarkankan basis data yang akan digunakan dalam sistem yang dibangun.
6. Menghasilkan Desain Solusi berdasarkan pendekatan mendesain sistem menggunakan HCD dan pengembangan UX menurut Garett dihasilkan rancangan dari antarmuka sistem yang akan dikembangkan melalui tiga elemen yaitu *structure* yang melakukan desain interaksi sistem dengan membuat *screenflow* dari sistem, *skeleton* yang akan menghasilkan sebuah rancangan kasar antarmuka dalam bentuk kerangka (*wireframe*), dan elemen terakhir yaitu *surface* yang merupakan tahap akhir dalam mendesain sistem sehingga menghasilkan suatu rancangan antarmuka yang menyerupai hasil implementasinya yang dibuat dalam sebuah *mockup*.
7. Mengevaluasi Desain yang telah dibuat dan melakukan wawancara dengan 1 orang calon pengguna lansia yang menderita tuna netra sedang dengan

menunjukan hasil rancangan dalam bentuk *mockup* untuk melakukan konfirmasi kesesuaian desain dengan kebutuhan pengguna.

3.4 Implementasi Sistem

Berdasarkan hasil yang diperoleh dari tahapan sebelumnya yaitu perancangan, di mana pada perancangan telah dibuat *sequence diagram* dan *class diagram* dan pada tahap ini akan diimplementasikan desain tersebut dalam bentuk kode program. Nantinya akan di implementasi sistem menjadi baris kode program sehingga dapat menjadi sebuah aplikasi yang sudah jadi dan dapat digunakan. Dalam implementasinya adalah pembuatan aplikasi *client* dengan pemrograman typescript dan ionic sebagai *framework* perancangan.

3.5 Pengujian dan Analisis

Pada Tahap ini dilakukan pendekatan mendesain sistem berdasarkan pendekatan HCD dilakukan dengan cara Mengevaluasi Aplikasi yang akan dilakukan terhadap hasil dari desain sistem dalam bentuk aplikasi dan disesuaikan dengan kebutuhan pengguna yang telah disebutkan sebelumnya. Evaluasi ini dilakukan untuk mengetahui sejauh mana perancangan yang dihasilkan sesuai kebutuhan pengguna. Pengujian terdiri dari menguji validasi, *usability,* yang beguna untuk menguji aspek kemudahan penggunaan aplikasi. Pengujian yang akan dilakukan dengan menggunakan metode *Black-box Testing,* metode *System Usability Scale* (SUS) dan *Mean Opinion Score* (MOS) dengan cara melakukan wawancara ke pengguna". Selanjutnya hasil dari pengujian digunakan sebagai kesimpulan akhir dari pengembangan aplikasi.

3.6 Kesimpulan dan Saran

Pada tahap ini kesimpulan diperoleh dari hasil pengujian dan analisis dari pengembangan aplikasi ini. Tahap akhir dari penelitian ini adalah saran yang dijabarkan untuk menyempurnakan pengembangan aplikasi ini serta memperbaiki kesalahan-kesalahan yang terjadi serta menjadi pertimbangan sebagai pengembangan aplikasi ini selanjutnya.

BAB 4 ANALISIS KEBUTUHAN SISTEM

Bab ini berisikan pembahasan mengenai analis kebutuhan sistem *voice call* dengan manajemen kontak berbasis suara bagi pengguna tuna netra. Dalam proses analisis terdiri dari penjabaran mengenai identifikasi masalah yang akan diselesaikan pada sistem, pembuatan gambaran umum sistem, menentukan konteks penggunaan sistem, membuat daftar kebutuhan, dan memodelkannya ke dalam *use case diagram* dan menjabarkannya dalam *use case scenario*

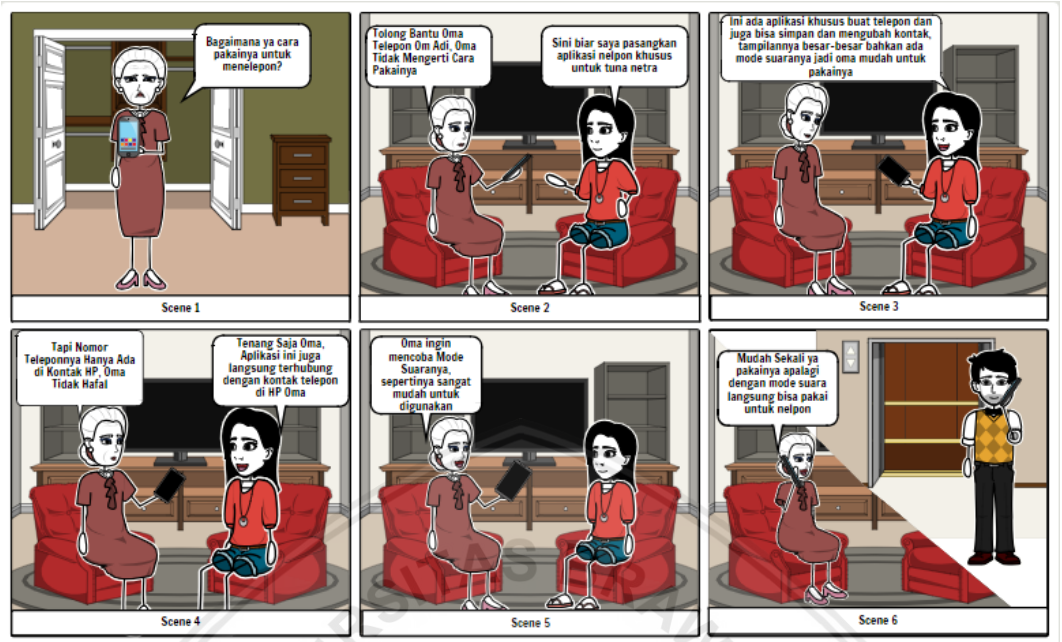
4.1 Identifikasi Masalah

Masalah yang diangkat dalam penelitian ini adalah bagi mereka yang memiliki keterbatasan penglihatan atau yang biasa disebut tuna netra sering kali merasa kesulitan dalam menggunakan alat komunikasi modern seperti *smartphone*. Dahulu ponsel diciptakan hanya sekedar menjadi alat komunikasi bergerak yang lebih praktis dan mudah dibawa ke mana saja, namun seiring dengan perkembangan jaman kini semakin banyak diciptakan ponsel cerdas atau yang sering dikenal dengan kata *smartphone*. Banyak hal yang ditawarkan dalam sebuah *smartphone* tidak hanya untuk berkomunikasi tetapi juga memiliki banyak fungsi lainnya seperti menjadi buku digital untuk menyimpan tulisan, tempat menyimpan file dalam bentuk foto atau video, alat untuk bermain game dan masih banyak hal lain yang dapat dilakukan dalam sebuah *smartphone*.

Namun fungsi-fungsi tersebut tidak terlalu dibutuhkan terutama bagi mereka yang memiliki keterbatasan penglihatan. Yang paling mereka butuhkan adalah kegunaannya sebagai alat komunikasi suara (telepon) yang bisa dibawa kemana-mana (*portable*), akan tetapi hampir seluruh *smartphone* dirancang dalam sebuah layar sentuh datar dan butuh kemampuan visual yang baik untuk dapat mengoperasikannya, karena pada dasarnya *smartphone* tidak diciptakan khusus bagi mereka yang memiliki keterbatasan penglihatan. Sehingga berdasarkan masalah tersebut diidentifikasi perlunya sebuah sistem khusus untuk menangani masalah dalam melakukan panggilan telepon berdasarkan masukkan nomor, melakukan panggilan kontak telepon dan melakukan manajemen kontak seperti melakukan cari kontak, edit kontak, dan hapus kontak.

Karena masalah yang dialami adalah keterbatasan konteks visual dalam mengoperasikan *smartphone*, maka dianjurkan sebuah solusi untuk membuat sistem khusus di mana masukkan dan keluaran dari proses yang dilakukan oleh sistem dapat diproses dalam bentuk suara. Adapun tampilan yang dirancang juga dalam ukuran yang besar sehingga bagi mereka yang hanya memiliki keterbatasan penglihatan sebagian setidaknya masih dapat melihat dan mengerti fungsi-fungsi yang ada di dalam sistem dengan tulisan yang besar dan bantuan *icon* yang menjelaskan secara visual fungsi dari tulisan tersebut. Serta semua proses yang dibuat mudah dipahami sehingga dapat digunakan oleh pengguna dengan usia diatas 50 tahun/lansia.

4.2 Gambaran Umum Sistem



Gambar 4.1 Gambaran Umum Sistem

Sistem v*oice call* dengan manajemen kontak berbasis suara ini merupakan aplikasi yang dirancang secara khusus dan bertujuan untuk membantu para penyandang tunanetra dalam berkomunikasi via suara menggunakan perangkat *smartphone*. Dengan menggunakan aplikasi ini diharapkan para pengguna, khususnya penyandang tunanetra dapat melakukan panggilan suara dengan memasukkan nomor telepon atau melakukan panggilan kontak yang sudah ada pada kontak telepon di *smartphone* mereka. Dikarenakan sistem ini dirancang untuk pengguna tuna netra, maka fungsional dibuat mudah diakses oleh pengguna tunanetra dengan ukuran tampilan antarmuka yang besar-besar juga dapat berjalan berdasarkan perintah suara, serta *feedback* yang diberikan juga dalam bentuk suara.

Seperti dalam Gambar 4.1 dibuat sebuah ilustrasi dalam bentuk gambar penggunaan sistem oleh seorang pengguna lansia yang menderita tunanetra sedang dan tidak dapat melihat dengan jelas karena faktor usia. Dalam Gambar 4.1 terbagi menjadi 6 *scene* yang menjelaskan gambaran umum sistem, pada *scene* 1 menggambarkan seorang tunanetra lansia yang memiliki sebuah *smartphone* dan ingin melakukan panggilan ke seorang kerabatnya, namun ia tidak dapat mengoperasikan *smartphone* miliknya sehingga tidak dapat melakukan panggilan telepon.

Selanjutnya dalam *scene 2* menggambarkan tunanetra lansia tersebut meminta bantuan cucunya untuk melakukan panggilan telepon dan cucunya berinisiatif untuk menginstallkan aplikasi *voice call* pada *smartphone* lansia tersebut. Lalu dalam *scene 3* menggambarkan aplikasi sudah berhasil terinstall pada *smartphone* milik lansia tersebut dan cucunya juga menjelaskan keunggulan dari aplikasi tersebut yaitu tampilan yang besar-besar sehingga memudahkan

lansia yang menderita tunanetra sebagian itu agar dapat melihat dan mengetahui lebih jelas masing-masing fungsi dari aplikasi tersebut, dan juga menjelaskan mengenai mode suara yang sangat membantu dalam penggunaan aplikasi.

Kemudian pada *scene* 4 menggambarkan cucunya yang memberi tahu keunggulan lain dari aplikasi yaitu dapat terhubung dengan kontak pada *smartphone* sehingga kontak yang sudah ada sebelumnya dapat langsung terdaftar dan ada pada aplikasi. Selanjutnya pada *scene* 5 menggambarkan lansia tersebut mencoba menggunakan aplikasi, karena masih pertama kali menggunakan aplikasi lansia tersebut memanfaatkan mode suara yang dirasa sangat mudah untuk digunakan. Terakhir pada *scene* 6 menggambarkan lansia tersebut berhasil melakukan panggilan telepon terhadap kontak yang sudah ia miliki sebelumnya tanpa bantuan orang lain dan sudah dapat terhubung melalui panggilan telepon dengan keponakan yang ingin dia telepon.

4.3 Konteks Penggunaan Sistem

Sistem yang akan dirancang berdasarkan pendekatan HCD, sehingga melibatkan manusia dalam perancangannya, dalam mempermudah pengembangan aplikasi ini serangkaian proses penggalian kebutuhan pengguna tuna netra terkait aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tuna netra dilakukan pada seorang lansia yang berusia 75 tahun yang mengalami keterbatasan penglihatan terutama terhadap tulisan kecil dikarenakan faktor usia sehingga mederita tuna netra sedang (*partially sighted*). Tahap penggalian kebutuhan dilakukan melalui proses wawancara, berdasarkan hasil wawancara disimpulkan kebutuhan pengguna terhadap sistem seperti pada Tabel 4.1.

Tabel 4.1 Analisis Daftar Kebutuhan Pengguna

No.	Pernyataan Kebutuhan
1	Sistem menyediakan mode suara pada setiap fungsi yang disediakan.
2	Sistem harus bisa digunakan pengguna tanpa bantuan orang lain.
3	Sistem dapat digunakan oleh pengguna tuna netra lansia yang berusia diatas 50 tahun.
4	Sistem tersedia di perangkat bergerak *smartphone* dengan sistem operasi Android.
5	Sistem perlu menyediakan tampilan *icon* untuk menjelaskan fungsi aplikasi sehingga lebih mudah dipahami tanpa perlu membaca tulisan mengenai keterangan fungsi sistem.
6	Tampilan pada sistem perlu dibuat semudah mungkin untuk dimengerti oleh pengguna tuna netra sedang terutama bagi pengguna dengan usia diatas 50 tahun/lansia

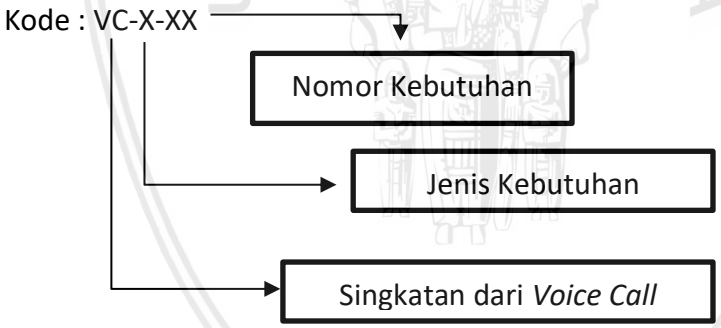
Tabel 4.1 merupakan daftar dari analisis kebutuhan pengguna yang didapatkan dari proses wawancara dengan satu orang calon pengguna yang menderita tuna

netra sedang dan berusia 75 tahun dan sudah termasuk lansia. Berdasarkan dari analisis kebutuhan pengguna maka disimpulkan karakteristik pengguna pada sistem *voice call* dengan menajemen kontak adalah:

1. Pengguna adalah orang-orang yang menderita gangguan penglihatan (tuna netra) sedang sampai berat dengan rentang usia diatas 50 tahun, namun tidak menutup kemungkinan pengguna tuna netra yang berusia 9 tahun keatas juga dapat menggunakannya karena aplikasi ini dirancang semudah mungkin digunakan oleh penderita tuna netra dan masukkan dan keluaran aplikasi ini juga berupa suara.
2. Dalam penggunaan sistem tidak membatasi jeniskelamin pengguna, jadi baik laki-laki maupun perempuan dapat menggunakan sistem ini.
3. Tidak adanya ketentuan pekerjaan atau profesi khusus dalam menggunakan sistem ini

4.4 Daftar Kebutuhan

Dalam perancangan aplikasi *voice call* dengan manajemen kontak berbasis suara dibagi menjadi dua kategori yaitu: kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional berisikan mengenai layanan apa saja yang disediakan oleh sistem. Kebutuhan non-fungsional berisikan mengenai batasan sistem. Untuk pemberian identitas terhadap sebuah kebutuhan, digunakan sebuah kode dengan ketentuan seperti pada Gambar 4.2 dan Tabel 4.1.



Gambar 4.2 Aturan Kode kebutuhan

Gambar 4.2 merupakan aturan kode kebutuhan yang dimulai dengan huruf VC yang merupakan singkatan dari *voice call*, dan selanjutnya diikuti dengan angka jenis kebutuhan, lalu nomor kebutuhan.

Tabel 4.2 Aturan Penomoran Kebutuhan

Kode	Hal/ Bagian
VC	*Voice Call*
1	Kebutuhan Fungsional
2	Kebutuhan Non-Fungsional
1..n	Kode Kebutuhan [1.....n]

Tabel 4.2 Aturan Penomoran Kebutuhan merupakan aturan penomoran kebutuhan berdasarkan kode kebutuhan yang dimulai dari istilah VC yang merupakan singkatan dari *voice call*, dan untuk digit kedua X menunjukkan jenis kebutuhan yang terdiri dari 1 yang menandakan kebutuhan fungsional, dan 2 yang menandakan kebutuhan non-fungsional. Untuk digit ketiga XX menunjukkan kode kebutuhan yang dimulai dari angka 1 sampai n.

Tabel 4.3 Tabel Kebutuhan Fungsional

Kode	Kebutuhan	*Use Case*
VC-1-1	Aplikasi harus mampu melakukan penambahan kontak baru pada *smartphone* pengguna berdasarkan perintah suara.	Menambah Kontak
VC-1-2	Aplikasi harus mampu melakukan pencarian kontak yang ada pada *smartphone* pengguna berdasarkan nama kontak melalui perintah suara.	Mencari Kontak
VC-1-3	Aplikasi harus mampu melakukan pengeditan terhadap kontak yang sudah ditemukan dari hasil pencarian kontak pada *smartphone* pengguna melalui perintah suara.	Mengedit Kontak
VC-1-4	Aplikasi harus mampu menghapus kontak yang sebelumnya sudah terdaftar dalam kontak telepon di *smartphone* pengguna.	Menghapus Kontak
VC-1-5	Aplikasi harus mampu melakukan panggilan suara terhadap kontak yang sudah ditemukan dari hasil pencarian kontak pada *smartphone* pengguna melalui perintah suara.	Menelepon Kontak
VC-1-6	Aplikasi harus mampu melakukan panggilan suara berdasarkan nomor telepon yang dimasukkan pengguna melalui perintah suara.	Menelepon Berdasarkan Masukkan Nomor Telepon
VC-1-7	Aplikasi harus mampu melakukan pengaturan tampilan yang muncul ketika aplikasi pertama dibuka berupa mode suara ataupun mode tampilan.	Melakukan Pengaturan Tampilan Awal

Tabel 4.3 merupakan daftar tabel kebutuhan fungsional dari sistem yang akan dirancang. Pada Tabel 4.3 terdapat 6 kebutuhan fungsional yang akan dimodelkan

ke dalam bahasa pemodelan UML *Use Case* dan diimplementasikan dalam wujud fitur aplikasi.

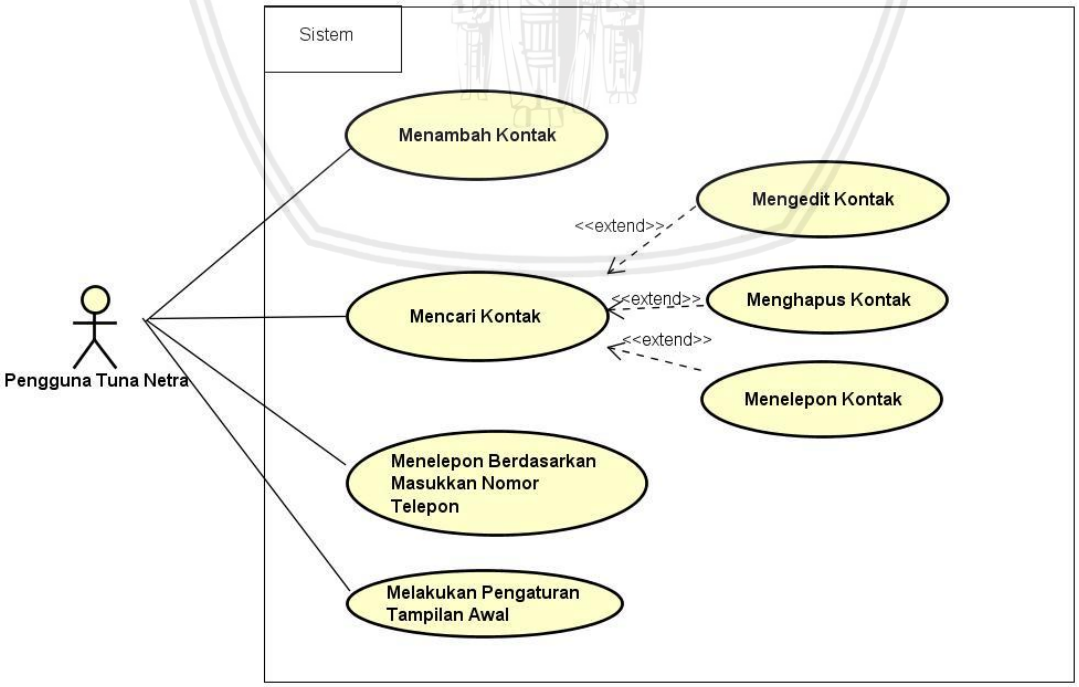
Tabel 4.4 Tabel Kebutuhan Non-fungsional

Kode	Parameter	Deskripsi Kebutuhan
VC-2-1	*Usability*	Aplikasi harus mampu memberikan kemudahan penggunaan aplikasi, terutama bagi pengguna tunanetra dengan nila minimal untuk hasil pengujian *usability* menggunakan metode SUS sebesar 80.
VC-2-2	*Quality of Experience* (QoE)	Aplikasi harus mampu memberikan nilai *Quality of Experience* (QoE) yang akan diukur menggunakan MOS testing dengan nilai rata-rata minimal 3.

Tabel 4.4 merupakan daftar tabel kebutuhan non-fungsional dari sistem yang dirancang. Ada 2 kebutuhan non-fungsional pada sistem ini yaitu kebutuhan dengan parameter *Usability* dan *Quality of Experience* (QoE).

4.5 *Use Case* Diagram

Use Case Diagram merupakan sebuah diagram yang digunakan dalam memodelkan perangkat lunak berdasarkan dari kebutuhan perangkat lunak berdasarkan sudut pandang aktor yang digambarkan pada Gambar 4.3.



Gambar 4.3 *Use Case* Diagram

Dalam Gambar 4.3 merupakan *use case* diagram dalam aplikasi ini, seperti yang dapat dilihat dalam Gambar 4.3 terdapat satu aktor yaitu Pengguna Tuna Netra dan terdapat 7 *use case* yaitu: Menambah Kontak, Mencari Kontak, Mengedit Kontak, Menghapus Kontak, Menelepon Kontak, Menelepon Berdasarkan Masukkan Nomor Telepon dan Melakukan Pengaturan Tampilan Awal. Yang mana terdapat relasi *extend* yang menunjukkan ada *use case* yang dapat dijalankan secara opsional yaitu *use case* Edit Kontak dan Telepon Kontak yang merupakan *supplier use case* dari *base use case* Cari Kontak.

Tabel 4.5 *Use Case Scenario* Menambah Kontak

Use Case Scenario Menambah Kontak (VC-1-1)	
Objective	Menambahkan kontak baru pada smartphone pengguna
Actor	Pengguna Tunanetra
Pre-Condition	Voice User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode suara atau Actor menekan tombol mode suara pada halaman utama.
	Graphical User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode tampilan dan Actor berada pada halaman utama.
Main Flow	*Voice User Interface*
	1. Sistem memberikan *feedback* "Selamat datang di aplikasi, katakan oke untuk memulai" 2. *Actor* menyebutkan "Oke" 3. Sistem memberikan *feedback* pilihan menu yang ada pada sistem. 4. *Actor* menyebutkan pilihan "2". 5. Sistem akan memberikan *feedback* suara berupa "Sebutkan nama kontak yang akan disimpan" 6. *Actor* menyebutkan nama kontak yang ingin ditambahkan

	7. Sistem memberikan *feedback* konfirmasi nama kontak yang akan disimpan berupa "Apakah benar nama yang akan disimpan adalah <nama> katakan ya bila benar katakan tidak bila salah" 8. *Actor* menyebutkan pilihan "Ya" 9. Sistem memberikan *feedback* suara berupa "Sebutkan nomor kontak yang akan disimpan" 10. *Actor* menyebutkan nomor kontak yang akan disimpan 11. Sistem memberikan *feedback* konfirmasi nomor kontak yang akan disimpan berupa "Apakah benar nomor kontak yang akan disimpan adalah <nomor> katakan ya bila benar katakan tidak bila salah" 12. *Actor* menyebutkan pilihan "Ya" 13. Sistem memberikan *feedback* suara berupa "Berhasil Menyimpan Kontak"
	Graphical User Interface
	1. *Graphical User Interface* 2. *Actor* menekan tombol tambah kontak 3. Sistem menampilkan halaman tambah kontak awal 4. *Actor* memilih tombol tambah kontak pada halaman awal 5. Sistem menampilkan halaman yang berisi *form input* untuk tambah kontak 6. Actor mengisikan seluruh *form input* dan menekan tombol simpan
	Voice User Interface
Alternatif Flow	1. Jika suara *actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas" 2. Jika *actor* menyebutkan pilihan tidak maka sistem akan mengulangi sesi yang sebelumnya
	Graphical User Interface
	-

Post-Condition	Kontak baru berhasil ditambahkan pada *smartphone actor*.

Pada Tabel 4.5 merupakan *use case scenario* menambah kontak yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara "2" atau *Actor menekan* tombol "Tambah Kontak"

Tabel 4.6 *Use Case Scenario* Mencari Kontak

Use Case Scenario Mencari Kontak (VC-1-2)	
Objective	Mencari kontak yang sudah ada pada *smartphone* pengguna
Actor	Pengguna Tunanetra
Pre-Condition	*Voice User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode suara atau *Actor* menekan tombol mode suara pada halaman utama.
	Graphical User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode tampilan dan *Actor* berada pada halaman utama.
Main Flow	*Voice User Interface*
	1. Sistem memberikan *feedback* "Selamat datang di aplikasi, katakan oke untuk memulai" 2. *Actor* menyebutkan "Oke" 3. Sistem memberikan *feedback* pilihan menu yang ada pada sistem. 4. *Actor menyebutkan pilihan* "1". 5. Sistem akan memberikan *feedback* suara berupa "Sebutkan nama kontak yang akan dicari" 6. *Actor* menyebutkan nama kontak yang ingin dicari

	7. Sistem memberikan *feedback* suara nama kontak yang ditemukan beserta urutannya 8. *Actor* memilih urutan kontak yang ditemukan
	Graphical User Interface
	1. *Actor* menekan tombol Cari Kontak 2. Sitem menampilkan halaman Cari Kontak
	Voice User Interface
Alternatif Flow	1. Jika suara *actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas" 2. Jika kontak yang dicari tidak ditemukan maka sistem akan memberikan *feedback* "Hasil pencarian tidak ada silahkan ulangi pencarian" dan mengulang memberikan *feedback* mulai dari "Sebutkan nama kontak yang ingin dicari "
	Graphical User Interface
	-
Post-Condition	Kontak berhasil ditemukan

Pada Tabel 4.6 merupakan *use case scenario* mencari kontak yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara "1" atau *Actor* menekan tombol "Cari Kontak".

Tabel 4.7 *Use Case Scenario* Mengedit Kontak

Use Case Scenario Mengedit Kontak (VC-1-3)	
Objective	Mengedit kontak yang sudah ada pada *smartphone* pengguna
Actor	Pengguna Tunanetra
Pre-Condition	*Voice User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode suara atau *Actor* menekan tombol mode suara pada halaman utama.
	Graphical User Interface

	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* berada pada halaman cari kontak dan sudah menemukan satu nama kontak yang ingin diedit
Main Flow	*Voice User Inteface*
	1. Sistem memberikan *feedback* "Selamat datang di aplikasi, katakan oke untuk memulai" 2. *Actor* menyebutkan "Oke" 3. Sistem memberikan *feedback* pilihan menu yang ada pada sistem. 4. *Actor menyebutkan pilihan* "3". 5. Sistem memberikan *feedback* pilihan menu untuk edit kontak yaitu: 1 untuk mengedit nama saja, 2 untuk mengedit nomor saja, 3 untuk mengedit keduanya 6. *Actor* menyebutkan pilihan menu untuk edit kontak 7. Sistem *memberikan feedback* untuk meminta masukkan berupa suara dari *Actor* sesuai dengan menu yang dipilih oleh *Actor* 8. *Actor* memberikan masukkan suara berdasarkan *feedback* dari sistem 9. Sistem memberikan *feedback* berupa konfirmasi masukkan dari *Actor* 10. *Actor* 11. 1 nama saja, 2 nomor saja 3 keduanya 12. *Actor* menyebutkan pilihan "3" atau menekan tombol dengan tulisan "Edit" 13. Sistem akan memberikan *feedback* suara berupa "Sebutkan nama kontak yang akan diubah" 14. *Actor* menyebutkan nama kontak yang akan diubah 15. Sistem memberikan *feedback* konfirmasi nama kontak

	16. *Actor* mengkonfirmasi nama kontak yang akan diubah 17. Sistem akan memberikan *feedback* suara berupa "Sebutkan nomor kontak yang akan diubah" 18. *Actor* menyebutkan nomor kontak yang akan diubah 19. Sistem memberikan *feedback* konfirmasi nomor kontak berupa Ya/Tidak 20. *Actor* menyebutkan "Ya" 21. Sistem menyimpan perubahan kontak memberikan *feedback* suara "Berhasil Mengedit Kontak".
	Graphical User Interface
	1. *Actor* menekan tombol edit pada halaman cari kontak 2. Sistem menampilkan halaman edit kontak dan *form* input untuk melakukan edit kontak. 3. *Actor* melakukan perubahan pada salah satu/kedua *for*m input 4. *Actor* menekan tombol simpan
Alternatif Flow	*Voice User Interface*
	1. Jika suara *actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas" 2. Jika *actor* menyebutkan pilihan tidak maka sistem akan mengulangi sesi yang sebelumnya
	Graphical User Interface
	-
Post-Condition	Kontak berhasil diedit

Pada Tabel 4.7 merupakan *use case scenario* megedit kontak yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara "1/2/3" setelah melakukan *scenario* mencari kontak atau *Actor* menekan tombol "Edit Kontak"

Tabel 4.8 *Use Case Scenario* Menghapus Kontak

Use Case Scenario Mengedit Kontak (VC-1-4)	
Objective	Menghapus kontak yang sudah ada pada *smartphone* pengguna
Actor	Pengguna Tunanetra
Pre-Condition	*Voice User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* sudah melakukan perintah edit kontak dan memilih satu kontak yang akan dihapus
	Graphical User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* berada pada halaman edit kontak pada satu kontak milik *Actor*
Main Flow;	*Voice User Interface*
	1. Sistem memberikan *feedback* pilihan yang akan dilakukan pada kontak yang akan dilakukan perubahan. 2. *Actor menyebutkan* pilihan "4" 3. Sistem memberikan *feedback* konfirmasi untuk menghapus kontak berupa pilihan Ya/Tidak 4. *Actor* menyebutkan pilihan "Ya" 5. Sistem memberikan *feedback* "Berhasil Menghapus Kontak"
	Graphical User Interface
	1. *Actor* menekan tombol hapus *2.* Sistem menampilkan dialog konfirmasi untuk menghapus kontak *3.* *Actor* menekan tombol Iya pada dialog konfirmasi
	Voice User Interface

Alternatif Flow	1. Jika suara *Actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas" 2. Jika *Actor* menyebutkan pilihan "Tidak" maka sistem akan mengulangi sesi yang sebelumnya
	Graphical User Interface
	-
Post-Condition	Kontak berhasil dihapus

Pada Tabel 4.8 merupakan *use case scenario* megedit kontak yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara hapus kontak saat selesai melakukan pencarian kontak atau *Actor* tombol "Edit Kontak".

Tabel 4.9 *Use Case Scenario* Menelepon Kontak

Use Case Scenario Menelepon Kontak (VC-1-5)	
Objective	Menelepon kontak yang sudah ada pada *smartphone* pengguna
Actor	Pengguna Tunanetra
Pre-Condition	*Voice User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* sudah melakukan perintah cari kontak dan memilih satu kontak yang akan ditelepon
	Graphical User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* berada pada halaman cari kontak dan sudah menemukan satu nama kontak yang ingin ditelepon
Main Flow	*Voice User Interface*
	1. Sistem memberikan *feedback* pilihan urutan nomor kontak yang ditemukan *2.* *Actor* menyebutkan perintah berupa nomor urutan kontak yang akan dipanggil

	3. Sistem mengarahkan tampilan ke halaman panggilan telepon milik *smartphone Actor*
	Graphical User Inteface
	1. *Actor* menekan tombol telepon 2. Sistem mengarahkan tampilan ke halaman panggilan telepon milik *smartphone Actor*
Alternatif Flow	*Voice User Interface*
	Jika suara *actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas"
	Graphical User Interface
	-
Post-Condition	Kontak berhasil ditelepon dan sistem mengarahkan tampilan ke halaman panggilan kontak.

Pada Tabel 4.9 merupakan *use case scenario* menelepon kontak yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara "3" setelah menemukan salah satu kontak saat melakukan pencarian kontak atau *Actor menekan* tombol "Telepon Kontak" pada halaman Cari Kontak.

Tabel 4.10 *Use Case Scenario* Menelepon Bedasarkan Masukkan Nomor telepon

Use Case Scenario Menelepon Berdasarkan Masukkan Nomor Telepon(VC-1-6)	
Objective	Menelepon kontak yang sudah ada pada *smartphone* pengguna
Actor	Pengguna Tunanetra
Pre-Condition	*Voice User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur dalam mode suara atau *Actor* menekan tombol mode suara pada halaman utama.
	Graphical User Interface
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta mode awal tampilan diatur

	dalam mode tampilan dan *Actor* berada pada halaman utama.
Main Flow	*Voice User Interface*
	1. Sistem memberikan *feedback* "Selamat datang di aplikasi, katakan oke untuk memulai" 2. *Actor* menyebutkan "Oke" 3. Sistem memberikan *feedback* pilihan menu yang ada pada sistem. 4. *Actor menyebutkan pilihan* "4". 5. Sistem akan memberikan *feedback* suara berupa "Sebutkan nomor telepon yang akan dipanggil" 6. *Actor* menyebutkan nomor telepon yang dipanggil. 7. Sistem memberikan *feedback* konfirmasi nomor yang telah disebutkan dan menanyakan konfirmasi dari *Actor* berupa Ya/Tidak 8. *Actor* menyebutkan perintah "Ya" 9. Sistem mengarahkan tampilan ke halaman panggilan telepon milik *smartphone Actor*
	Graphical User Interface
	1. *Actor* menekan tombol Telepon 2. Sitem menampilkan halaman Panggilan Nomor 3. *Actor* memasukkan nomor telepon yang akan dipanggil 4. *Actor* menekan tombol Telepon 5. Sistem mengarahkan tampilan ke halaman panggilan telepon milik *smartphone Actor*
Alternatif Flow	*Voice User Interface*
	Jika suara *actor* kurang jelas maka sistem akan memberikan *feedback* suara berupa "Maaf suara anda kurang jelas"
	Graphical User Interface
	-

Post-Condition	Nomor telepon yang dimasukkan berhasil ditelepon dan sistem mengarahkan tampilan ke halaman panggilan kontak.

Pada Tabel 4.10 merupakan *use case scenario* menelepon berdasarkan masukkan nomor telepon yang dapat dilakukan oleh pengguna tuna netra. Pada sistem *scenario* ini dipicu melalui perintah suara "3" atau *Actor menekan* tombol "Telepon Nomor".

Tabel 4.11 *Use Case Scenario* Melakukan Pengaturan Tampilan Awal pada Aplikasi

Use Case Scenario Melakukan Pengaturan Tampilan Awal (VC-1-7)	
Objective	Mengubah tampilan awal ketika aplikasi pertama kali dijalankan
Actor	Pengguna Tunanetra
Pre-Condition	*Graphical User Interface*
	Aplikasi telah dibuka dan *smartphone* pengguna terhubung dengan jaringan internet, serta *Actor* telah berada pada halaman utama aplikasi
Main Flow	*Graphical User Interface*
	1. Pengguna menekan tombol dengan tulisan pengaturan 2. Sistem menampilkan kotak dialog pengaturan 3. Pengguna memilih salah satu pilihan *radio Buton* yang disediakan 4. Pengguna menekan tombol "Ok"
Alternatif Flow	*Graphical User Interface*
	-
Post-Condition	Sistem menyimpan data perubahan pengaturan tampilan utama dan mengubah tampilan utama ketika aplikasi dibuka kembali.

Pada Tabel 4.11 merupakan *use case scenario* untuk melakukan pengaturan tampilan awal pada sistem, yang mana tampilan awal yang dapat diubah adalah tampilan awal untuk masuk ke mode suara ataupun tampilan awal untuk masuk ke mode tampilan.

BAB 5 PERANCANGAN SISTEM

Bab ini berisikan mengenai pembahasan dalam perancangan sistem *voice call* dengan manajemen kontak berbasis suara bagi pengguna tuna netra. Perancangan sistem bertujuan untuk memodelkan perangkat lunak berdasarkan dari hasil analisis kebutuhan yang diperoleh pada tahap sebelumnya. Pada tahap ini terdiri dari perancangan arsitektur sistem, perancangan *activity diagram,* perancangan *sequence* diagram *,* perancangan *class diagram,* perancangan *screenflow*, perancangan *screenflow, perancangan antarmuka* dalam bentuk *wireframe* dan perancangan antarmuka dalam bentuk *mockup*.

5.1 Perancangan Arsitektur Sistem

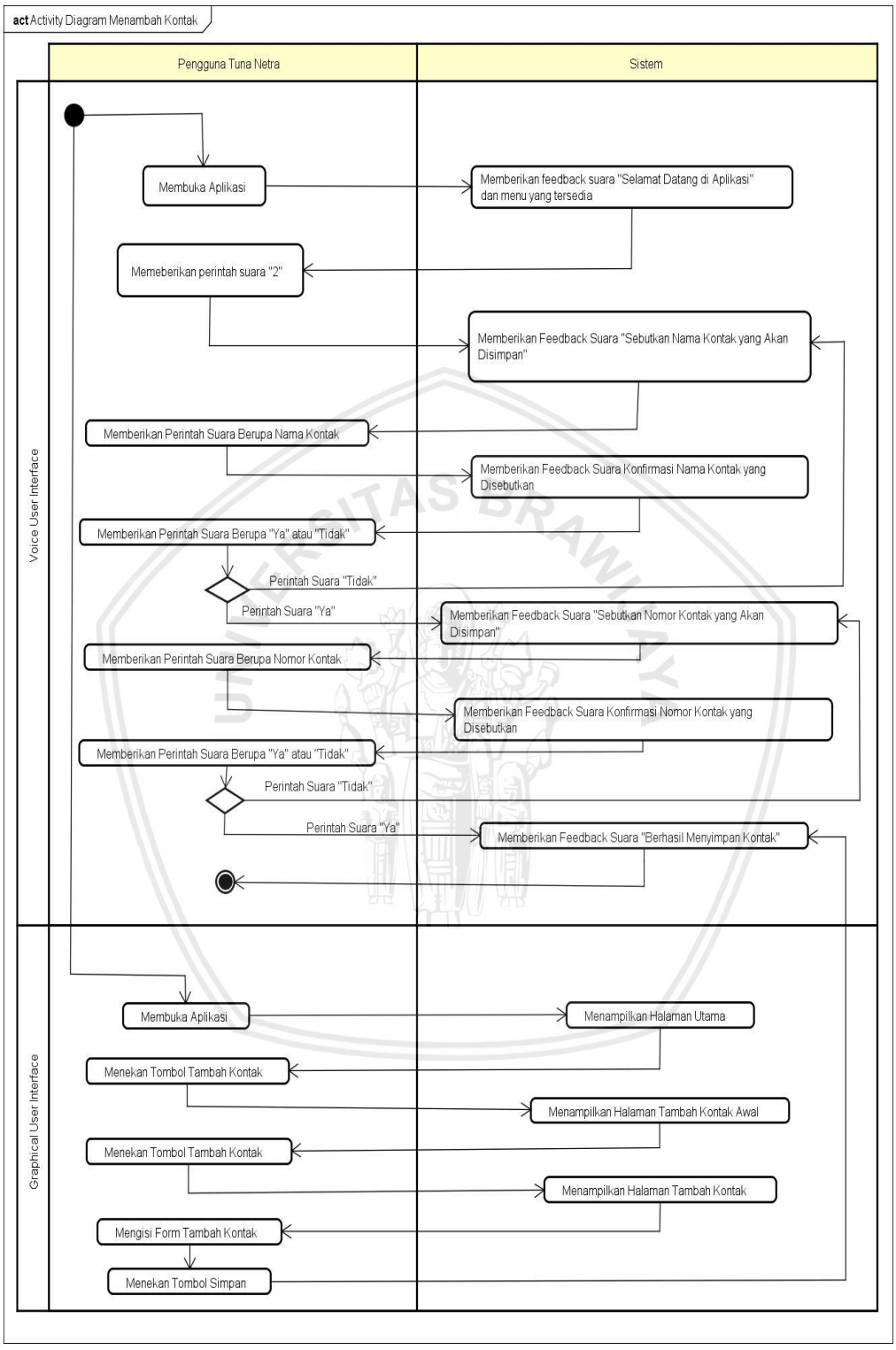
Gambar 5.1 Rancangan Arsitektur Sistem

Gambar 5.1 merupakan rancangan arsitektur dalam sistem dalam pemrosesan mode suara. Seperti yang dapat dilihat pada Gambar 5.1 untuk melakukan pemrosesan masukkan dan keluaran dalam bentuk suara menggunakan Google API yaitu menggunakan Google Text to Speech untuk pemrosesan *Text to Speech* pada sistem agar *feedback* yang diberikan kepada pengguna adalah dalam bentuk suara, dan juga menggunakan Google Speech Recognition dalam pemrosesan *Speech to Text* yang berfungsi untuk menangkap masukkan berupa suara yang berikan oleh pengguna dapat diolah menjadi bentuk teks sehingga nantinya dapat dimengerti dan di proses oleh sistem.

5.2 Perancangan *Activity Diagram*

Activity Diagram merupakan sebuah diagram yang digunakan dalam memodelkan perangkat lunak berdasarkan aktivitas-aktifitas yang terjadi pada sistem ataupun aktivitas yang melibatkan sistem. Pembuatan *activity diagram* mengacu pada *use case diagram* yang telah dirancang sebelumnya. *Activity diagram* berfungsi untuk mengambarkan diagram alir dari aktivitas yang terjadi dalam sistem yang sedang dirancang, dan juga menggambarkan bagaimana tiap-tiap dari diagram alir ini berawal, keputusan apa saja yang mungkin terjadi, dan bagaimana prosesnya diakhiri. Pada aplikasi ini terdapat 2 pelaku yaitu Pengguna Tuna Netra dan Aplikasi.

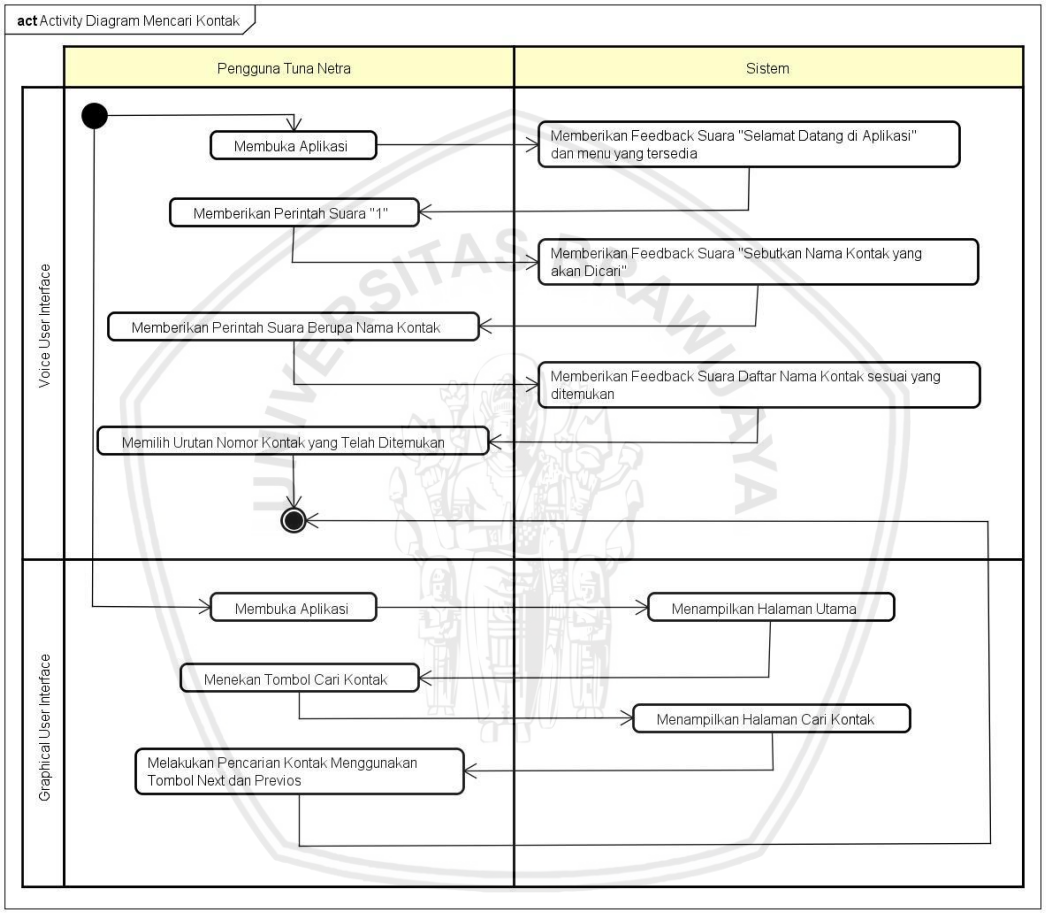
1) *Activity Diagram* Menambah Kontak



Gambar 5.2 *Activity Diagram* Menambah Kontak

Gambar 5.2 merupakan gambar *activity diagram* untuk melakukan tambah kontak baru pada *smartphone* pengguna. Dalam Gambar 5.2 memodelkan proses tambah kontak yang melibatkan pengguna tuna netra dan aplikasi. Di mana pada *activity diagram* ini terdiri dari *voice user interface* dan *graphical user interface* di mana untuk *voice user interface* sistem akan berjalan sesuai dengan perintah suara yang dimasukkan oleh pengguna dan akan mengembalikan respons/*feedback* berupa suara kepada pengguna.

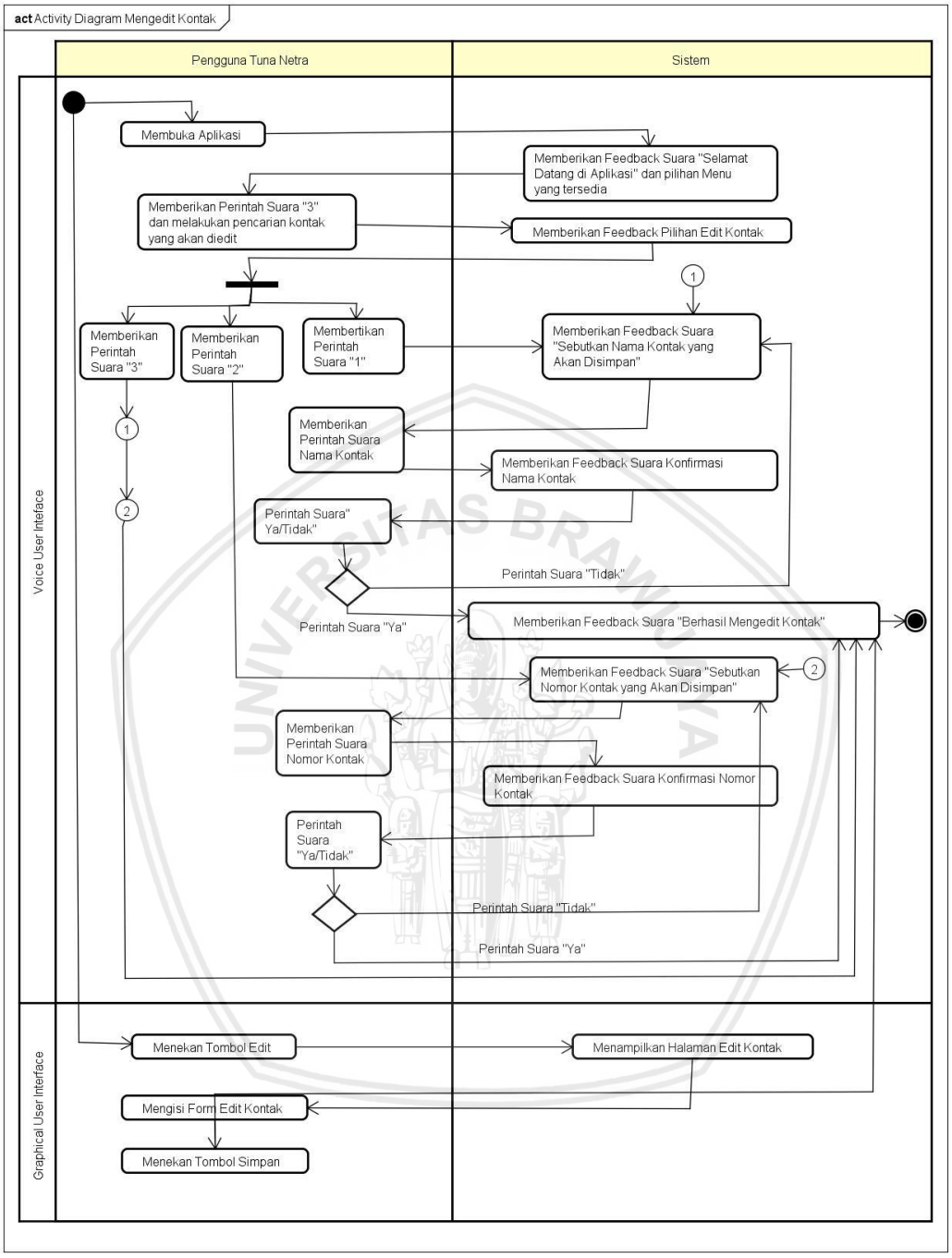
2) *Activity Diagram* Mencari Kontak



Gambar 5.3 *Activity Diagram* Mencari Kontak

Gambar 5.3 merupakan gambar *activity diagram* untuk melakukan cari kontak. Dalam Gambar 5.3 memodelkan proses tambah kontak yang melibatkan pengguna tuna netra dan aplikasi. Di mana pada *activity diagram* ini terdiri dari *voice user interface* dan *graphical user interface* di mana untuk *voice user interface* sistem akan berjalan sesuai dengan perintah suara yang dimasukkan oleh pengguna dan akan mengembalikan respons/*feedback* berupa suara kepada pengguna.

3) *Activity* Diagram Mengedit Kontak

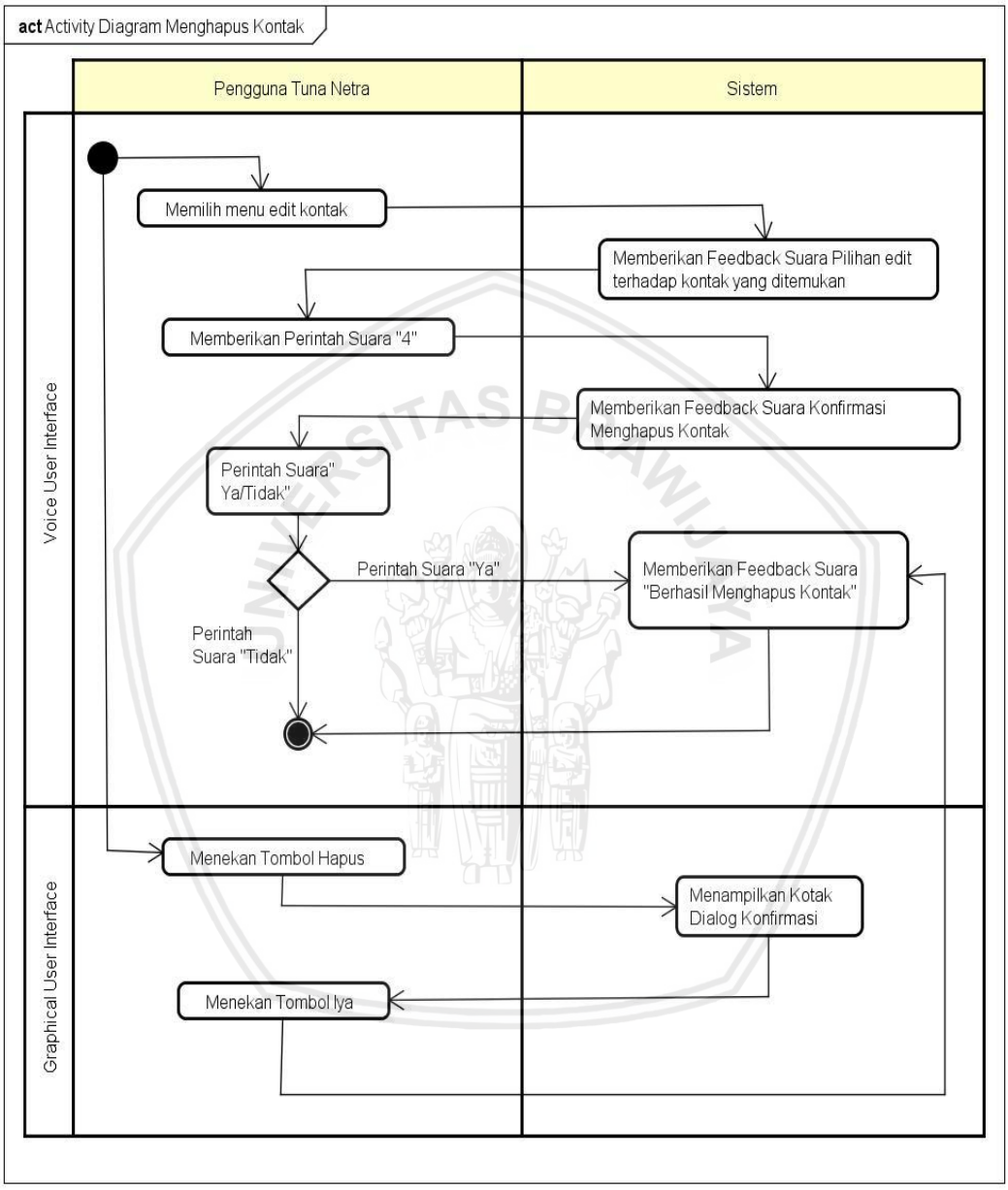


Gambar 5.4 *Activity Diagram* Mengedit Kontak

Gambar 5.4 merupakan *activity diagram* untuk mengedit kontak, di mana *activity diagram* ini dapat berlangsung apabila pengguna sudah melakukan pencarian kontak dan menentukan satu kontak yang akan diedit. Seperti yang dapat dilihat pada Gambar 5.4 proses mengedit kontak melibatkan pengguna tuna netra dan aplikasi. Ada dua pilihan untuk melakukan edit kontak bisa menggunakan *voice user interface* di mana pengguna memberikan perintah

melalui masukkan suara dan aplikasi memberikan respons atau *feedback* juga berupa suara dan bisa juga menggunakan *graphical user interface* seperti aplikasi lain pada umumnya.

4) *Activity Diagram* Menghapus Kontak

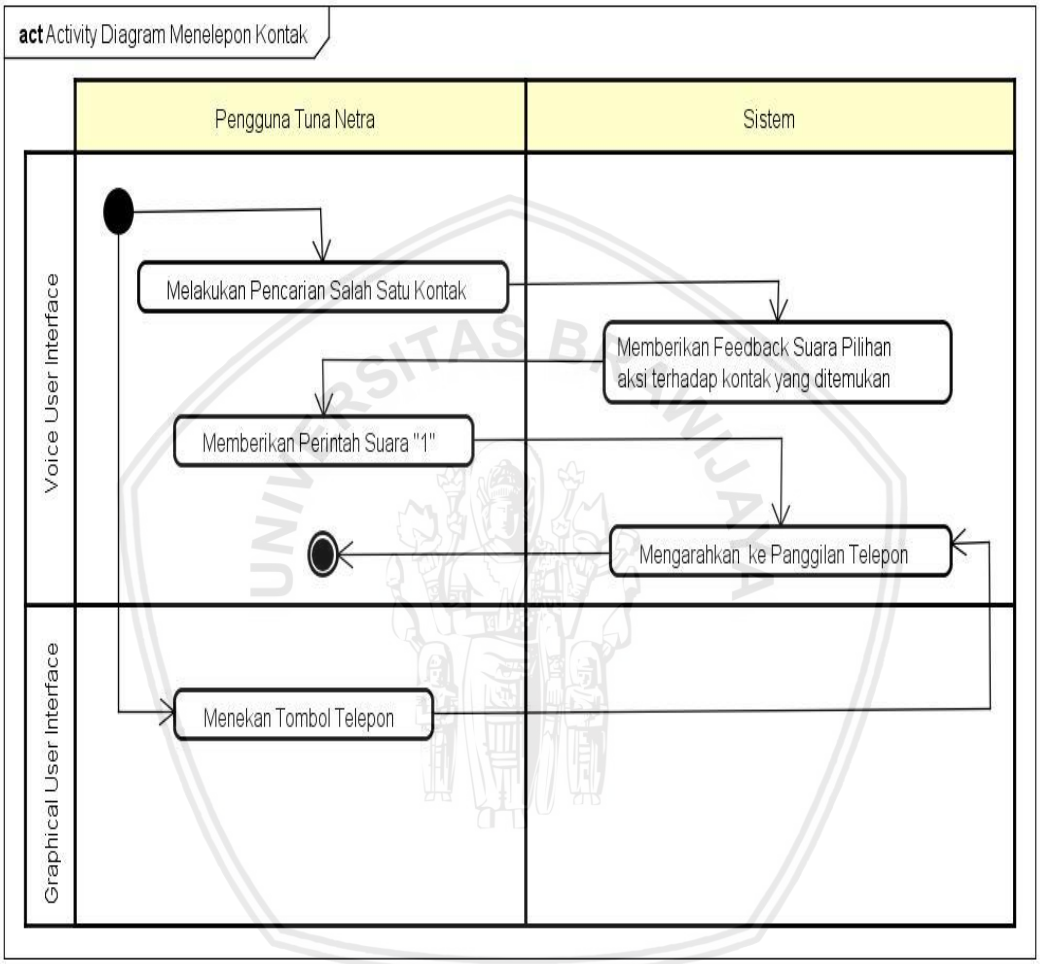


Gambar 5.5 *Activity Diagram* Menghapus Kontak

Gambar 5.5 merupakan *activity diagram* untuk menghapus kontak, di mana *activity diagram.* Seperti yang dapat dilihat pada Gambar 5.5 proses mengedit kontak melibatkan pengguna tuna netra dan aplikasi. Ada dua pilihan untuk melakukan hapus kontak bisa menggunakan *voice user interface* di mana pengguna memberikan perintah melalui masukkan suara dan aplikasi memberikan respons atau *feedback* juga berupa suara dan bisa juga menggunakan *graphical*

user interface seperti aplikasi lain pada umumnya. Untuk *activity diagram* pada *voice user interface* pengguna harus terlebih dahulu melakukan pencarian kontak dan sudah menemukan satu kontak yang akan dihapus, dan untuk *activity diagram graphical user interface,* pengguna harus berada pada halaman edit kontak pada salah satu kontak yang sudah terdaftar pada *smartphone* pengguna.

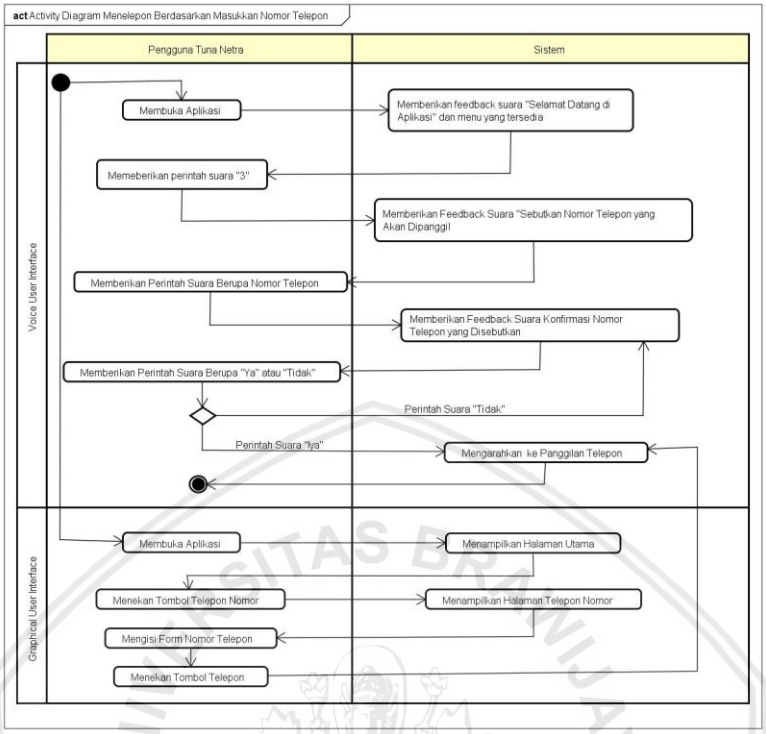
5) *Activity Diagram* Menelepon Kontak



Gambar 5.6 *Activity Diagram* Menelepon Kontak

Gambar 5.6 merupakan gambar *activity diagram* untuk melakukan menelepon kontak. Dalam Gambar 5.6 memodelkan proses telepon kontak yang melibatkan pengguna tuna netra dan aplikasi. Ada dua pilihan untuk melakukan menelepon kontak bisa menggunakan *voice user interface* di mana pengguna memberikan perintah melalui masukkan suara dan aplikasi memberikan respons atau *feedback* juga berupa suara dan bisa juga menggunakan *graphical user interface* seperti aplikasi lain pada umumnya. Untuk *activity diagram* pada *voice user interface* pengguna harus terlebih dahulu melakukan pencarian kontak dan sudah menemukan satu kontak yang akan ditelepom, dan untuk *activity diagram graphical user interface,* pengguna harus berada pada halaman edit kontak pada salah satu kontak yang sudah terdaftar pada *smartphone* pengguna.

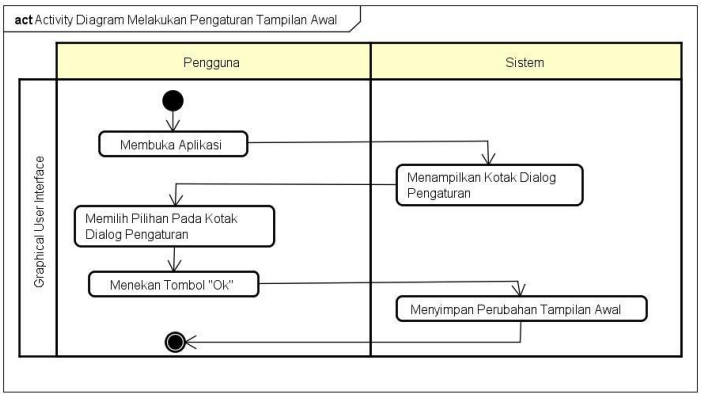
6) *Activity Diagram* Menelepon Berdasarkan Masukkan Nomor Telepon



Gambar 5.7 *Activity Diagram* Menelepon Berdasarkan Masukkan Nomor Telepon

Gambar 5.7 merupakan gambar *activity diagram* untuk melakukan panggilan melalui masukkan nomor telepon. Dalam Gambar 5.7 memodelkan proses telepon nomor yang melibatkan pengguna tuna netra dan aplikasi. Di mana pada *activity diagram* ini terdiri dari *voice user interface* dan *graphical user interface* di mana untuk *voice user interface* sistem akan berjalan sesuai dengan perintah suara yang dimasukkan oleh pengguna dan akan mengembalikan respons/*feedback* berupa suara kepada pengguna.

7) *Activity Diagram* Melakukan Pengaturan Tampilan Awal



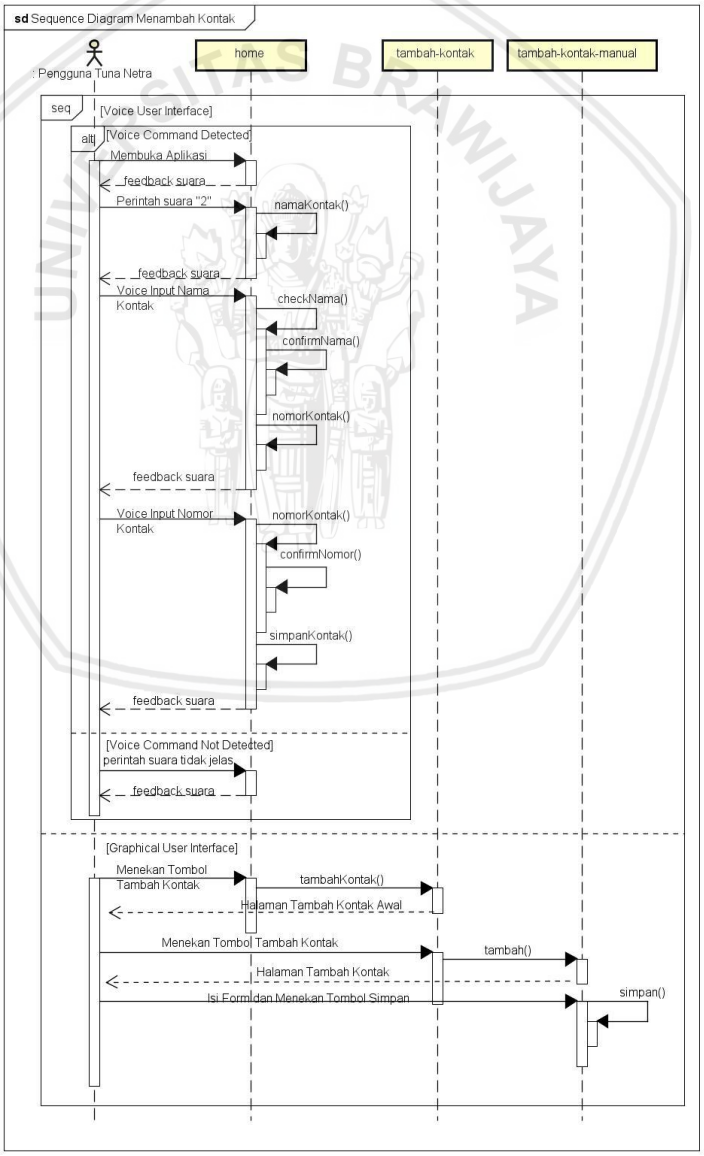
Gambar 5.8 *Activity Diagram* Melakukan Pengaturan Tampilan Awal

Gambar 5.8 merupakan *activity diagram* untuk melakukan proses perubahan pengaturan tampilan awal ketika aplikasi pertama dibuka. Di mana pada aplikasi ini hanya memanfaatkan tampilan *Graphical User Interface* saja dan yang dapat melakukan pengaturan tampilan awal pada aplikasi ini adalah pengguna namun tidak dibuat khusus untuk digunakan oleh pengguna tuna netra karena fungsi ini hanyalah fungsi tambahan yang sifatnya opsional yang disediakan oleh sistem.

5.3 Perancangan *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan urutan proses antar objek-objek yang ada dalam suatu aktivitas. Aktivitas yang dimaksud adalah aktivitas berdasarkan dari hasil analisis kebutuhan sistem yang dilakukan pada tahap sebelumnya.

1) *Sequence Diagram* Tambah Kontak



Gambar 5.9 *Sequence Diangram* Menambah Kontak

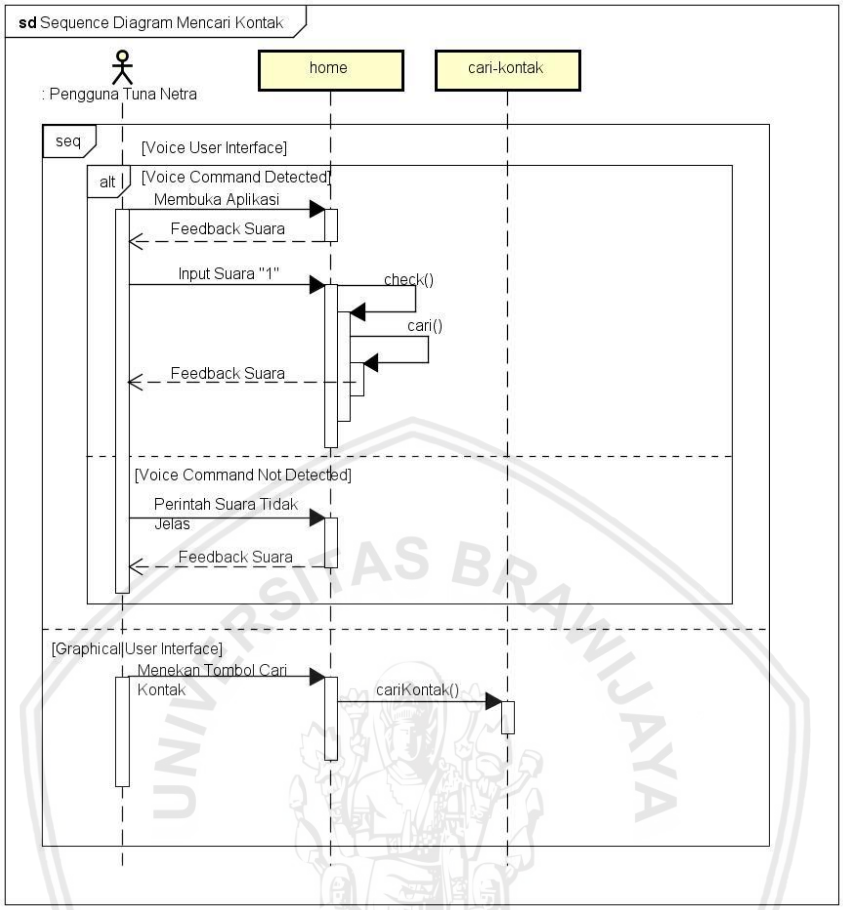
Gambar 5.9 merupakan *sequence diagram* untuk proses menambahkan kontak baru. Dapat dilihat dalam Gambar 5.9 *diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas menambah kontak. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*.

Untuk *voice user interface* yang hanya melibatkan satu *class* yaitu *class* home, proses dimulai ketika pengguna membuka aplikasi dan sistem akan memberikan *feedback* berupa suara, selanjutnya pengguna memasukkan perintah suara "2" dan pada *class* home akan dijalankan *method* namaKontak() dan hasil kembalian dari *method* tersebut berupa suara. Kemudian pengguna memberikan input suara berupa nama kontak dan pada *class* home akan dijalankan *method* checkNama() yang berfungsi untuk mengecek apakah nama yang diinputkan oleh pengguna sudah pernah ada tersimpan pada kontak pengguna dan memanggil *method* confirmNama() yang berfungsi untuk mengkonfirmasi nama yang telah disebutkan oleh pengguna.

Setelah proses memasukkan nama selesai, selanjutnya dilakukan proses memasukkan nomor kontak proses ini dimulai dengan pengguna memasukkan perintah suara berupa nomor kontak dan pada *class* home akan menjalankan *method* nomorKontak() dan *method* confirmNomor() untuk melakukan konfirmasi terhadap nomor kontak. Setelah nama dan nomor kontak sudah dimasukkan melalui perintah suara, maka selanjutnya *method* simpanKontak() akan dijalankan dan mengembalikan *feedback* berupa suara. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara pula.

Untuk proses melalui tampilan *graphical user interface* melibatkan 3 kelas dan 3 tampilan yang berbeda, yaitu *class* home, tambah-kontak dan tambah-kontak-manual. Pertama-tama pengguna menekan tombol tambah kontak yang ada pada halaman utama, selanjutnya *class* home akan menjalankan *method* tambahKontak() yang berfungsi untuk membuka halaman tambah kontak awal, setelah itu pengguna menekn kembali tombol tambah kontak yang ada pada halaman tambah kontak awal yang akan menjalankan *method* tambah() yang berfungsi untuk membuka halaman tambah kontak. Selanjutnya pengguna akan mengisikan for yang ada dan menekan tombol simpan pada halaman tambah kontak maka *method* simpan pada *class* tambah-kontak-manual akan dijalankan.

2) *Sequence Diagram* Cari Kontak



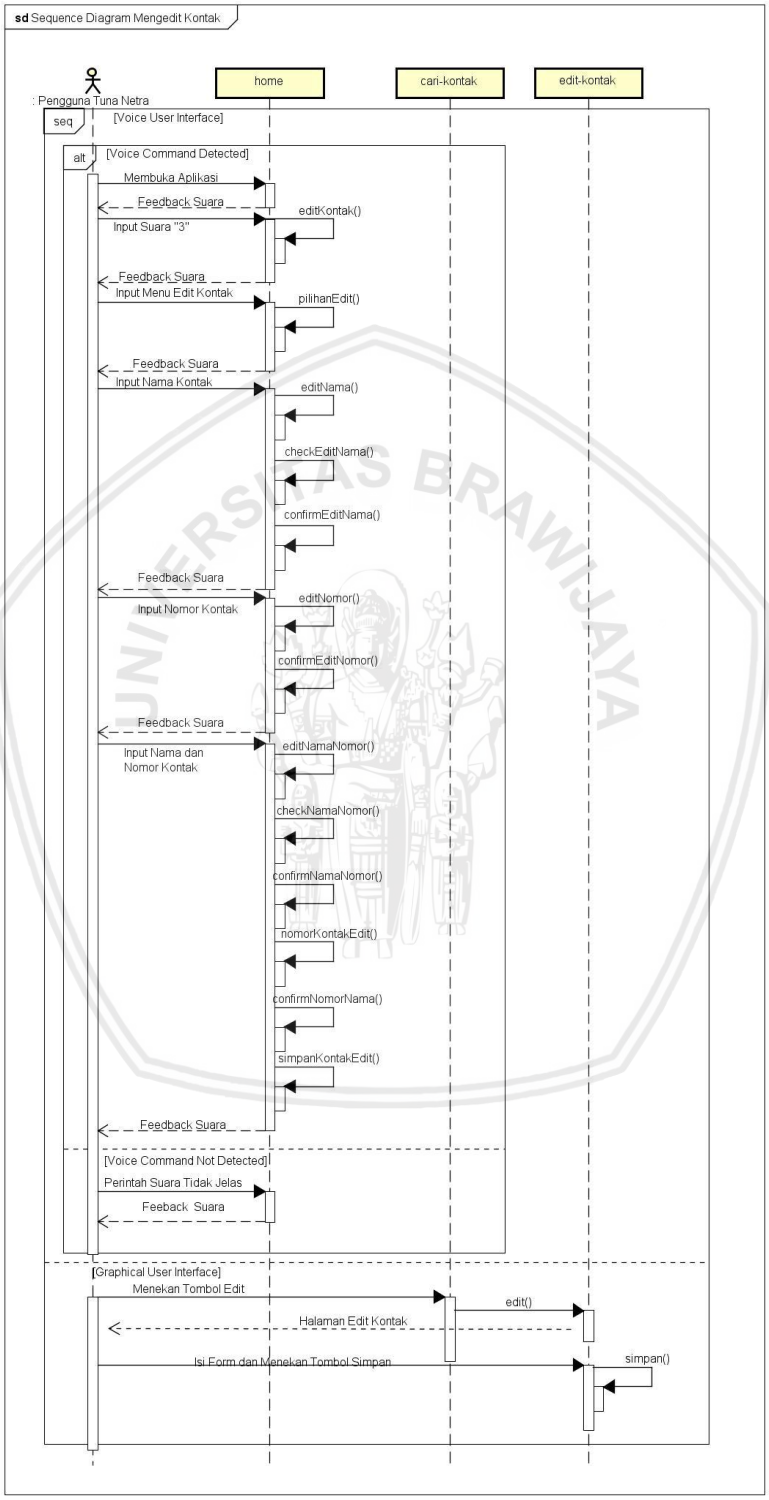
Gambar 5.10 *Sequence Diagram* Mencari Kontak

Gambar 5.10 merupakan *sequence diagram* untuk proses melakukan pencarian kontak. Dapat dilihat dalam Gambar 5.10 *diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas mencari kontak. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*. Untuk *voice user interface* yang hanya melibatkan satu *class* yaitu *class* home, proses dimulai ketika pengguna membuka aplikasi dan sistem akan memberikan *feedback* berupa suara, selanjutnya pengguna memasukkan perintah suara "1" dan pada *class* home akan dijalankan *method* check() dan cari() untuk melakukan pencarian data kontak dan hasil kembalian dari *method* tersebut berupa *feedback* suara. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara pula.

Untuk *graphical user interface* akan melibatkan 2 *class* yaitu home dan cari-kontak. Pertama-tama pengguna akan menekan tombol cari kontak yang ada pada halaman utama, selanjutnya *class* home akan menjalankan *method* cariKontak()

dan pada halaman cari kontak akan muncul kontak dimulai dari abjad paling awal yang ada pada kontak.

3) *Sequence Diagram* Mengedit Kontak



Gambar 5.11 *Sequence Diagram* Mengedit Kontak

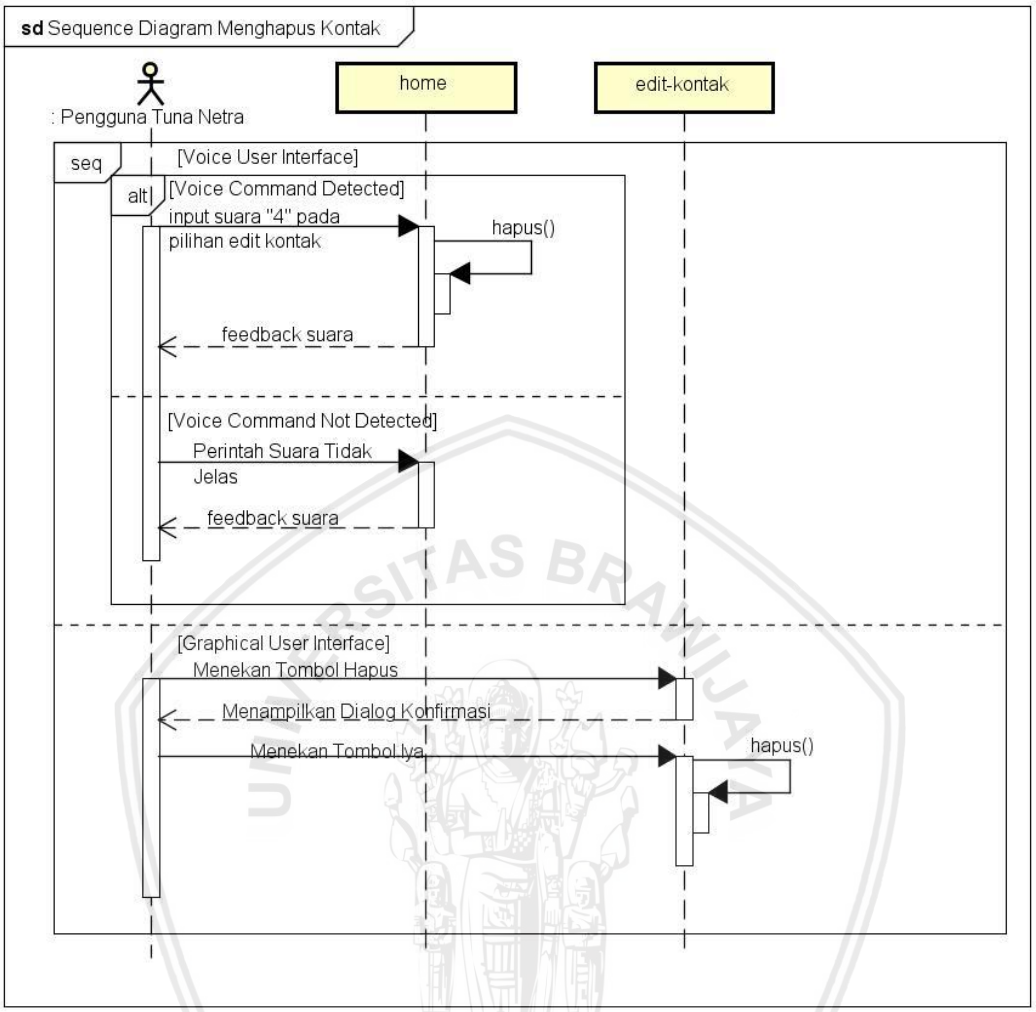
Gambar 5.11 merupakan *sequence diagram* untuk proses melakukan perubahan data kontak. Dapat dilihat dalam Gambar 5.11*diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas mengedit kontak. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*. Untuk interaksi menggunakan *voice user interface* hanya akan melibatkan satu *class* yaitu *class* home.

Untuk melakukan proses edit kontak, pengguna terlebih dahulu sudah melakukan proses mencari kontak dan menentukan satu kontak yang akan diedit dan menyebutkan perintah suara "2", kemudian *method* editKontak() pada *class* home akan dijalankan dan akan mengembalikan *feedback* berupa suara berisikan menu pada edit kontak yaitu edit nama saja, nomor saja dan edit keduanya. Apabila pengguna memilih menu untuk mengedit nama saja, maka yang akan pengguna diminta untuk memberikan masukkan berupa nama kontak dan *method* editNama(), checkEditNama() dan confirmEditNama() akan dijalankan dan mengembalikan *feedback* berupa suara.

Jika pengguna memilih pilihan untuk mengedit nomor saja maka pengguna akan diminta untuk memberikan perintah suara berupa nomor kontak dan *method* editNomor() dan confirmEditNomor() yang akan dijalankan serta mengembalikan *feedback* berupa suara. Sebaliknya, jika pengguna memilih menu untuk mengedit keduanya maka pengguna perlu memberikan masukan suara berupa nama dan sistem akan menyimpannya kemudian pengguna perlu memasukkan perintah suara berupa nama kontak, kemudian *method* editNamaNomor(), checkNamaNomor(), nomorKontakEdit(), confirmNomorNama() dan simpanKontakEdit() akan dijalankan dan memberikan kembalian berupa *feedback* suara. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara pula.

Untuk *graphical user interface* akan melibatkan 2 *class* yaitu cari-kontak dan edit-kontak. Pertama-tama pengguna menekan tombol edit pada halaman cari kontak, kemudian *method* edit akan dijalankan dan akan mengembalikan tampilan pengguna pada halaman edit kontak. Kemudian pengguna mengisikan *form* untuk melakukan edit kontak di mana pada *form* tersebut sebelumnya telah terisi dengan data kontak yang dipilih, pengguna dapat memilih untuk melakukan perubahan data kontak baik nama saja, nomor telepon saja, maupun mengubah kedua data pada *form* yang disediakan. Apabila pengguna sudah melakukan perubahan data kontak yang dilakuakn maka ketika pengguna menekan tombol simpan makan lalu pada *class* edit-kontak akan dijalankan *method* simpan() untuk menyimpan perubahan yang dilakukan oleh pengguna.

4) *Sequence Diagram* Menghapus Kontak



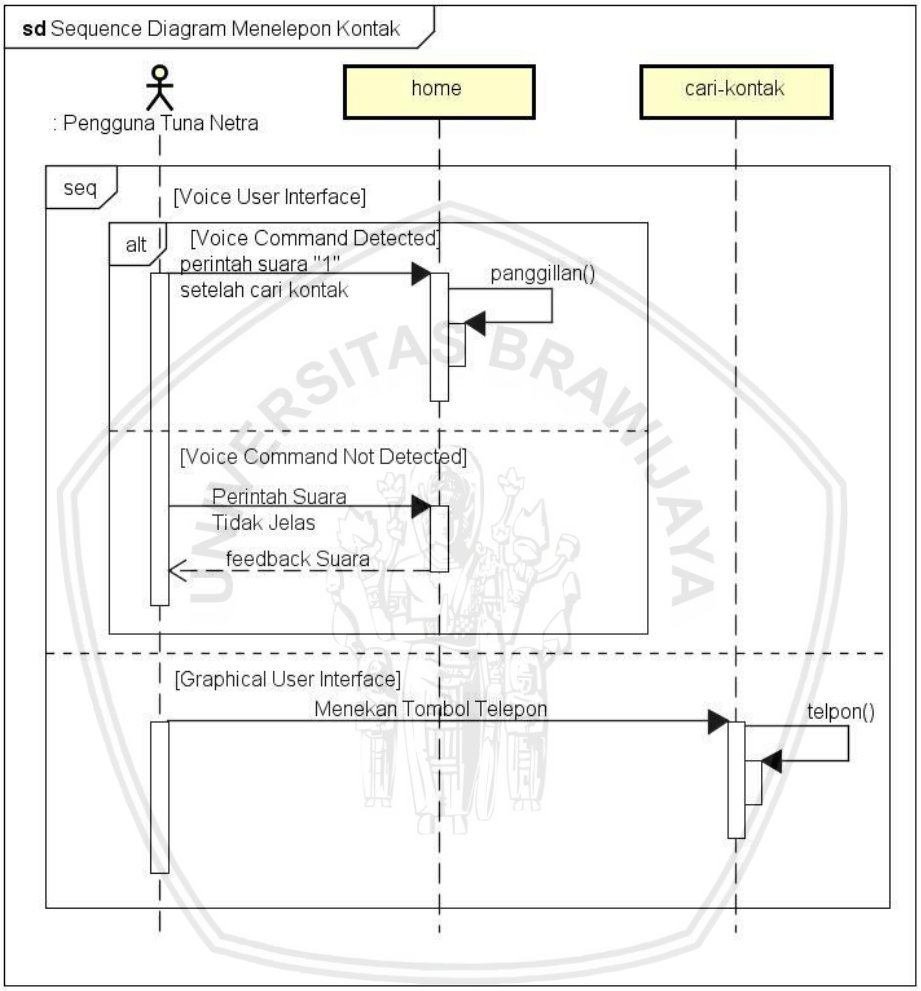
Gambar 5.12 *Sequence Diagram* Menghapus Kontak

Gambar 5.12 merupakan *sequence diagram* untuk proses melakukan penghapusan kontak. Dapat dilihat dalam Gambar 5.12 *diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas menghapus kontak. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*. Untuk interaksi menggunakan *voice user interface* hanya akan melibatkan satu *class* yaitu *class* home.

Untuk melakukan proses edit kontak, pengguna terlebih dahulu sudah melakukan proses mencari kontak dan menentukan satu kontak yang akan dihapus dan menyebutkan perintah suara “3”, selanjutnya *method* hapusKontak() pada *class* home akan dijalankan dan memberikan nilai kembalian berupa suara. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara.

Pada *graphical user interface* juga hanya melibatkan satu *class* yaitu edit-kontak pengguna menekan tombol hapus yang ada pada halaman edit kontak kemudian akan muncul dialog konfirmasi, dan pengguna menekan tombol iya maka selanjutnya *method* hapus() akan dijalankan dan sistem akan menghapus kontak tersebut.

5) *Sequence diagram* Menelepon Kontak



Gambar 5.13 *Sequence Diagram* Menelepon Kontak

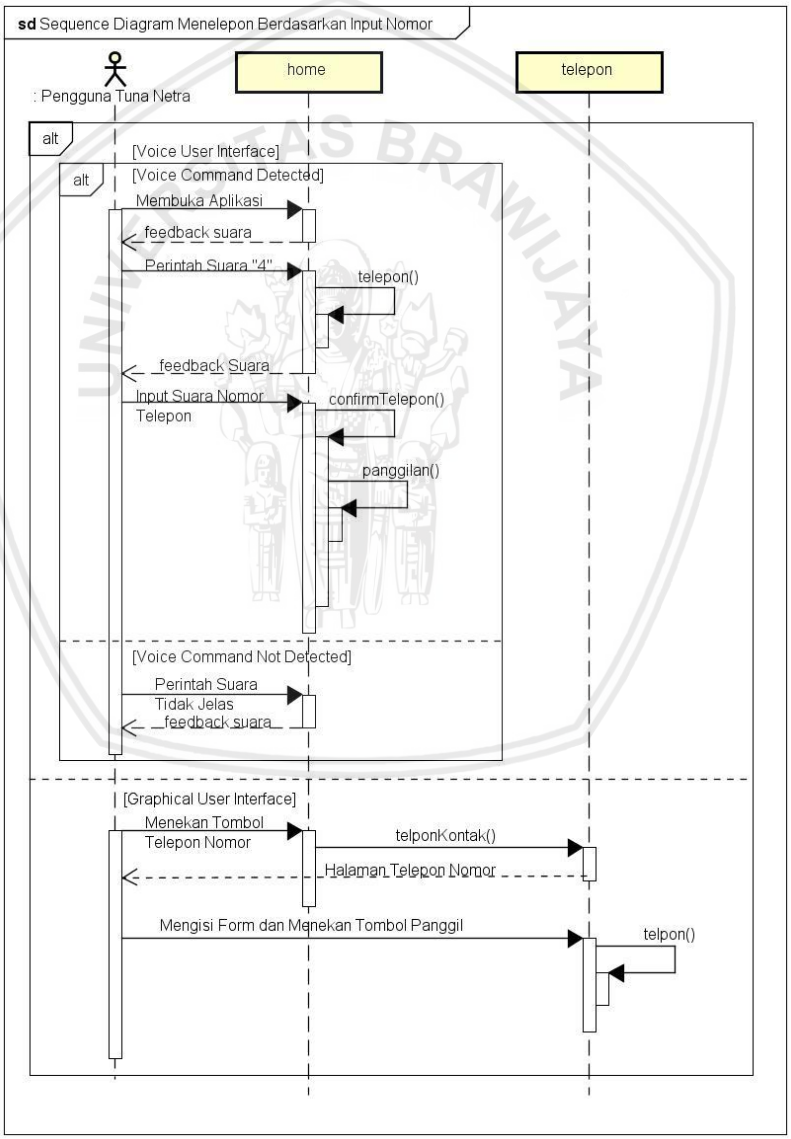
Gambar 5.13 merupakan *sequence diagram* untuk proses melakukan telepon kontak. Dapat dilihat dalam Gambar 5.13 *diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas menelepon kontak. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*. Untuk interaksi menggunakan *voice user interface* hanya akan melibatkan satu *class* yaitu *class* home.

Untuk melakukan proses edit kontak, pengguna terlebih dahulu sudah melakukan proses mencari kontak dan menentukan satu kontak yang akan ditelepon dan menyebutkan perintah suara “1”, maka sistem akan menjalankan

method panggilan() pada halaman home dan mengarahkan tampilan pengguna ke tampilan panggilan pada *smartphone* pengguna dan menutup aplikasi. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara.

Sedangkan pada tampilan *graphical user interface* juga hanya akan melibatkan satu *class* yaitu cari-kontak di mana pengguna menekan tombol telepon yang ada pada halaman cari kontak. Lalu *method* telpon() akan dijalankan dan mengarahkan tampilan pengguna ke tampilan panggilan pada *smartphone* pengguna dan menutup aplikasi.

6) *Sequence diagram* Menelepon Berdasarkan Input Nomor



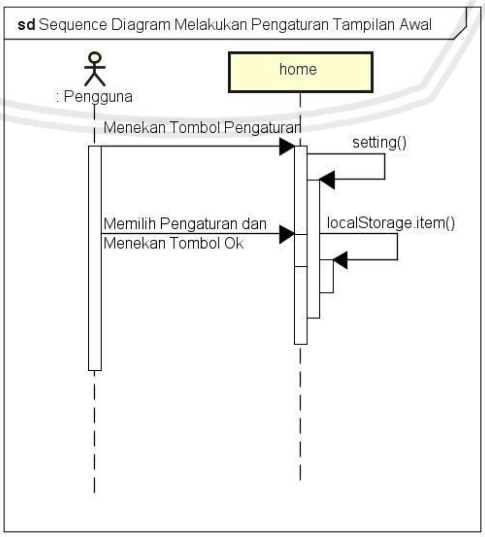
Gambar 5.14 *Sequence Diagram* Menelepon Berdasarkan Input Nomor

Gambar 5.14 merupakan *sequence diagram* untuk proses melakukan panggilan berdasarkan masukkan nomor telepon. Dapat dilihat dalam Gambar 5.13 *diagram* ini menjelaskan proses interaksi objek-objek yang terlihat pada aktivitas menelepon berdasarkan input nomor. Aktor yang terlibat adalah pengguna tuna netra. Dan pada sequence diagram ini menjelaskan mengenai interaksi objek baik melalui *voice user interface* ataupun *graphical user interface*. Untuk interaksi menggunakan *voice user interface* hanya akan melibatkan satu *class* yaitu *class* home.

Pertama-tama saat pengguna membuka aplikasi maka sistem akan memberikan *feedback* suara berupa menu yang tersedia, selanjutnya pengguna akan memasukkan perintah suara "3" dan *method* telepon() akan dijalankan dan akan diberikan *feedback* suara untuk meminta pengguna memasukkan perintah suara berupa nomor telepon yang akan dipanggil, selanjutnya pengguna memasukkan perintah suara berupa nomor telepon yang akan dipanggil dan sistem akan menjalankan *method* confirmTelepon() dan panggilan() yang ada pada *class* home. Namun apabila pengguna menginputkan perintah suara yang tidak jelas baik dari pengucapan maupun perintah suara yang tidak sesuai dengan yang ada dalam sistem, maka sistem akan mengembalikan *feedback* berupa suara.

Untuk tampilan *graphical user interface* melibakan 2 *class* yaitu home dan telepon, pertama-tama pengguna akan menekan tombol telepon nomor yang ada pada halaman utama kemudian sistem akan menjalankan *method* telponKontak() dan mengarahkan tampilan pengguna ke halaman telepon nomor. Kemudian pengguna mengisikan *form* berupa nomor telepon yang akan dipanggil dan kemudian menekan tombol panggil, selanjutnya sistem akan menjalankan *method* telpon() dan mengarahkan tampilan pengguna ke tampilan panggilan pada *smartphone* pengguna dan menutup aplikasi.

7) *Sequence diagram* Menelepon Berdasarkan Input Nomor

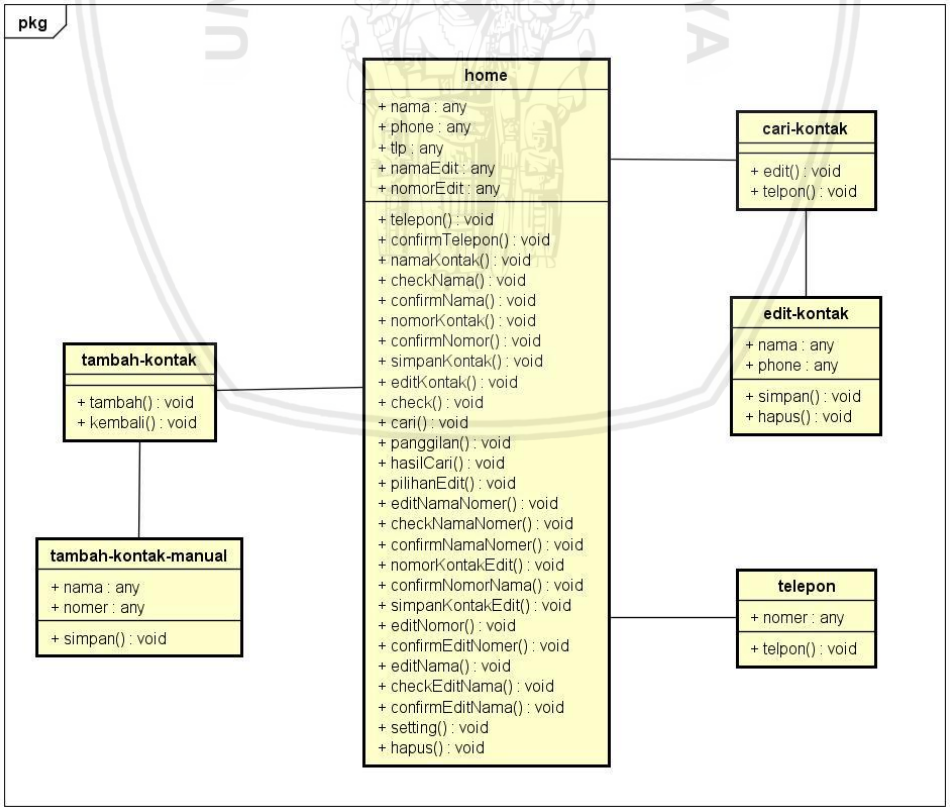


Gambar 5.15 *Sequence Diagram* Melakukan Pengaturan Tampilan Awal

Gambar 5.15 merupakan *sequence diagram* yang dilakukan untuk mengubah pengaturan tampilan awal pada sistem. Di mana pada *sequence diagram* ini hanya melibatkan satu *class* yaitu home, dan ketika pengguna menekan tombol pengaturan maka sistem akan menjalankan *method* setting() dan ketika tombol ok yang ditekan maka method localStorage.Item() yang akan dijalankan.

5.4 Perancangan *Class Diagram*

Class Diagram merupakan sebuah model statis dalam perancangan UML yang digunakan untuk memodelkan sebuah sistem berdasarkan dari sudut pandang *class-class* dan relasi antar *class* yang membentuk suatu sistem. Tiap *class* berfungsi sebagai *blueprint* (cetak biru) dari setiap objek-objek yang saling berinteraksi seperti pada *sequence diagram*. Pada sebuah *class* tersusun atas *attribute* atau variabel yang ada dalam *class tersebut* dan *behavior* atau *method-method* yang terdapat dalam *class* tersebut, dan juga dapat tersusun dari relasi atar *class* yang dimiliki oleh *class* tersebut. Relasi antar *class* dalam sebuah *class diagram* digambarkan dalam sebuah simbol garis penghubung atau yang biasa dikenal juga dengan sebutan istilah *link*, setiap simbol garis dalam *class* diagram bisa berbeda-beda jenisnya tergantung juga pada relasi yang terdapat pada *class* tersebut seperti misalnya relasi asosiasi, agregasi, generalisasi dan lain-lain. Berikut ini merupakan rancangan *class* diagram dari sistem yang akan dirancang:



Gambar 5.16 *Class Diagram*

Gambar 5.16 merupakan *class* diagram pada sistem ini dapat dilihat terdapat 6 *class* yaitu home, tambah-kontak, tambah-kontak-manual, cari-kontak, edit-

kontak dan telepon. Di mana pada *class* home terdapat 5 atribut yaitu nama, phone, tlp, namaEdit, dan nomor Edit dan pada *class* ini juga terdapat 25 *method yang* akan mengurusi *voice user interface* pada awal aplikasi dibuka dan juga menampilkan halaman utama aplikasi, *class* tambah-kontak yang akan mengurusi halaman tambah kontak awal di mana pada *class* ini terdapat 2 *method* tambah() yang akan mengarahkan kontak ke halaman tambah-kontak-manual dan *method* kembali() yang akan mengurusi tampilan pengguna ke halaman utama *class* dari tambah-kontak. *Class* tambah-kontak-manual memiliki relasi asosiasi dengan tambah-kontak-manual dan *class* home, selanjutnya terdapat *class* cari-kontak yang akan mengurusi halaman cari kontak yang memiliki 2 atribut yaitu nama dan nomor, serta sebuah *method* simpan(), *class* ini memiliki relasi asosiasi dengan *class home*. Kemudian terdapat *class* edit-kontak yang akan mengurusi proses edit dan menghapus kontak dan halaman telepon yang akan mengurusi proses menelepon berdasarkan masukkan nomor telepon. *Class* edit-kontak memiliki 2 atribut yaitu nama dan phone, dan memiliki 2 *method* yaitu simpan() dan hapus().Sedangkan *class* telepon yang berfungsi untuk mengurusi pemanggilan kontak berdasarkan masukkan nomor telepon secara manual, pada *class* ini terdapat sebuah atribut nomer dan sebuah *method* telpon(), *class* ini memiliki relasi asosiasi dengan *class* home.

5.5 Perancangan Basis Data

Basis data merupakan tempat penyimpanan data aplikasi. Berdasarkan kebutuhan penggunaan pada sistem yang dibangun, tidak diperlukannya membuat basis data secara terpisah, dikarenakan hanya membutuhkan penyimpanan nama dan nomor kontak telepon. Sehingga untuk basis data langsung menyimpan pada *storage local* pada *smartphone* pengguna, selain itu kelebihan dari penyimpanan data langsung pada *storage local* adalah data kontak yang sebelumnya sudah ada/terdaftar pada *smartphone* pengguna bisa langsung tersinkronisasi dengan sistem. Penggunaan penyimpanan data pada sistem ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Basis Data Sistem

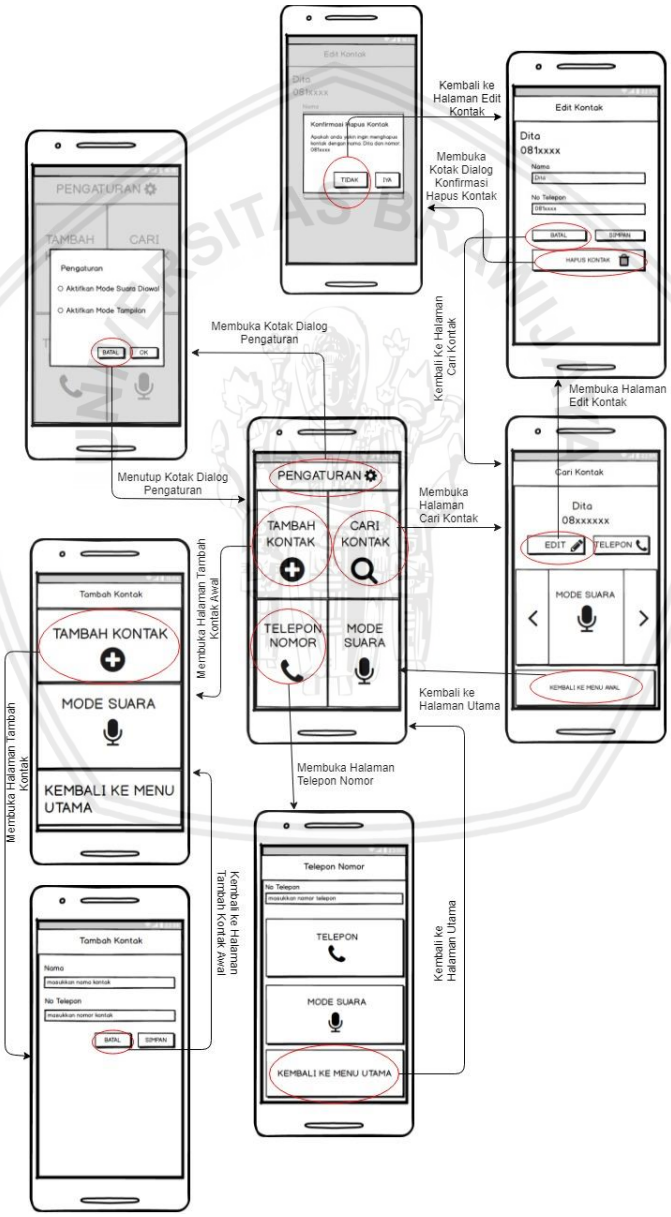
Param	***Type***	***Detail***
displayName	string	*The name of this Contact, suitable for display to end users.*
phoneNumbers	IContactField[]	*An array of all the contact's phone numbers.*

Tabel 4.5 merupakan tabel basis data yang digunakan pada sistem ini, terdapat 3 kolom yaitu kolom Param yang berisikan nama parameter yang digunakan yang terdiri dari *displayName* dan *phoneNumbers*, selanjutnya kolom *Type* yang menjelaskan tipe data dari parameter yaitu untuk *displayName* tipe data yang digunakan adalah string, sedangkan untuk *phoneNumbers* tipe data yang digunakan adalah IContactField[]. Sedangkan kolom *Detail* menjelaskan penjelaskan dari fungsi parameter yang digunakan yaitu untuk *displayName*

merupakan nama dari kontak, sedangkan *phoneNumbers* merupakan sebuah array untuk menyimpan nomor kontak.

5.6 Perancangan *Screenflow*

Perancangan *Screenflow* merupakan aliran perubahan tampilan antarmuka yang akan muncul pada layar *smartphone* pengguna. Tujuan dari pembuatan *screenflow* adalah sebagai gambaran dari navigasi perpindahan antarmuka sistem yang akan dirancang. Pada sistem ini proses perancangan *screenflow* dirancang berdasarkan hasil perancangan antarmuka yang telah diperoleh dari tahap sebelumnya.



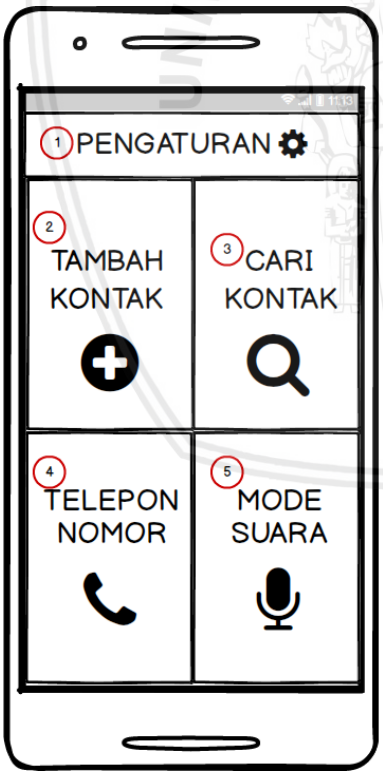
Gambar 5.17 Rancangan *Screenflow*

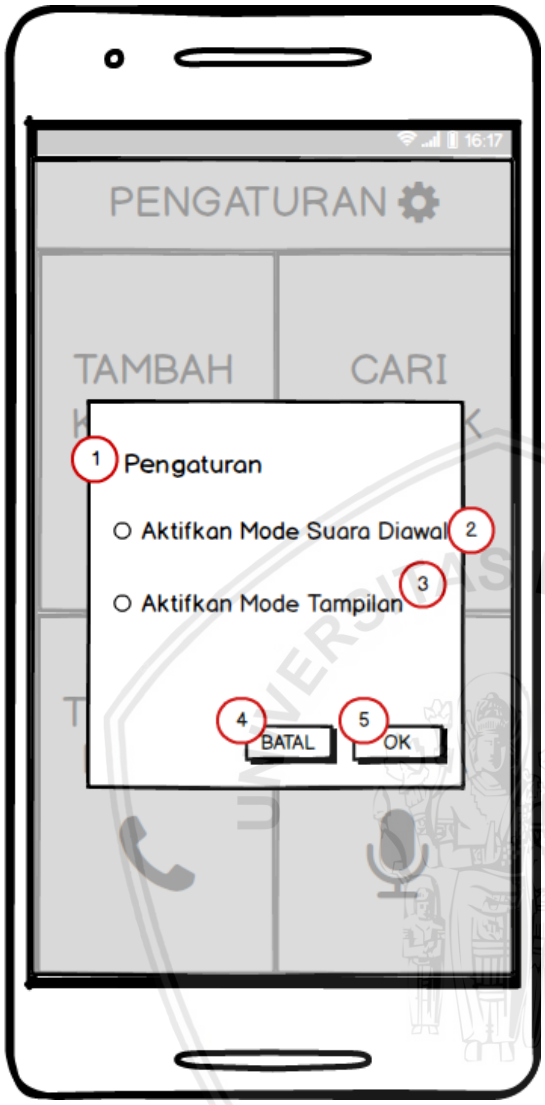
Gambar 5.17 merupakan rancangan *screenflow* atau perpindahan halaman yang ada pada sistem yang akan dirancang di mana terdapat 6 halaman yaitu halaman utama, halaman telepon nomor, halaman tambah kontak awal, halaman tambah kontak, halaman cari kontak, dan halaman edit kontak, serta terdapat pula 2 kotak dialog yaitu kotak dialog untuk melakukan hapus kontak dan kotak dialog untuk melakukan pengaturan tampilan awal pada aplikasi. Untuk perpindahan tiap halaman dan halaman yang akan dituju dapat dilihat seperti dalam Gambar 5.14.

5.7 Perancangan *Wireframe*

Perancangan *wireframe* berfungsi sebagai perancangan kasar yang bersifat *low-fidelity* berdasarkan bagaimana antarmuka atau tampilan sistem akan dirancang dan menjadi acuan dalam mengembangkan antarmuka sistem ke tahap perancangan *mockup.*Perancangan *wireframe* sistem dapat dilihat seperti pada Tabel 5.2.

Tabel 5.2 Perancangan *Wireframe* Sistem

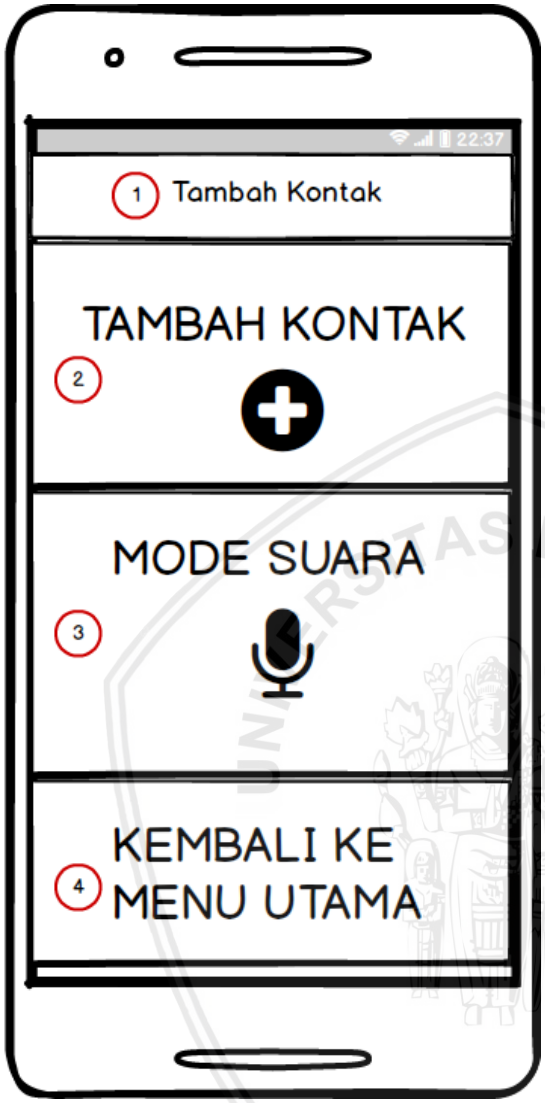
Wireframe	Keterangan
 **Gambar Rancangan *Wireframe* Halaman Utama**	Pada Gambara Rancangan *Wireframe* Halaman Utama terdapat 5 elemen pada halaman ini, elemen-elemen tesebut adalah sebagai berikut. 1. Tombol untuk melakukan pengaturan *mode* tampilan awal aplikasi dengan tulisan "pengaturan" dan sebuah *icon setting*. 2. Tombol untuk melakukan tambah kontak secara manual dengan sebuah *icon add*. 3. Tombol untuk melakukan cari kontak secara manual dan sebuah *icon search*. 4. Tombol untuk melakukan telepon berdasarkan masukkan nomor dan sebuah *icon phone.* 5. Tombol untuk mengaktifkan mode *suara* pada sistem dan sebuah *icon microphone.*



Gambar Rancangan *Wireframe* Kotak Dialog Pengaturan

Pada Gambar Rancangan *Wireframe* Kotak Dialog Pengaturan terdapat 4 elemen yang ada pada kotak dialog pengaturan. Elemen-elemen tersebut adalah:

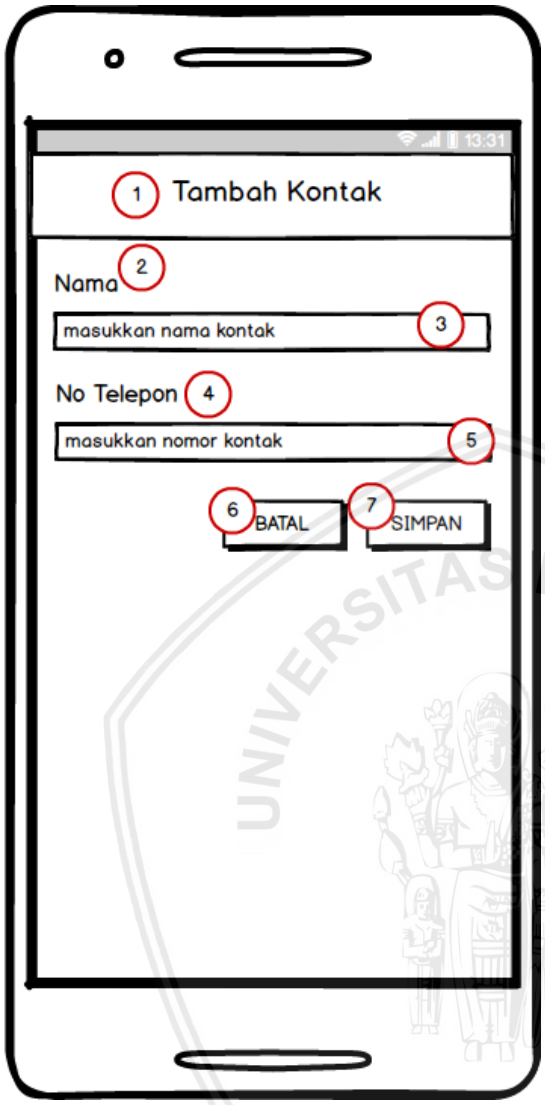
1. Tulisan "Pengaturan" yang menunjukkan judul dari kotak dialog ini.
2. *Radiobutton* dengan perintah "Aktifkan Model Suara Diawal" yang berfungsi untuk menjalankan mode suara di awal tampilan halaman utama ketika sistem pertama kali dijalankan.
3. *Radionbutton* dengan perintah "Aktifkan Mode Tampilan" yang akan menampilkan hanya tampilan berupa tulisan, tombol dan *icon* seperti pada Gambar 5.14.
4. Tombol dengan tulisan "Batal" yang berfungsi untuk menutup kotak dialog.
5. Tombol dengan tulisan "Ok" yang berfungsi untuk menyimpan perubahan yang dilakukan pada kotak dialog.



Gambar Rancangan *Wireframe* Halaman Tambah Kontak Awal

Pada Gambar Rancangan Wireframe Halaman Tambah Kontak Awal terdapat 4 elemen yang ada pada halaman ini, elemen-elemen tesebut adalah:

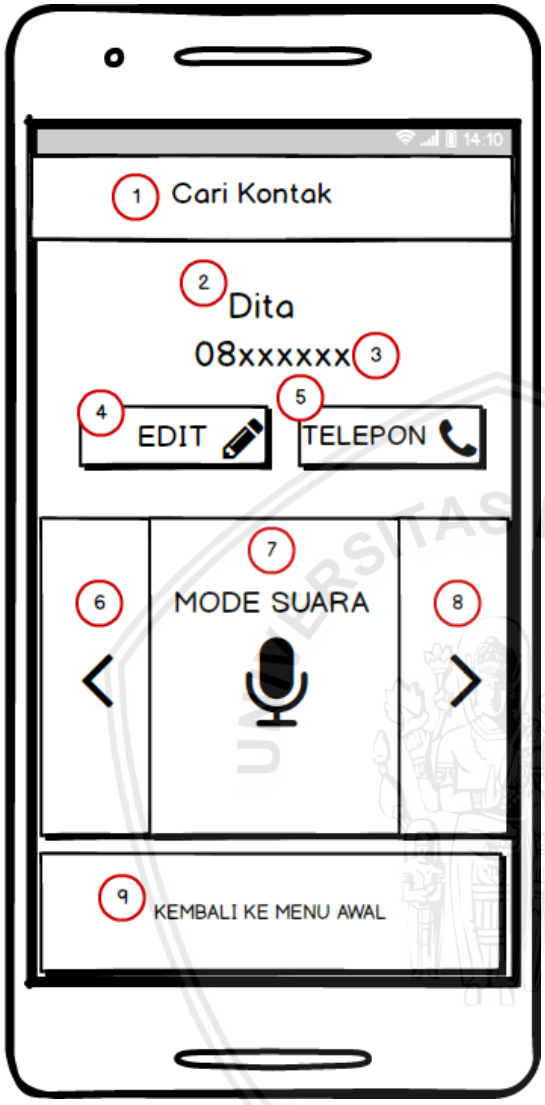
1. *Title Bar* yang merupakan judul dari halaman ini yang bertuliskan "Tambah Kontak".
2. Tombol dengan tulisan "Tambah Kontak" yang berfungsi untuk menjalankan halaman tambah kontak secara manual dan sebuah *icon add*.
3. Tombol dengan tulisan "Mode Suara" yang berfungsi untuk menjalankan mode suara pada sistem dan sebuah *icon microphone*
4. Tombol dengan tulisan "Kembali Ke Menu Utama" yang berfungsi untuk mengembalikan tampilan ke halaman utama



Gambar Rancangan *Wireframe* Halaman Tambah Kontak dengan Masukkan Data Secara Manual

Pada Gambar Rancangan *Wireframe* Halaman Tambah Kontak dengan Masukkan Data Secara Manual terdapat 6 elemen pada halaman ini, elemen-elemen tersebut yaitu:

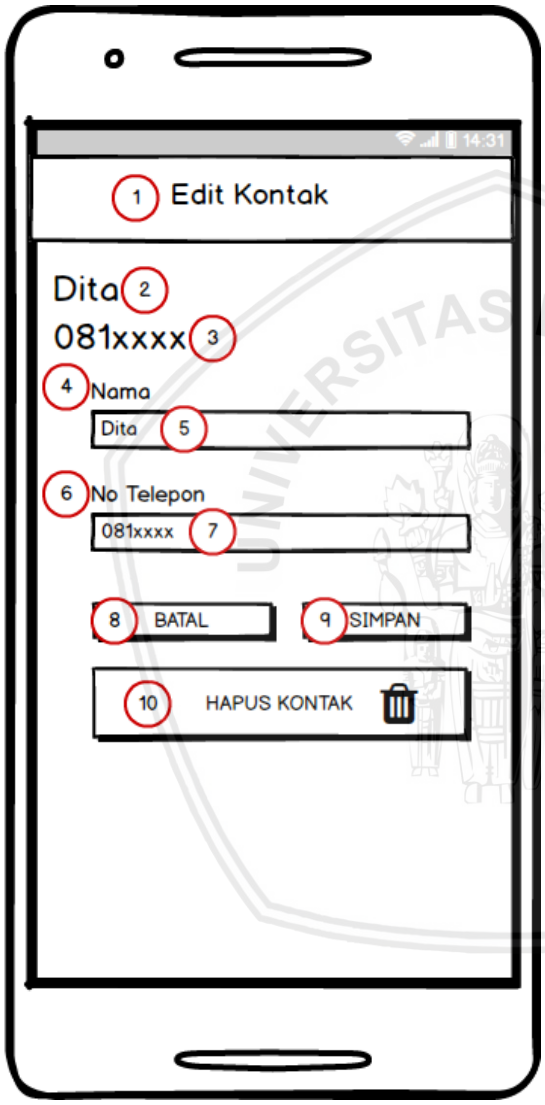
1. *Title Bar* yang merupakan judul dari halaman ini yang bertuliskan "Tambah Kontak".
2. Label dengan tulisan "Nama" yang merupakan keterangan dari masukkan teks pada *text input* yang ada dibawahnya.
3. *Text input* yang berfungsi untuk mengisikan data nama kontak.
4. Label dengan tulisan "No Telepon" yang merupakan keterangan dari masukkan teks pada *textfield* yang ada dibawahnya.
5. *Text input* yang berfungsi untuk mengisikan data nomor kontak.
6. Tombol "Batal" yang berfungsi untuk membatalkan aksi pengguna dan mengarahkan halaman ke halaman tambah kontak awal
7. Tombol "Simpan" yang berfungsi untuk menyimpan aksi pengguna.



Gambar Rancangan *Wireframe* Halaman Cari Kontak

Pada Gambar Rancangan *Wireframe* Halaman Cari Kontak terdapat 9 elemen pada halaman ini, elemen-elemen tesebut adalah:

1. *Title Bar* yang merupakan judul dari halaman ini yang bertuliskan "Cari Kontak".
2. *Label* yang menampilkan nama kontak.
3. *Label* yang menampilkan nomor kontak.
4. Tombol dengan tulisan "Edit" dan *icon edit* yang berfungsi untuk menampilkan halaman edit kontak.
5. Tombol dengan tulisan "Telepon" dan *icon call* yang berfungsi untuk melakukan panggilan terhadap kontak tersebut.
6. Tombol dengan *icon previous* yang berfungsi untuk menampilkan kontak dengan huruf abjad sebelumnya.
7. Tombol dengan tulisan "Mode Suara" dan *icon microphone* yang berfungsi untuk mengaktifkan mode suara
8. Tombol dengan *icon next* yang berfungsi untuk menampilkan kontak dengan huruf abjad setelahnya.
9. Tombol dengan tulisan "Kembali ke menu awal" yang akan mengarahkan tampilan ke menu awal.



Gambar Rancangan *Wireframe* Halaman Edit Kontak

Pada Gambar Rancangan *Wireframe* Halaman Edit Kontak terdapat 10 elemen pada halaman ini, elemen-elemen tesebut adalah:

1. *Title Bar* yang merupakan judul dari halaman ini yang bertuliskan "Edit Kontak".
2. *Label* dengan isi nama dari kontak yang ditampilkan pada halaman ini.
3. *Label* dengan isi nomor kontak yang ditampilkan pad halaman ini.
4. Label dengan tulisan "Nama" yang merupakan keterangan dari masukkan teks pada *text input* yang ada dibawahnya.
5. *Text input* yang berfungsi untuk mengisikan data nama kontak.
6. Label dengan tulisan "No Telepon" yang merupakan keterangan dari masukkan teks pada *textfield* yang ada dibawahnya.
7. *Text input* yang berfungsi untuk mengisikan data nomor kontak.
8. Tombol "Batal" yang berfungsi untuk membatalkan aksi pengguna dan mengarahkan halaman ke halaman utama
9. Tombol "Simpan" yang berfungsi untuk menyimpan aksi pengguna.
10. Tombol "Hapus Kontak" dan *icon delete* yang berfungsi untuk menghapus kontak tersebut.

Gambar Rancangan *Wireframe* Kotak Dialog Konfirmasi Hapus Kontak

Pada Gambar Rancangan *Wireframe* Kotak Dialog Konfirmasi Hapus Kontak terdapat 4 elemen pada halaman ini. Elemen-elemen tersebut adalah:

1. Tulisan "Konfirmasi Hapus Kontak" yang merupakan judul dari kotak dialog konfirmasi.
2. Tulsan "Apakah anda yakin menghapus kontak dengan nama: <nama kontak> dan nomor: <nomor kontak>" yang merupakan isi dari kotak dialog konfirmasi.
3. Tombol dengan tulisan "Tidak" yang berfungsi untuk menutup kotak dialog konfirmasi.
4. Tombol dengan tulisan "Iya" yang berfungsi untuk menyimpan perubahan yang dilakukan oleh pengguna.

Gambar Rancangan *Wireframe* Halaman Telepon Berdasarkan Masukkan Nomor Telepon

Pada Gambar Rancangan *Wireframe* Halaman Telepon Berdasarkan Masukkan Nomor Telepon terdapat 6 elemen pada halaman ini. Elemen-elemen tersebut adalah:

1. *Title Bar* yang merupakan judul dari halaman ini yang bertuliskan "Telepon Nomor".
2. Label dengan tulisan "No Telepon" yang merupakan keterangan dari masukkan teks pada *text input* yang ada dibawahnya.
3. *Text input* yang berfungsi untuk mengisikan data nomor telepon.
4. Tombol dengan tulisan "Telepon" dan *icon call* yang berfungsi untuk memanggil nomor yang sudah diinputkan sebelumnya.
5. Tombol dengan tulisan "Mode Suara" dan *icon microphone* yang berfungsi untuk menjalankan mode suara.
6. Tombol dengan tulisan "Kembali Ke Menu Utama" yang berfungsi untuk mengarahkan tampilan ke menu utama.

5.8 Perancangan *Mockup*

Perancangan *mockup* bertujuan untuk memodelkan sebuah sistem dari sisi antarmuka yang ditampilkan kepada pengguna dalam bentuk yang menyerupai tampilan yang akan diimplementasi baik dari segi tata letak dan warna. Pada perancangan *mockup* terdapat rancangan dari komponen dan elemen yang dipakai pada sistem. Pada perancangan *mockup* pada sistem ini dibuat sedemikian rupa agar mudah digunakan oleh pengguna tuna netra sebagian yang berusia lansia dan terdapat pula pilihan untuk mengaktifkan mode suara.

1. Perancangan *Mockup* Halaman Utama

Gambar 5.18 Rancangan *Mockup* Halaman Utama

Gambar 5.18 merupakan rancangan *mockup* dari halaman utama. Pada halaman ini terdapat 5 tombol. Masing-masing tombol digunakan untuk menjalankan fungsionalitas sistem. Pada bagian atas terdapar tombol pengaturan yang berfungsi untuk menampilkan kotak dialog pengaturan tampilan awal sistem. Sedangkat terdapat juga 4 tombol lain yaitu tombol tambah kontak yang berfungsi

untuk menjalankan fungsionalitas menambah kontak, tombol cari kontak yang berfungsi untuk menjalankan fungsionalitas mencari kontak, tombol telepon nomor yang berfungsi untuk menjalankan fungsionalitas menelepon berdasarkan input nomor, dan tombol mode suara yang berfungsi untuk mengaktifkan mode suara pada sistem ini.

2. Perancangan *Mockup* Kotak Dialog Pengaturan

Gambar 5.19 Rancangan *Mockup* Kotak Dialog Pengaturan

Gambar 5.19 merupakan rancangan *mockup* untuk kotak dialog pengaturan yang muncul pada halaman utama ketika tombol pengaturan ditekan. Seperti yang dapat dilihat dalam Gambar 5.19 judul dari kotak dialog ini adalah pengaturan, pada kotak dialog ini juga terdapat 2 *radio button* yang berfungsi untuk memilih pengaturan pada tampilan awal di sistem. Terdapat 2 pilihan dalam bentuk *radio button* yaitu pilihan untuk Aktifkan Mode Suara di Awal atau Aktifkan Mode Tampilan. Kemudian terdapat pula 2 tombol yaitu *cancel* dan *ok*, tombol *cancel* berfungsi untuk menutup kotak dialog tanpa melakukan perubahan apapun,

sedangkan tombol ok adalah untuk menyimpan perubahan yang dilakukan oleh pengguna dan juga menutup kotak dialog.

3. Perancangan *Mockup* Halaman Tambah Kontak Awal

Gambar 5.20 Rancangan *Mockup* Halaman Tambah Kontak Awal

Gambar 5.20 merupakan rancangan *mockup* untuk halaman tambah kontak awal yang muncul pertama kali ketika tombol tambah kontak pada halaman utama ditekan. Pada halaman ini terdapat 1 *tittle bar* yang menjelaskan judul dari halaman ini yaitu "Tambah Kontak" dan 3 tombol yaitu tombol tambah kontak yang berfungsi untuk membuka halaman tambah kontak untuk menambah kontak berdasarkan input teks secara manual, tombol mode suara yang berfungsi untuk mengaktifkan mode suara pada sistem, tombol kembali ke menu utama yang

berfungsi yang berfungsi untuk mengembalikan tampilan pengguna ke halaman utama.

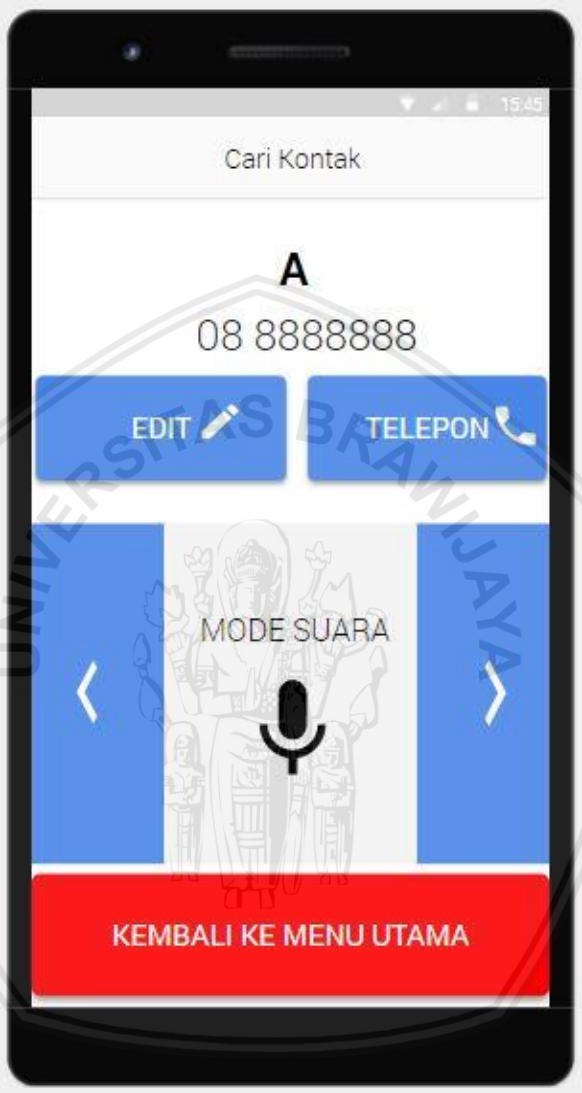
4. Perancangan *Mockup* Halaman Tambah Kontak Input Manual

Gambar 5.21 Rancangan *Mockup* Halaman Tambah Kontak Input Manual

Gambar 5.21 merupakan rancangan *mockup* untuk halaman tambah kontak dengan masukkan data teks secara manual oleh pengguna. Seperti yang dapat dilihat dalam Gambar 5.2 *tittle bar* atau judul dari halaman ini adalah "Tambah Kontak" dan terdapat dua buah *field* yang perlu diinputkan dan sebuah *label* untuk menjelaskan *field* yang perlu diinputkan yang terdiri dari *label* nama yang menjelaskan bahwa *field input* dibawahnya harus diisikan dengan nama, selain itu terdapat pula *placeholder* yang menerangkan "masukkan nama kontak", *label* No Telepon yang menjelaskan bahwa *field input* dibawahnya harus diisikan dengan nomor telepon, selain itu terdapat pula *placeholder* yang menerangkan "masukkan nomor kontak". Selain itu terdapat pula dua buah tombol yaitu tombol "Batal" yang berfungsi untuk mentup halaman dan kembali ke halaman

sebelumnya, dan tombol simpan yang berfungsi untuk menyimpan perubahan yang dilakukan oleh pengguna.

5. Perancangan *Mockup* Halaman Cari Kontak



Gambar 5.22 Rancangan *Mockup* Halaman Cari Kontak

Gambar 5.22 merupakan rancangan *mockup* dari halaman cari kontak, seperti yang dapat dilihat dalam Gambar 5.22 untuk *title bar* atau judul dari halaman ini yaitu "Cari Kontak", kemudian pada bagian atas dari halaman ini terdapat 2 *label* yaitu *label* yang paling atas yang menunjukkan nama kontak yang muncul pada halaman ini dan *label* dibawahnya yang menunjukkan nomor kontak yang muncul pada halaman ini. Dibawahnya lagi terdapat 2 tombol yaitu tombol edit yang berfungsi untuk membuka halaman edit kontak dan melakukan edit terhadap kontak yang ditemukan dan tombol telepon yang berfungsi untuk melakukan panggilan terhadap kontak yang muncul pada halaman ini.

Pada bagian tengah terdapat 3 tombol yaitu tombol dengan *icon chevron left* yang berfungsi untuk menggeser atau mengganti kontak ke urutan/abjad sebelumnya, tombol mode suara dengan *icon microphone* yang berfungsi untuk mengaktifkan mode suara, dan tombol dengan *icon chevron right* yang berfungsi untuk menggeser atau mengganti kontak ke urutan atau abjad setelahnya. Kemudian pada bagian paling bawah terdapat sebuah tombol yang berfungsi untuk mengembalikan tampilan ke halaman utama.

6. Rancangan *Mockup* Halaman Edit Kontak

Gambar 5.23 Rancangan *Mockup* Halaman Edit Kontak

Gambar 5.23 merupakan rancangan *mockup* untuk halaman edit kontak berdasarkan kontak yang dipilih atau ditemukan pada halaman cai kontak. Seperti yang dapat dilihat pada Gambar 5.23 *title bar* atau judul dari halaman ini "Edit Kontak". Pada bagian atas halaman ini terdapat 2 tulisan yang menunjukan nama dan kontak sebelumnya yang belum diedit, kemudian terdapat *form* yang berfungsi sebagai tempat mengedit nama dan nomor telepon dari kontak tersebut, masing-masing *field* sebelumnya sudah terisi dengan nilai nama/nomor

kontak yang tersimpan terakhir dan belum diedit. Pada bagian bawah terdapat 3 tombol yaitu tombol batal yang berfungsi untuk menutup halaman ini tanpa melakukan perubahan apapun, tombol simpan berfungsi unuk menyimpan segala perubahan yang ada pada halaman ini dan tombol hapus kontak yang berfungsi untuk menghapus kontak.

7. Rancangan *Mockup* Kotak Dialog Konfirmasi Hapus Kontak

Gambar 5.24 Rancangan *Mockup* Kotak Dialog Konfirmasi Hapus Kontak

Gambar 5.24 merupakan rancangan *mockup* untuk kotak dialog konfirmasi untuk melakukan hapus kontak yang muncul pada halaman edit kontak. Seperti yang dapat dilihat dalam Gambar 5.24 pada kotak dialog konfirmasi hapus kontak terdapat judul yang bertuliskan "Konfirmasi Hapus Kontak", isi yang menanyakan "Apakah anda yakin ingin mengapus kontak dengan nama: <nama kontak> dan nomor: <nomor kontak>" dan dua buah tombol yaitu tombol tidak yang berfungsi untuk menutup kotak dialog konfirmasi tanpa melakukan perubahan apapun dan tombol iya yang berfungsi untuk melakukan menghapus kontak.

8. Rancangan *Mockup* Halaman Telepon Berdasarkan Input Nomor

Gambar 5.25 Rancangan *Mockup* Halaman Halaman Telepon Berdasarkan Masukkan Nomor Telepon

Gambar 5.25 merupakan rancangan *mockup* dari halaman telepon berdasarkan masukkan nomor telepon secara manual oleh pengguna. Seperti yang dapat dilihat pada Gambar 5.29 *title bar* atau judul dari halaman ini adalah "Telepon Nomor", terdapat sebuah *field input* untuk memasukkan nomor telepon dengan keterangan berupa tulisan "No Telepon" yang terdapat di atasnya, kemudian terdapat pula 3 buah tombol yaitu tombol telepon yang berfungsi untuk melakukan panggilan terhadap nomor yang sudah dimasukkan tersebut, tombol mode suara yang berfungsi untuk mengaktifkan mode suara pada sistem dan tombol kembali ke menu utama yang berfungsi untuk mengembalikan tampilan halaman ke menu utama.

5.9 Evaluasi Desain

Setelah melakukan perancangan antarmuka dalam bentuk *mockup*, kemudian dilakukan evaluasi terhadap hasil dari *design* (perancangan) *mockup* terhadap satu narasumber calon pengguna dengan menunjukkan hasil *design* dan menanyakan perihal kemudahan penggunaan aplikasi bila dilihat dari sisi antarmukanya.

Untuk menghasilkan perancangan *mockup* seperti yang dilampirkan dan dijelaskan pada subbab Perancangan *Mockup* dilakukan proses yang melibatkan pengguna dimulai dari pembuatan rancangan awal, lalu ditunjukkan kepada calon pengguna, kemudian dilakukan perubahan lagi sehingga menghasilkan rancangan yang sesuai dengan kebutuhan calon pengguna. Beberapa perubahan yang dilakukan seperti perubahan ukuran tampilan, penambahan *icon* sebagai bentuk gambaran visual untuk menjelaskan fungsi yang akan dilakukan, dan perubahan tata letak misalnya untuk tampilan awal yang sebelumnya dirancang tata letak tulisan berada pada posisi atas dan bawah, berdasarkan hasil yang diamati pengguna tuna netra sedang cenderung tidak membaca tulisan dan lebih fokus memanfaatkan *icon* dalam penggunaan tombol-tombol yang ada pada aplikasi sehingga pada halaman utama posisi peletakan tulisan dan *icon* pada tombol dipindahkan ke tengah agar lebih mudah untuk dibacaoleh pengguna tuna netra sedang.

BAB 6 IMPLEMENTASI SISTEM

Bab ini berisikan tentang pengimplementasian kebutuhan fungsional yang telah dirancang pada bab sebelumnya. Implementasi dilakukan berdasarkan kepada hasil dari tahap perancangan yang diperoleh sebelumnya. Implementasi sistem dilakukan dengan menggunakan *Framework* Ionic dan bahasa pemrograman Typescript. Pada Bab ini akan dijelaskan mengenai spesifikasi sistem yang digunakan dalam pengembangan sistem yang sedang dibangun, batasan-batasan apa saja dalam mengimplementasi sistem, implementasi *class diagram*, implementasi kode program dan implementasi antarmuka dari sistem.

6.1. Spesifikasi Lingkungan Pengembangan Sistem

Pada pengembangan aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tunanetra dilakukan pada sebuah lingkungan (*environment*) yang terdiri menjadi dua bagian yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Spesifikasi perangkat keras terbagi menjadi dua yaitu spesifikasi perangkat keras mesin pengembang dapat dilihat pada Tabel 6.1 dan spesifikasi perangkat keras *smartphone* yang digunakan dapat dilihat pada Tabel 6.2.

Tabel 6.1 Spesifikasi Perangkat Keras Mesin Pengembang

Nama Komponen	Spesifikasi
Sistem Model	ASUS X455LDB
Processor	Intel CORE i5-52000
Memory	4GB
Display	NVIDIA GEFORCE 820M

Dapat dilihat pada Tabel 6.1 merupakan spesifikasi perangkat keras mesin pengembang yang menggunakan komputer lipat (*laptop)* dengan jenis Asus X455LDB yang memiliki *processor* dengan tipe Intel CORE i5-52000, RAM sebesar 4GB dan layar *display* dengan tipe NVIDIA GEFORCE 820M.

Tabel 6.2 Spesifikasi Perangkat Keras *Smartphone*

Nama Komponen	Spesifikasi
Sistem Model	Redmi 4X
Processor	Qualcomm Snapdragon 435 MSM8940
Memory	3 GB
Display	Adreno 505

Pada Tabel 6.2 menjelaskan mengenai spesifikasi perangkat keras perangkat bergerak atau *smartphone* yang digunakan selama pengembangan sistem yang merupakan *smartphone* dengan merek dagang Xiaomi dengan tipe Redmi 4X,

dengan *processor* Qualcomm Snapdragon 435 MSM8940, RAM sebesar 3GB dan *display* dengan jenis Adreno 505.

Untuk spesifikasi perangkat lunak juga terbagi menjadi dua yaitu spesifikasi perangkat lunak mesin pengembang dapat dilihat pada Tabel 6.3 dan spesifikasi perangkat lunak *smartphone* yang digunakan dapat dilihat pada Tabel 6.3.

Tabel 6.3 Spesifikasi Perangkat Lunak Mesin Pengembang

Nama Komponen	Spesifikasi
Operating System	Windows 10 Pro
Bahasa Pemrograman	TypeScript, HTML
Intergrated Development Environment (IDE)	Visual Studio Code

Dapat lihat pada Tabel 6.3 merupakan spesifikasi dari perangkat lunak yang digunakan dalam pengembangan sistem ini yaitu dengan Sistem Operasi jenis Windows 10 Pro dengan versi 64 bit, bahasa pemrograman yang digunakan adalah bahasa TypeScript dan perangkat lunak IDE yang digunakan dalam membantu proses pengembangan adalah Visual Studio Code.

Tabel 6.4 Spesifikasi Perangkat Lunak *Smartphone*

Nama Komponen	Spesifikasi
Operating System	Android 7.1.2 (Nougat)
SDK	27

Pada Tabel 6.4 menjelaskan mengenai spesifikasi perangkat lunak dari *smartphone* yang digunakan atau menjadi acuan dalam implementasi sistem yaitu memiliki jenis sistem operasi Android 7.1.2 atau yang lebih dikenal dengan tipe Nougat dan memiliki SDK (Software Development Kit) 27.

6.2. Batasan-Batasan Implementasi

Dalam proses pengimplementasian aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tunanetra memiliki batasan-batasan sebagai berikut.

1. Sistem perangkat lunak dibangun dengan bahasa pemrograman Typescript dan HTML.
2. Sistem perangkat lunak dibangun tanpa melibatkan *database* terpisah dan langsung terhubung dengan *database* pada *smartphone* pengguna.
3. Sistem perangkat lunak dibangun menggunakan layanan Google Text-to-Speech dan Google Voice Recognition secara *cloud service* sehingga tidak dapat dijalankan dalam keadaan *offline* atau tidak terhubung dengan jaringan internet.

4. Pengimplementasian sistem dilakukan dengan melakukan *build* .apk menggunakan *command prompt* dan kemudian .apk tersebut dipasangkan (*install*) pada *smartphone* Android milik pengguna.

6.3. Implementasi Perancangan *Class*

Setiap *class* yang dirancang akan diimplementasikan menggunakan bahasa pemrograman Typescript dan akan disimpan dalam sebuah Ike yang berekstensi .ts. Masing-masing *class* yang diimplementasikan berdasarkan hasil dari rancangan sistem yang dijelaskan melalui Tabel 6.5.

Tabel 6.5 Implementasi *Class*

Folder	Nama *Class*	Deskripsi
voice dita/src/pages /home	home.ts	Digunakan untuk mengurusi tampilan pada halaman utama baik pada tampilan *graphical user interface* maupun *voice user interface*.
voice dita/src/pages /tambah-kontak	tambah-kontak.ts	Digunakan sebagai tampilan tambah kontak awal yang memberikan pilihan pada pengguna untuk menggunakan mode tampilan ataupun mode suara untuk melakukan tambah kontak.
voice dita/src/pages /tambah-kontak-manual	tambah-kontak-manual.ts	Digunakan dalam mengurusi penambahan kontak baru melalui mode tampilan melalui masukkan *teks*.
voice dita/src/pages /cari-kontak	cari-kontak.ts	Digunakan untuk mengurusi proses pencarian kontak pada *smartphone* pengguna dan proses melakukan panggilan terhadap kontak yang telah ditemukan.
voice dita/src/pages /edit-kontak	edit-kontak.ts	Digunakan dalam proses melakukan edit atau perubahan data kontak berupa data nama dan atau nomor kontak, dan proses hapus kontak.
voice dita/src/pages /telepon	telepon.ts	Digunakan dalam proses melakukan panggilan melalui masukkan nomor telepon.

Dapat dilihat pada Tabel 6.5 merupakan tabel yang menjelaskan mengenai perancangan dari *class* yang digunakan dalam sistem pada Tabel 6.5 menjelaskan mengenai letak folder dari tiap *class*, nama *class* tersebut, dan deskripsi atau penjelasan mengenai fungsi dari *class* tersebut.

6.4. Implementasi Kode Program

Dalam melakukan proses implementasi kode program pertama-tama akan dibuat kode program dari setiap fungsionalitas yang ada menggunakan bahasa pemrograman Typescript. Berikut merupakan beberapa implementasi kode program dari fungsionalitas utama pada sistem.

1. Implementasi Kode Program Menambah Kontak

Dalam proses menambahkan kontak baru ke dalam daftar kontak pada *smartphone* dapat dilakukan dengan dua cara yaitu melalui mode suara atau *voice user interface* dan melalui mode tampilan biasa menggunakan *graphical user interface* di mana pada proses data kontak yang dapat ditambahkan adalah berupa nama dan nomor kontak saja. Untuk implementasi kode program menambahkan kontak dapat dilihat seperti pada Tabel 6.6.

Tabel 6.6 Implementasi Kode Program Menambah Kontak

home.ts

```
1  namaKontak(){
2      this.tts.speak({
3        text: 'sebutkan nama kontak yang akan di simpan',
4        locale: 'id-ID',
5        rate: 1
6      })
7      .then(() => {
8        let options={
9          language: 'id-ID'
10       }
11       this.speechRecognition.startListening(options)
12       .subscribe(
13         (matches: Array<string>) => {
14           console.log("nama",matches)
15           this.cd.detectChanges();
16           this.checkNama(matches[0])
17         },
18         (onerror) => console.log('error:', onerror)
19       )
20     })
```

```
21    }
22    checkNama(nama){
23      this.contacts.find(
24        ["displayName", "phoneNumbers"],
25        {multiple: true, hasPhoneNumber: true, filter:nama}
26      ).then(res=>{
27        if (res.length==0) {
28          this.confirmNama(nama)
29        }else{
30          this.tts.speak({
31            text: nama+' telah ada di kontak. apakah anda tetap akan melanjutkan? katakan iya jika inging melanjutkan. katakan tidak jika ingin kembali.',
32            locale: 'id-ID',
33            rate: 1
34          })
35          .then(() => {
36            let options={
37              language: 'id-ID'
38            }
39            this.speechRecognition.startListening(options)
40            .subscribe(
41              (matches: Array<string>) => {
42                console.log("iya/tidak",matches)
43                this.cd.detectChanges();
44               if(this.matches[0].toLocaleLowerCase()==="iya"){
45                  this.confirmNama(nama)
                  }else{
46                  this.namaKontak()
47                }
48              },
49              (onerror) => console.log('error:', onerror)
50            )
51          })
52        }
53      })
54    }
55    confirmNama(nama){
56      this.tts.speak({
```

```
      text: 'Apakah Benar nama yang akan disimpan adalah.'+nama+"?     katakan iya bila benar. katakan tidak bila salah.",
      locale: 'id-ID',
      rate: 0.6
    })
    .then(() => {
      let options={
        language: 'id-ID'
      }
      this.speechRecognition.startListening(options)
      .subscribe(
        (matches: Array<string>) => {
          console.log("konfirm nama",matches)
          this.cd.detectChanges();
          if (matches[0].toLowerCase()=="iya") {
            this.nama=nama
            this.nomorKontak()
          } else if (matches[0].toLowerCase()=="tidak"){
            this.namaKontak()
          } else{
            this.tts.speak({
              text: "Maaf Suara Anda Kurang Jelas.",
              locale: 'id-ID',
              rate: 1
            }).then(()=>{
              this.confirmNama(nama)
            })
          }
        },
        (onerror) => console.log('error:', onerror)
      )
    })
  }
  nomorKontak(){
    this.tts.speak({
      text: 'sebutkan nomor kontak yang akan di simpan',
      locale: 'id-ID',
      rate: 1
```

```
93      })
94      .then(() => {
95        let options={
96          language: 'id-ID'
97        }
98        this.speechRecognition.startListening(options)
99        .subscribe(
100         (matches: Array<string>) => {
101           console.log("nomor",matches)
102           this.cd.detectChanges();
103           this.confirmNomor(matches[0])
104         },
105         (onerror) => console.log('error:', onerror)
106       )
107     })
108   }
109   confirmNomor(nomor){
110     this.tts.speak({
111       text: 'Apakah Benar Nomor yang akan disimpan adalah.'+nomor+"?    katakan iya bila benar. katakan tidak bila salah.",
112       locale: 'id-ID',
113       rate: 0.6
114     })
115     .then(() => {
116       let options={
117         language: 'id-ID'
118       }
119       this.speechRecognition.startListening(options)
120       .subscribe(
121         (matches: Array<string>) => {
122           console.log("konfirm nama",matches)
123           this.cd.detectChanges();
124           if (matches[0].toLowerCase()=="iya") {
125             this.phone=nomor
126             this.simpanKontak()
127           } else if (matches[0].toLowerCase()=="tidak"){
128             this.nomorKontak()
129           } else{
```

```
130             this.tts.speak({
131               text: "Maaf Suara Anda Kurang Jelas.",
132               locale: 'id-ID',
133               rate: 1
134             }).then(()=>{
135               this.confirmNomor(nomor)
136             })
137           }
138         },
139         (onerror) => console.log('error:', onerror)
140       )
141     })
142   }
143   simpanKontak(){
144     let contact: Contact = this.contacts.create();
145
146     contact.name = new ContactName(null, this.nama, '');
147     contact.phoneNumbers = [new ContactField('mobile',
this.phone)];
148     contact.save().then(
149       () => {
150         this.tts.speak({
151           text: "Berhasil Menyimpan kontak",
152           locale: 'id-ID',
153           rate: 1
154         })
155         .then(() => {
156           this.ionViewWillEnter()
157         })
158         console.log('Contact saved!', contact)
159       },
160       (error: any) => {
161         this.tts.speak({
162           text: "Maaf terjadi Kesalahan.",
163           locale: 'id-ID',
164           rate: 1
165         })
166         .then(() => {
167           this.ionViewWillEnter()
```

```
168           })
169           console.error('Error saving contact.', error)
170         }
171       );
172
```

Dapat dilihat Pada Tabel 6.6 merupakan kode program untuk melakukan penambahan kontak pada *smatphone* pengguna, penjelasan kode program tersebut dapat dilihat seperti dibawah ini.

1. Baris 1-54 merupakan *method* yang berfungsi untuk memberikan *feedback* berupa suara untuk meminta pengguna menyebutkan nama kontak yang akan ditambahkan
2. Baris 55-87 merupakan *method* yang berfungsi untuk memberikan *feedback* berupa konfirmasi ulang nama kontak yang sebelumnya telah disebutkan oleh pengguna.
3. Baris 88-108 merupakan *method* yang berfungsi untuk memberikan untuk memberikan *feedback* kepada pengguna untuk melakukan melakukan penambahan nama kontak.
4. Baris 110-143 merupakan *method* yang berfungsi untuk memberikan *feedback* berupa konfirmasi nomor kontak yang akan ditambahkan.
5. Baris 144-172 merupakan *method* yang berfungsi untuk menyimpan data nama dan nomor yang telah dimasukkan dan mendaftarkannya menjadi sebuah kontak baru pada *smartphone* pengguna.

2. Implementasi Kode Program Mencari Kontak

Dalam proses melakukan pencarian kontak yang sebelumnya sudah ada dan terdaftar pada *smartphone* pengguna dapat dilakukan melalui dua cara yaitu melalui mode suara atau voice user interface dan melalui mode tampilan biasa menggunakan graphical user interface di mana pada proses mencari kontak melalui suara data kontak yang diperlukan adalah data nama kontak, untuk proses pencarian kontak dapat dilihat dalam Tabel 6.7.

Tabel 6.7 Implementasi Kode Program Mencari Kontak

home.ts

```
1  cari(){
2      this.tts.speak({
3        text: 'sebutkan nama kontak',
4        locale: 'id-ID',
5        rate: 1
6      })
7      .then(() => {
8        let options={
9          language: 'id-ID'
```

```
10        }
11        this.speechRecognition.startListening(options)
12        .subscribe(
13          (matches: Array<string>) => {
14            console.log(matches)
15            this.cd.detectChanges();
16            this.hasilCari(matches)
17          },
18          (onerror) => console.log('error:', onerror)
19        )
20      })
21    }
22  hasilCari(hasil){
23      this.contacts.find(
24        ["displayName", "phoneNumbers"],
25        {multiple: true, hasPhoneNumber: true, filter:hasil[0]}
      ).then(res=>{
26        if (res.length==0) {
27          this.tts.speak({
28            text: 'hasil pencarian adalah tidak ada. silahkan
29  ulangi kembali pencarian.',
            locale: 'id-ID',
30          rate: 1
31        })
          .then(() => {
32          this.cari()
33        })
34      } else {
35        this.hasil=res;
36        console.log("hasil",this.hasil)
37        let string=""
38        res.forEach((element,index) => {
39          index=index+1
40          string=string+index+",                                    "+
41  element.displayName+".          ";
        });
42        this.tts.speak({
43          text: 'hasil pencarian adalah '+string+'. sebutkan
    nomor urutan untuk memilih.',
```

```
44              locale: 'id-ID',
                rate: 1
45            })
46      })
47    }
```

Tabel 6.7 merupakan kode program untuk melakukan pencarian kontak pada *smatphone* pengguna, penjelasan kode program tersebut dapat dilihat seperti dibawah ini.

1. Baris 1-21 merupakan *method* yang berfungsi untuk memberikan *feedback* kepada pengguna untuk memasukkan nama kontak yang akan dicari
2. Baris 21-47 merupakan *method* yang berfungsi untuk menampilkan daftar hasil pencarian kontak yang telah ditemukan dan mengembalikan *feedback* kepada pengguna dalam bentuk suara dan.

3. Implementasi Kode Program Mengedit Kontak

Dalam proses melakukan perubahan terhadap data kontak yang sebelunya sudah terdaftar pada *smartphone* pengguna dapat dilakukan melalui dua cara yaitu melalui mode suara atau voice user interface dan melalui mode tampilan biasa menggunakan graphical user interface di mana pada proses mengedit kontak harus terlebih dahulu dilakukan pencarian kontak dan ditemukan satu kontak yang nantinya akan diubah datanya baik data nama, nomor, atau keduanya proses perubahan data kontak dapat dilihat dalam Tabel 6.8.

Tabel 6.8 Implementasi Kode Program Mengedit Kontak

home.ts

```
1   editNomer(){
2       this.tts.speak({
3         text: 'sebutkan nomor kontak yang akan di simpan',
4         locale: 'id-ID',
5         rate: 1
6       })
7       .then(() => {
8         let options={
9           language: 'id-ID'
10        }
11        this.speechRecognition.startListening(options)
12        .subscribe(
13          (matches: Array<string>) => {
14            console.log("nama",matches)
15            this.cd.detectChanges();
16            this.confirmEditNomor(matches[0])
```

```
17          },
18          (onerror) => console.log('error:', onerror)
19        )
20      })
21    }
22    editNama(){
23      this.tts.speak({
24        text: 'sebutkan nama kontak yang akan di simpan',
          locale: 'id-ID',
25        rate: 1
26      })
27      .then(() => {
28        let options={
29          language: 'id-ID'
30        }
31        this.speechRecognition.startListening(options)
32        .subscribe(
33          (matches: Array<string>) => {
34            console.log("nama",matches)
35            this.cd.detectChanges();
36            this.checkEditNama(matches[0])
37          },
38          (onerror) => console.log('error:', onerror)
39        )
40      })
41    }
42  editNamaNomer(){
43      this.tts.speak({
44        text: 'sebutkan nama kontak yang akan di simpan',
          locale: 'id-ID',
45        rate: 1
46      })
47      .then(() => {
48        let options={
49          language: 'id-ID'
50        }
51        this.speechRecognition.startListening(options)
52        .subscribe(
```

```
53          (matches: Array<string>) => {
54            console.log("nama",matches)
55            this.cd.detectChanges();
56            this.checkNamaNomer(matches[0])
57          },
58          (onerror) => console.log('error:', onerror)
59        )
60      })
61    }
62    simpanKontakEdit(){
63      let contact: Contact = this.contacts.create();
64      contact.name = new ContactName(null, this.namaEdit, '');
65      contact.phoneNumbers = [new ContactField('mobile',
this.nomerEdit)];
66      contact.save().then(
67        () => {
68          this.tts.speak({
69            text: "Berhasil Menyimpan kontak",
70            locale: 'id-ID',
71            rate: 1
72          })
73          .then(() => {
74            this.ionViewWillEnter()
75          })
76          console.log('Contact saved!', contact)
77        },
78        (error: any) => {
79          this.tts.speak({
80            text: "Maaf terjadi Kesalahan.",
81            locale: 'id-ID',
82            rate: 1
83          })
84          .then(() => {
85            this.ionViewWillEnter()
86          })
87          console.error('Error saving contact.', error)
88        }
89      );
90    }
```

Tabel 6.8 merupakan kode program untuk melakukan perubahan data kontak pada *smatphone* pengguna, penjelasan kode program tersebut dapat dilihat seperti dibawah ini.

1. Baris 1-21 merupakan *method* yang digunakan untuk melakukan perubahan data nama saja pada kontak
2. Baris 22-41 merupakan *method* yang digunakan dalam melakukan perubahan data nomor saja.
3. Baris 41-61 merupakan proses untuk melakukan pengeditan kedua data kontak yaitu nama dan nomor kontak.
4. Baris 62-90 merupakan proses untuk melakukan penyimpanan perubahan data yang telah dilakukan oleh pengguna.

4. Implementasi Kode Program Menghapus Kontak

Dalam proses melakukan penghapusan terhadap salah satu data kontak yang sebelumnya sudah ada dan terdaftar dalam *smartphone* pengguna dapat dilakukan melalui dua cara yaitu melalui mode suara atau voice user interface dan melalui mode tampilan biasa menggunakan graphical user interface, di mana pada proses menghapus kontak dapat dilakukan setelah proses pencarian kontak dan ditemukan satu kontak yang akan dihapus datanya dari daftar kontak pada *smartphone* pengguna untuk implementasi kode program menghapus kontak dapat dilihat dalam Tabel 6.9.

Tabel 6.9 Implementasi Kode Program Menghapus Kontak

home.ts	
1	`hapus(element){`
2	`this.tts.speak({`
3	`text: 'Apakah Anda yakin untuk menghapus? katakan iya jika yakin. katakan tidak jika tidak yakin.',`
4	`locale: 'id-ID',`
5	`rate: 1`
6	`})`
7	`.then(() => {`
8	`let options={`
9	`language: 'id-ID'`
10	`}`
11	`this.speechRecognition.startListening(options)`
12	`.subscribe(`
13	`(matches: Array<string>) => {`
14	`console.log(matches)`
15	`this.cd.detectChanges();`
16	`if(matches.indexOf("iya")!=-1){`

```
17              let del : Contact = this.hasil[parseInt(element)-
1]
18
19              del.remove().then(res=>{
20                console.log("del res",res)
21                this.tts.speak({
22                        text: "Berhasil Menghapus kontak",
23                        locale: 'id-ID',
24                        rate: 1
25                      })
26                      .then(() => {
27                        this.ionViewWillEnter()
28                      })
29              }).catch(err=>{
30                console.log("del err",err)
31              })
32            }
33          },
34          (onerror) => console.log('error:', onerror)
35        )
36      })
    }
```

Tabel 6.9 merupakan kode program untuk melakukan perubahan data kontak pada *smatphone* pengguna, penjelasan kode program tersebut dapat dilihat seperti dibawah ini.

1. Baris 1-37 merupakan serangkaian proses yang ada dalam sebuah *method* yang berfungsi untuk melakukan penghapusan data kontak pada *smartphone* pengguna menggunakan *method* remove(), namun sebelumnya pengguna akan diminta konfirmasi apakah benar bahwa kontak tersebut akan dihapus jika pengguna memberikan konfirmasi “ya” barulah sistem akan menghapus kontak telepon tersebut pada *storage local smartphone* pengguna.

5. Implementasi Kode Program Melakukan Panggilan Kontak

Dalam proses melakukan panggilan terhadap satu kontak yang telah terdaftar dalam kontak telepon pada *smartphone* pengguna yang dapat dilakukan melalui dua cara yaitu melalui mode suara atau voice user interface dan melalui mode tampilan biasa menggunakan graphical user interface, di mana pada proses melakukan panggilan terhadap kontak dapat dilakukan setelah proses pencarian kontak telah dilakukan dan telah terpilih satu kontak yang ingin dilakukan

pemanggilan kontak, untuk implementasi kode program melakukan panggilan kontak dapat dilihat dalam Tabel 6.10.

Tabel 6.10 Implementasi Kode Program Melakukan Panggilan Kontak

home.ts

```
hasilCari(hasil){
this.tts.speak({
      text: 'sebutkan nama kontak',
      locale: 'id-ID',
      rate: 1
        .then(() => {
          let options={
            language: 'id-ID'
          }
          this.speechRecognition.startListening(options)
          .subscribe(
            (matches: Array<string>) => {
             console.log(matches)
             this.cd.detectChanges();
             let keepGoing=true
              matches.forEach(element => {
                if(keepGoing) {
                  if(isNaN(parseInt(element))){
                    keepGoing = true;
                  }   }else{
                    keepGoing=false;
                   console.log("pasreInt",parseInt(element))
                    this.listenStop();
this.callNumber.callNumber(this.hasil[parseInt(element)-
1].phoneNumbers[0].value,true)
                    this.platform.exitApp()
                  }
                }
              });
            },
            (onerror) => console.log('error:', onerror)
          )
        })
      }
```

Tabel 6.10 merupakan kode program untuk melakukan panggilan kontak pada *smatphone* pengguna, penjelasan kode program tersebut dapat dilihat seperti dibawah ini.

1. Baris 1-31 merupakan proses untuk melakukan panggilan kontak yang sebelumnya sudah terdaftar pada *smartphone* pengguna, sebelumnya pengguna akan diminta untuk memasukkan nama kontak yang akan dipanggil dan sistem akan mengecek berdasarkan *element* dari nama kontak yang dicari, lalu setelah data ditemukan maka sistem akan membuka layar panggilan telepon dan memanggil berdasarkan nomor telepon pada kontak tersebut,kemudian sistem akan menutup aplikasi.

6. Implementasi Kode Program Melakukan Panggilan Berdasarkan Masukkan Nomor Telepon

Dalam proses melakukan panggilan terhadap nomor telepon dapat dilakukan melalui dua cara yaitu melalui mode suara atau voice user interface dan melalui mode tampilan biasa menggunakan graphical user interface, di mana pada proses ini pengguna perlu memasukkan nomor telepon yang akan dipanggil dan untuk mode suara pengguna perlu menyebutkan nomor telepon yang ingin dipanggil, untuk implementasi kode program melakukan panggilan berdasarkan masukkan nomor telepon dapat dilihat dalam Tabel 6.11.

Tabel 6.11 Implementasi Kode Program Melakukan Panggilan Berdasarkan Masukkan Nomor Telepon

home.ts

```
1   telepon(){
2       this.tts.speak({
          text: 'sebutkan nomor telepon yang akan dipanggil',
3         locale: 'id-ID',
4         rate: 1
5       })
6       .then(() => {
7         let options={
8           language: 'id-ID'
9         }
10        this.speechRecognition.startListening(options)
11        .subscribe(
12          (matches: Array<string>) => {
13            console.log("nomor",matches)
14            this.cd.detectChanges();
15            this.confirmTelepon(matches[0])
16          },
17          (onerror) => console.log('error:', onerror)
```

```
18        )
19      })
20    }

21    confirmTelepon(tlp){
22      this.tts.speak({
23        text: 'Apakah Benar Nomor yang akan telepon
24  adalah.'+tlp+"?   katakan iya bila benar. katakan tidak bila
    salah.",
25        locale: 'id-ID',
26        rate: 0.6
27      })
28      .then(() => {
29        let options={
30          language: 'id-ID'
31        }
32        this.speechRecognition.startListening(options)
33        .subscribe(
34          (matches: Array<string>) => {
35            console.log("konfirm nama",matches)
36            this.cd.detectChanges();
33            if (matches[0].toLowerCase()=="ya") {
34              this.tlp=tlp
35              this.panggilan()
36            } else if (matches[0].toLowerCase()=="tidak"){
37              this.telepon()
38            } else{
39              this.tts.speak({
40                text: "Maaf Suara Anda Kurang Jelas.",
41                locale: 'id-ID',
42                rate: 1
43              }).then(()=>{
44                this.confirmTelepon(tlp)
45              })
46            }
47          },
48          (onerror) => console.log('error:', onerror)
49        )
50      })
```

```
51    }


52  panggilan(){
53     this.listenStop();
54     this.callNumber.callNumber(this.tlp,true)
55     this.platform.exitApp()
56    }
```

Tabel 6.11 merupakan kode program untuk melakukan panggilan terhadap nomor telepon yang dimasukkan oleh pengguna, untuk kode programnya dapat dilihat pada Kode 6.6

1. Baris 1-20 merupakan *method* yang digunakan dalam memberikan respons ke pengguna dan meminta pengguna untuk memasukkan data nomor telepon yang akan dipanggil, serta menerima respons nomor telepon dari pengguna dan mengirimkan data tersebut ke dalam *method* confirmTelepon.
2. Baris 21-51 merupakan *method* yang digunakan untuk memberikan umpan balik ke pengguna dalam bentuk suara dan konfirmasi data nomor telepon yang sebelumnya telah diberikan oleh pengguna. Apabila pengguna memberikan respons berupa "tidak" maka sistem akan mengulang meminta masukkan data nomor telepon pada pengguna dan memanggil *method* telepon(), sedangkan apabila pengguna memberikan respons berupa "ya" maka akan dijalankan *method* panggilan()
3. Baris 52-57 merupakan *method* yang berfungsi untuk membuka halaman telepon pada *smartphone* pengguna dan melakukan pemanggilan nomor telepon berdasarkan data yang sebelumnya telah diberikan oleh pengguna.

6.5. Implementasi Antarmuka

Pada bagian ini akan menguraikan hasil dari implementasi antarmuka pada aplikasi *voice call*. Hasil dari implementasi antarmuka ini diperoleh berdasarkan proses implementasi mengggunakan bahasa pemrograman HTML yang mengacu pada perancangan *screenflow* yang telah diuraikan sebelumnya seperti yang ditunjukkan pada Gambar 5.17.

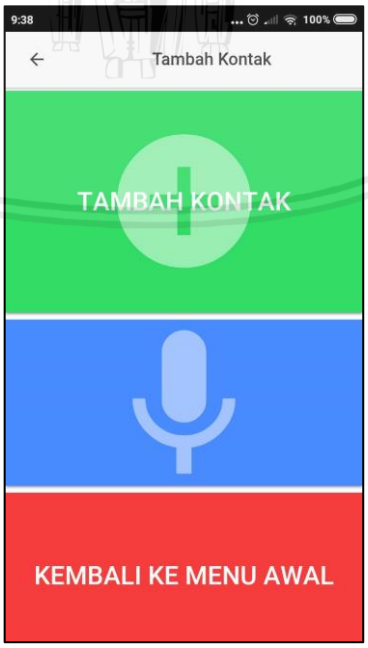
Hasil implementasi dari implementasi antarmuka yang diperoleh terdiri dari tampilan antarmuka halaman utama, halaman tambah kontak awal, halaman tambah kontak, halaman cari kontak, halaman edit kontak, halaman panggilan berdasarkan masukkan nomor telepon, kotak dialog pengaturan, dan kotak dialog konfirmasi hapus kontak.

1. **Implementasi Antarmuka Halaman Utama**

Gambar 6.1 Implementasi Antarmuka Halaman Utama

Dalam Gambar 6.1 merupakan hasil dari implementasi antarmuka halaman utama yang muncul ketika aplikasi pertama kali dibuka pada halaman ini dapat mengarahkan ke halaman tambah kontak awal, halaman cari kontak, dan halaman telepon

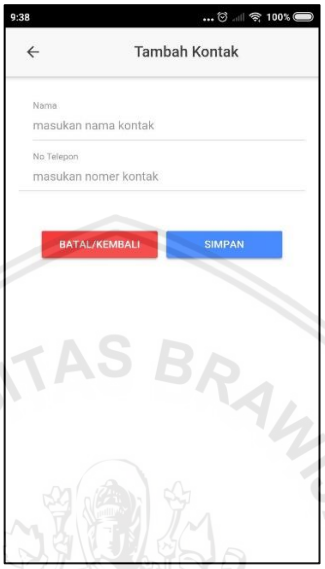
2. **Implementasi Halaman Tambah Kontak Awal**



Gambar 6.2 Implemantasi Antarmuka Halaman Tambah Kontak Awal

Dalam Gambar 6.2 merupakan hasil dari implementasi antarmuka halaman tambah kontak awal di mana pada halaman ini terdapat 3 tombol, tombol tambah kontak berfungsi untuk membuka halaman tambah kontak, tombol dengan *icon microphone* berfungsi untuk membuka mode suara, dan tombol kembali ke menu awal berfungsi untuk mengembalikan tampilan ke menu awal aplikasi.

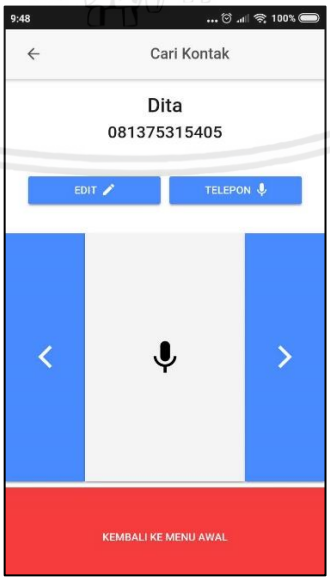
3. Implementasi Halaman Tambah Kontak



Gambar 6.3 Implementasi Antarmuka Halaman Tambah Kontak

Dalam Gambar 6.3 merupakan hasil dari implementasi antarmuka halaman tambah kontak di mana pada halaman ini terdapat *form input* untuk memasukkan data nama dan nomor telepon.

4. Implementasi Halaman Cari Kontak



Gambar 6.4 Impementasi Antarmuka Halaman Cari Kontak

Dalam Gambar 6.4 merupakan hasil dari implementasi antarmuka halaman cari kontak di mana pada halaman ini terdapat tombol untuk mengarahkan ke halaman edit kontak dan melakukan panggilan terhadap kontak yang ditemukan tersebut, selain itu juga dapat melakukan pencarian secara manual dengan menekan tombol dengan *icon previous* dan *next*.

5. Implementasi Halaman Edit Kontak

Gambar 6.5 Implementasi Halaman Edit Kontak

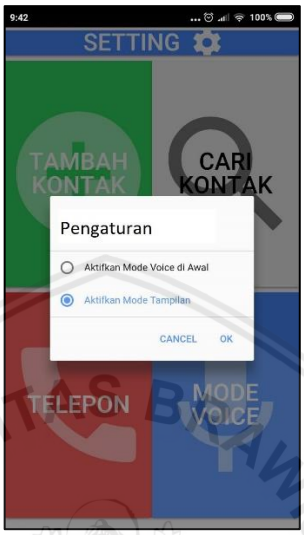
Dalam Gambar 6.5 merupakan hasil dari implementasi antarmuka halaman edit kontak di mana pada halaman ini akan menampilkan nama kontak dan nomor telepon dari kontak yang sudah dipilih sebelumnya yang dapat dilakukan perubahan.

6. Implementasi Halaman Panggilan Berdasarkan masukkan nomor telepon

Gambar 6.6 Implementasi Halaman Panggilan Nomer

Dalam Gambar 6.6 merupakan hasil dari implementasi antarmuka halaman panggilan nomor di mana pada halaman ini pengguna dapat melakukan panggilan berdasarkan masukkan nomor telepon, baik secara manual maupun melalui masukkan suara oleh pengguna.

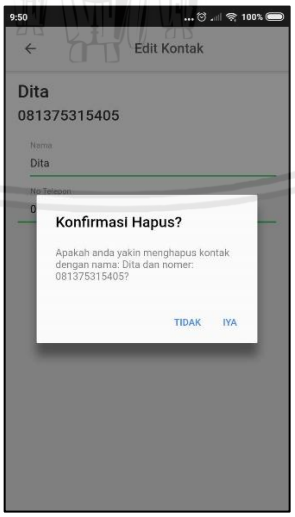
7. Implementasi Kotak Dialog Pengaturan



Gambar 6.7 Implementasi Kotak Dialog Pengaturan

Dalam Gambar 6.7 merupakan hasil dari implementasi antarmuka kotak dialog untuk melakukan pengaturan tampilan awal pada aplikasi ini. Pengaturan yang dapat dilakukan adalah untuk tampilan awal aplikasi berupa mode suara di awal ataupun langsung menampilkan mode tampilan.

8. Implementasi Kotak Dialog Konfirmasi Hapus Kontak



Gambar 6.8 Implementasi Kotak Dialog Hapus Kontak

Dalam Gambar 6.8 merupakan hasil dari implementasi antarmuka kotak dialog untuk melakukan hapus kontak, pada kotak dialog ini berisikan tulisan konfirmasi hapus kontak dan tombol tidak dan iya.

BAB 7 PENGUJIAN SISTEM

Bab ini menjelaskan mengenai proses pengujian yang dilakukan terhadap aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tunanetra pada perangkat bergerak dengan sistem operasi Android. Proses pengujian dilakukan dengan tujuan mengetahui seberapa baik kualitas dari aplikasi yang dibangun. Dalam prosesnya pengujian ini melibatkan pengujian validasi dengan menggunakan metode Black-box *testing*, pengujian *usability* dengan menggunakan SUS, dan pengujian dari *Quality of Experience* pengguna terhadap kualitas *voice interaction* yang disediakan oleh aplikasi *voice call* menggunakan MOS testing.

7.1. Pengujian Sistem

7.1.1. Pengujian Validasi

Pengujian validasi dilakukan berdasarkan kebutuhan kebutuhan fungsional dari aplikasi *voice call*. Dalam pengujian validasi dilakukan dengan menggunakan metode *blackbox testing* (pengujian kotak hitam), dalam proses pengujian menggunakan metode ini pertama-tama dilakukan perancangan studi kasus uji yang mengacu pada analisis hasil kebutuhan sistem yang telah dijabarkan sebelumnya. Daftar kasus uji yang digunakan dalam proses pengujian ini setiap fungsionalitasnya diuji menggunakan dua mode yaitu mode suara atau *Voice User Interface (VUI)* dan mode tampilan atau *Graphical User Interface* (GUI).

Tabel 7.1 Kasus Uji Melakukan Menambah Kontak VUI

Kode Pengujian	VC-3-1
Nama Kasus Pengujian	Melakukan Proses Menambahkan Kontak melalui VUI
Objek Uji	VC-1-1
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menambah kontak
Prosedur Pengujian	1. Membuka Aplikasi 2. Memberikan perintah suara “Oke” 3. Memberikan perintah suara “2” 4. Memberikan perintah suara berupa nama kontak 5. Memberikan perintah suara “Ya” 6. Memberikan perintah suara berupa nomor kontak

	7. Memberikan perintah suara 'Ya"
Hasil yang Diharapkan	Sistem dapat menyimpan data nama dan nomor kontak yang telah di masukkan oleh pengguna melalui VUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak".

Pada Tabel 7.1 menunjukkan kasus uji yang digunakan dalam melakukan proses menambah kontak melalui VUI yang mana objek yang diuji adalah VC-1-1 yaitu fungsionalitas untuk menambah kontak. Pada kasus uji ini tedapat 7 prosedur pengujian yang dilakukan.

Tabel 7.2 Kasus Uji Melakukan Tambah Kontak Melalui GUI

Kode Pengujian	VC-3-2
Nama Kasus Pengujian	Melakukan Proses Menambahkan Kontak melalui GUI
Objek Uji	VC-1-1
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menambah kontak
Prosedur Pengujian	1. Membuka Aplikasi 2. Menekan Tombol Tambah Kontak pada halaman tambah kontak awal 3. Menekan Tombol dengan tulisan Tambah Kontak 4. Memasukkan data nama dan nomor kotak pada kolom yang tersedia 5. Menekan Tombol Simpan
Hasil yang Diharapkan	Sistem dapat menyimpan data nama dan nomor kontak yang telah dimasukkan oleh pengguna melalui GUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"

Pada Tabel 7.2 menunjukkan kasus uji yang digunakan dalam melakukan proses menambah kontak melalui GUI yang mana objek yang diuji adalah VC-1-2 yaitu fungsionalitas untuk menambah kontak. Pada kasus uji ini tedapat 5 prosedur pengujian yang dilakukan.

Tabel 7.3 Kasus Uji Melakukan Pencarian Kontak Melalui VUI

Kode Pengujian	VC-3-3
Nama Kasus Pengujian	Melakukan Proses Pencarian Kontak melalui VUI
Objek Uji	VC-1-2
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas mencari kontak
Prosedur Pengujian	1. Membuka Aplikasi 2. Memberikan Perintah Suara "Oke" 3. Menyebutkan Perintah Suara "1" 4. Menyebutkan Perintah Suara berupa nama kontak 5. Memberikan perintah suara berupa nomor urutan kontak yang ditemukan
Hasil yang Diharapkan	Sistem dapat melakukan pencarian kontak berdasarkan masukkan data nama kontak oleh pengguna melalui VUI dan memberikan umpan balik suara berupa urutan nama kontak yang ditemukan apabila pengguna memasukkan data nama yang sudah terdaftar pada *smartphone* pengguna, dan akan mengembalikan umpan balik suara "Hasil Pencarian Tidak ada" dan mengulang meminta pengguna memasukkan kembali data nama apabila pengguna memasukkan data nama yang belum terdaftar pada *smartphone* pengguna

Pada Tabel 7.3 menunjukkan kasus uji yang digunakan dalam melakukan proses mencari kontak melalui VUI yang mana objek yang diuji adalah VC-1-2 yaitu fungsionalitas untuk mencari kontak. Pada kasus uji ini tedapat 5 prosedur pengujian yang dilakukan.

Tabel 7.4 Kasus Uji Melakukan Pencarian Kontak Melalui GUI

Kode Pengujian	VC-3-4
Nama Kasus Pengujian	Melakukan Proses Pencarian Kontak melalui GUI
Objek Uji	VC-1-2
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas mencari kontak
Prosedur Pengujian	1. Membuka Aplikasi 2. Menekan Tombol Cari Kontak 3. Menekan Tombol dengan *icon Next*
Hasil yang Diharapkan	Sistem dapat menampilkan data kontak yang ada pada *smartphone* pengguna melalui GUI dan data yang muncul pertama kali merupakan data nama kontak dengan huruf abjad paling awal

Pada Tabel 7.4 menunjukkan kasus uji yang digunakan dalam melakukan proses mencari kontak melalui GUI yang mana objek yang diuji adalah VC-1-2 yaitu fungsionalitas untuk mencari kontak. Pada kasus uji ini tedapat 3 prosedur pengujian yang dilakukan.

Tabel 7.5 Kasus Uji Melakukan Pengeditan Kontak Melalui VUI

Kode Pengujian	VC-3-5
Nama Kasus Pengujian	Melakukan Proses Pegeditan Kontak Melalui VUI
Objek Uji	VC-1-3
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas mengedit kontak
Prosedur Pengujian	1. Membuka Aplikasi 2. Memberikan Perintah Suara “Oke” 3. Memberikan perintah suara pilihan edit kontak *4.* Memberikan perintah suara untuk proses edit kontak.

Hasil yang Diharapkan	Sistem dapat melakukan perubahan data kontak berdasarkan perintah suara yang diberikan oleh pengguna melalui VUI

Pada Tabel 7.5 menunjukkan kasus uji yang digunakan dalam melakukan proses perubahan data kontak melalui VUI yang mana objek yang diuji adalah VC-1-3 yaitu fungsionalitas untuk mengedit kontak. Pada kasus uji ini tedapat 4 prosedur pengujian yang dilakukan.

Tabel 7.6 Kasus Uji Melakukan Proses Pengeditan Kontak Melalui GUI

Kode Pengujian	VC-3-6
Nama Kasus Pengujian	Melakukan Proses Pegeditan Kontak Melalui GUI
Objek Uji	VC-1-3
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas mengedit kontak
Prosedur Pengujian	1. Menekan tombol edit pada halaman cari kontak 2. Mengisi data edit kontak baik nama dan atau nomor kontak 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat mengubah data kontak yang pada *smartphone* pengguna berdasarkan data yang dimasukkan oleh pengguna melalui GUI

Pada Tabel 7.6 menunjukkan kasus uji yang digunakan dalam melakukan proses perubahan data kontak melalui GUI yang mana objek yang diuji adalah VC-1-3 yaitu fungsionalitas untuk mengedit kontak. Pada kasus uji ini tedapat 3 prosedur pengujian yang dilakukan.

Tabel 7.7 Kasus Uji Melakukan Pneghapusan Satu Data Kontak Melalui VUI

Kode Pengujian	VC-3-7
Nama Kasus Pengujian	Melakukan proses menghapus data kontak yang sebelumnya sudah terdaftar pada *smartphone* pengguna melalui VUI
Objek Uji	VC-1-4

Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menghapus kontak
Prosedur Pengujian	1. Membuka aplikasi 2. Melakukan proses edit kontak 3. Memberikan perintah suara "4" untuk menghapus kontak
Hasil yang Diharapkan	Sistem dapat menghapus data kontak yang dipilih oleh pengguna berdasarkan perintah suara melalui VUI

Pada Tabel 7.7 menunjukkan kasus uji yang digunakan dalam melakukan proses menghapus satu data kontak melalui VUI yang mana objek yang diuji adalah VC-1-4 yaitu fungsionalitas untuk menghapus kontak. Pada kasus uji ini tedapat 3 prosedur pengujian yang dilakukan.

Tabel 7.8 Kasus Uji Melakukan Penghapusan Satu Data Kontak Melalui GUI

Kode Pengujian	VC-3-8
Nama Kasus Pengujian	Melakukan proses menghapus data kontak yang sebelumnya sudah terdaftar pada *smartphone* pengguna melalui GUI
Objek Uji	VC-1-4
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menghapus kontak
Prosedur Pengujian	1. Membuka aplikasi 2. Menekan tombol hapus pada halaman edit kontak 3. Menekan perintah "Iya" pada dialog konfirmasi hapus kontak
Hasil yang Diharapkan	Sistem dapat melakukan penghapusan data kontak yang dipilih oleh pengguna melalui GUI

Pada Tabel 7.8 menunjukkan kasus uji yang digunakan dalam melakukan proses menghapus satu data kontak melalui GUI yang mana objek yang diuji adalah VC-1-4 yaitu fungsionalitas untuk menghapus kontak. Pada kasus uji ini tedapat 3 prosedur pengujian yang dilakukan.

Tabel 7.9 Kasus Uji Melakukan Panggilan Terhadap Kontak Melalui VUI

Kode Pengujian	VC-3-9
Nama Kasus Pengujian	Melakukan proses melakukan panggilan telepon terhadap kontak yang sudah terdaftar pada *smartphone* pengguna melalui VUI
Objek Uji	VC-1-5
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menelepon kontak.
Prosedur Pengujian	1. Melakukan serangkaian perintah suara mencari kontak 2. Memberikan perintah suara berupa nomor urutan kontak yang ditemukan untuk dilakukan panggilan
Hasil yang Diharapkan	Sistem dapat melakukan panggilan terhadap kontak yang dipilih melalui VUI dan membuka halaman telepon pada *smartphone* pengguna

Pada Tabel 7.9 menunjukkan kasus uji yang digunakan dalam melakukan proses menelepon salah satu kontak yang telah terdaftar pada *smartphone* pengguna melalui VUI yang mana objek yang diuji adalah VC-1-5 yaitu fungsionalitas untuk menelepon kontak. Pada kasus uji ini tedapat 2 prosedur pengujian yang dilakukan.

Tabel 7.10 Kasus Uji Melakukan Panggilan Terhadap Kontak Melalui GUI

Kode Pengujian	VC-3-10
Nama Kasus Pengujian	Melakukan proses melakukan panggilan telepon terhadap kontak yang sudah terdaftar pada *smartphone* pengguna melalui GUI
Objek Uji	VC-1-5
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menelepon kontak.
Prosedur Pengujian	1. Melakukan pencarian kontak melalui halaman cari kontak dan memilih satu kontak yang akan dipanggil.

	2. Menekan tombol pada salah satu kontak.
Hasil yang Diharapkan	Sistem dapat melakukan panggilan terhadap kontak yang dipilih melalui GUI dan membuka halaman telepon pada *smartphone* pengguna

Pada Tabel 7.10 menunjukkan kasus uji yang digunakan dalam melakukan proses menelepon salah satu kontak yang telah terdaftar pada *smartphone* pengguna melalui GUI yang mana objek yang diuji adalah VC-1-5 yaitu fungsionalitas untuk menelepon kontak. Pada kasus uji ini tedapat 2 prosedur pengujian yang dilakukan.

Tabel 7.11 Kasus Uji Melakukan Panggilan Berdasarkan Masukkan Nomor Telepon Melalui VUI

Kode Pengujian	VC-3-11
Nama Kasus Pengujian	Melakukan proses melakukan panggilan berdasarkan nomor telepon yang dimasukkan melalui VUI
Objek Uji	VC-1-6
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menelepon berdasarkan masukkan nomor telepon.
Prosedur Pengujian	1. Membuka aplikasi 2. Memberikan perintah suara “Oke” 3. Memberikan perintah suara “4” 4. Memberikan perintah suara berupa nomor telepon yang akan dipanggil 5. Memberikan perintah suara konfirmasi nomor telepon
Hasil yang Diharapkan	Sistem dapat melalukan panggilan terhadap perintah suara berupa nomor telepon yang dimasukkan oleh pengguna melalui VUI

Pada Tabel 7.11 menunjukkan kasus uji yang digunakan dalam melakukan proses menelepon berdasarkan masukkan nomor telepon oleh pengguna melalui VUI yang mana objek yang diuji adalah VC-1-6 yaitu fungsionalitas untuk

menelepon berdasarkan masukkan nomor telepon. Pada kasus uji ini tedapat 2 prosedur pengujian yang dilakukan.

Tabel 7.12 Kasus Uji Melakukan Panggilan Nomor Berdasarkan Masukkan Nomor Telepon melalui GUI

Kode Pengujian	VC-3-12
Nama Kasus Pengujian	Melakukan proses melakukan panggilan berdasarkan nomor telepon yang dimasukkan melalui GUI
Objek Uji	VC-1-6
Tujuan Pengujian	Memastikan aplikasi dapat menjalankan fungsionalitas menelepon berdasarkan masukkan nomor telepon.
Prosedur Pengujian	1. Menekan Tombol Telepon pada halaman utama 2. Memasukkan data berupa nomor telepon 3. Menekan tombol panggil
Hasil yang Diharapkan	Sistem dapat melakukan panggilan terhadap nomor telepon yang dimasukkan oleh pengguna melalui GUI

Pada Tabel 7.12 menunjukkan kasus uji yang digunakan dalam melakukan proses menelepon berdasarkan masukkan nomor telepon oleh pengguna melalui VUI yang mana objek yang diuji adalah VC-1-6 yaitu fungsionalitas untuk menelepon berdasarkan masukkan nomor telepon. Pada kasus uji ini tedapat 2 prosedur pengujian yang dilakukan.

7.1.2. Pengujian *Usability*

Dalam pengujian usablility dilakukan dengan menggunakan metode SUS yang dilakukan dengan cara melakukan uji coba aplikasi *voice call* kepada 5 orang calon pengguna. Di mana masing-masing pengguna diberikan 10 pernyataan, dikarenakan calon pengguna adalah pengguna tuna netra maka metode pengujian dilakukan dengan meyebutkan tiap pernyataan kepada calon pengguna dan pengguna diminta memberikan jawaban sesuai dengan skala SUS yaitu berkisar 1 sampai 5 yang menyatakan sangat tidak setuju sampai sangat setuju. 10 pernyataan tersebut diantaranya:

1. Saya merasa bahwa saya akan sering menggunakan aplikasi ini
2. Saya merasa bahwa aplikasi ini terlalu kompleks (sulit digunakan) padahal dapat dibuat lebih mudah untuk digunakan

3. Saya merasa bahwa aplikasi ini mudah untuk digunakan
4. Saya pikir saya membutuhkan bantuan dari orang yang ahli untuk dapat menggunakan aplikasi ini
5. Saya menemukan bahwa seluruh fungsi yang dapat dilakukan pada aplikasi ini saling terhubung dengan baik satu sama lain.
6. Saya rasa banyak hal yang tidak konsisten pada aplikasi
7. Saya merasa mayoritas orang walaupun dengan keterbatasan penglihatan dapat dengan mudah dan cepat mempelajari penggunaan aplikasi
8. Saya mendapati bahwa aplikasi ini tidak praktis ketika digunakan
9. Saya merasa sangat nyaman menggunakan aplikasi ini
10. Saya pikir saya perlu mempelajari banyak hal sebelum dapat menggunakan aplikasi ini

Setiap pengguna/responden diminta untuk memberikan penilaian terhadap kesepuluh pernyataan diatas dan hasilnya adat dilihat pada Tabel 7.13

Tabel 7.13 Hasil Pengujian SUS

	Pernyataan										
Responden		1	2	3	4	5	6	7	8	9	10
	1	5	1	5	2	5	1	4	2	5	2
	2	4	1	4	1	3	2	5	2	5	2
	3	5	1	5	2	4	1	5	3	4	1
	4	4	3	5	1	5	2	4	1	5	2
	5	5	2	4	1	4	2	5	2	5	2

Pada Tabel 7.13 menunjukkan hasil dari pengujian SUS yang dilakukan terhadap 5 orang orang responden, dan jawaban juga melampirkan jawaban mereka terhadap 10 pernyataan yang diberikan.

7.1.3. Pengujian MOS

MOS *testing* merupakan salah satu metode yang digunakan untuk menguji kepuasan pengguna terhadap komunikasi multimedia yang ditawarkan dalam bentuk *voice interaction* di mana diambil 10 orang pengguna 5 diantaranya adalah pengguna tuna netra yang digunakan sebagai subjek uji pada pengujian usabilitas dan 5 orang lagi dilakukan skenario pengguna tersebut diminta untuk menutup matanya dan memposisikan diri seolah-olah sebagai tunanetra. Seluruh pengguna diminta untuk mencoba menggunakan *voice interaction* yang ada pada aplikasi dan diminta untuk memberikan penilaian/pendapat mereka terhadap kualitas aplikasi dengan skala seperti pada Tabel 7.14.

Tabel 7.14 Skala Penilaian MOS yang Direkomendasikan Oleh ITU (*International Telecommunication Union*)

Pendapat	Skala
Excellent (Sangat Baik)	5

Good (Baik)	4
Fair (Cukup)	3
Poor (Kurang)	2
Bad (Sangat Kurang)	1

Dapat dilihat pada Tabel 7.14 menurut ITU skala pengujian pada MOS terdiri dari 1 sampai 5 yang masing-masing nilai menggambarkan pendapat pengguna terhadap aplikasi komunikasi multimedia yang disediakan. Untuk proses pengujian yang melibatkan 10 orang hasilnya dapat dilihat pada Tabel 7.14.

Tabel 7.15 Hasil Pengujian MOS *Testing*

No	Responden	Usia	Bahasa Asli	*Opinion Score*
1	Responden 1	71 Tahun	Bahasa Indonesia	5
2	Responden 2	72 Tahun	Bahasa Indonesia	4
3	Responden 3	80 Tahun	Bahasa Indonesia	4
4	Responden 4	21 Tahun	Bahasa Indonesia	5
5	Responden 5	18 Tahun	Bahasa Indonesia	5
6	Responden 6	23 Tahun	Bahasa Indonesia	3
7	Responden 7	22 Tahun	Bahasa Indonesia	4
8	Responden 8	23 Tahun	Bahasa Indonesia	5
9	Responden 9	27 Tahun	Bahasa Indonesia	3
10	Responden 10	30 Tahun	Bahasa Indonesia	4

Tabel 7.15 menjelaskan mengenai hasil dari pengujian MOS *testing* yang dilakukan pada 10 orang responden dengan rentang usia 18-75 tahun dengan bahasa asli Bahasa Indonesia.

7.2. Analisis Hasil Pengujian

7.2.1. Analisis Hasil Pengujian Validasi

Berdasarkan pengujian validasi yang dilakukan menggunakan metode *blackbox-testing* maka diperoleh hasil seperti pada Tabel 7.16 .

Tabel 7.16 Hasil Analiss Pengujian Validasi

Kode Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Status Valid
VC-3-1	Sistem dapat menyimpan data nama dan nomor kontak yang telah di masukkan oleh	Sistem berhasil menyimpan data nama dan nomor kontak yang telah di masukkan oleh	Valid

	pengguna melalui VUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	pengguna melalui VUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	
VC-3-2	Sistem dapat menyimpan data nama dan nomor kontak yang telah dimasukkan oleh pengguna melalui GUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	Sistem berhasil menyimpan data nama dan nomor kontak yang telah dimasukkan oleh pengguna melalui GUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	Valid
VC-3-3	Sistem dapat melakukan pencarian kontak berdasarkan masukkan data nama kontak oleh pengguna melalui VUI dan memberikan umpan balik suara berupa urutan nama kontak yang ditemukan apabila pengguna memasukkan data nama yang sudah terdaftar pada *smartphone* pengguna, dan akan mengembalikan umpan balik suara "Hasil Pencarian Tidak ada" dan mengulang meminta pengguna memasukkan kembali data nama apabila pengguna memasukkan data nama yang belum terdaftar pada *smartphone* pengguna.	Sistem berhasil melakukan pencarian kontak berdasarkan masukkan data nama kontak oleh pengguna melalui VUI dan memberikan umpan balik suara berupa urutan nama kontak yang ditemukan apabila pengguna memasukkan data nama yang sudah terdaftar pada *smartphone* pengguna, dan akan mengembalikan umpan balik suara "Hasil Pencarian Tidak ada" dan mengulang meminta pengguna memasukkan kembali data nama apabila pengguna memasukkan data nama yang belum terdaftar pada *smartphone* pengguna.	Valid
VC-3-4	Sistem dapat menampilkan data kontak yang ada pada *smartphone* pengguna	Sistem berhasil menampilkan data kontak yang ada pada *smartphone* pengguna	Valid

	melalui GUI dan data yang muncul pertama kali merupakan data nama kontak dengan huruf abjad paling awal.	melalui GUI dan data yang muncul pertama kali merupakan data nama kontak dengan huruf abjad paling awal.	
VC-3-5	Sistem dapat melakukan perubahan data kontak berdasarkan perintah suara yang diberikan oleh pengguna melalui VUI.	Sistem berhasil melakukan perubahan data kontak berdasarkan perintah suara yang diberikan oleh pengguna melalui VUI.	Valid
VC-3-6	Sistem dapat mengubah data kontak yang pada *smartphone* pengguna berdasarkan data yang dimasukkan oleh pengguna melalui GUI.	Sistem berhasil mengubah data kontak yang pada *smartphone* pengguna berdasarkan data yang dimasukkan oleh pengguna melalui GUI.	Valid
VC-3-7	Sistem dapat menghapus data kontak yang dipilih oleh pengguna berdasarkan perintah suara melalui VUI	Sistem berhasil menghapus data kontak yang dipilih oleh pengguna berdasarkan perintah suara melalui VUI	Valid
VC-3-8	Sistem dapat melakukan penghapusan data kontak yang dipilih oleh pengguna melalui GUI	Sistem berhasil melakukan penghapusan data kontak yang dipilih oleh pengguna melalui GUI	Valid
VC-3-9	Sistem dapat melakukan panggilan terhadap kontak yang dipilih melalui VUI dan membuka halaman telepon pada *smartphone* pengguna	Sistem berhasil melakukan panggilan terhadap kontak yang dipilih melalui VUI dan membuka halaman telepon pada *smartphone* pengguna	Valid
VC-3-10	Sistem dapat melakukan panggilan terhadap kontak yang dipilih melalui GUI dan membuka halaman	Sistem berhasil melakukan panggilan terhadap kontak yang dipilih melalui GUI dan membuka	Valid

	telepon pada *smartphone* pengguna	halaman telepon pada *smartphone* pengguna	
VC-3-11	Sistem dapat melalukan panggilan terhadap perintah suara berupa nomor telepon yang dimasukkan oleh pengguna melalui VUI	Sistem berhasil melalukan panggilan terhadap perintah suara berupa nomor telepon yang dimasukkan oleh pengguna melalui VUI	Valid
VC-3-12	Sistem dapat menyimpan data nama dan nomor kontak yang telah di masukkan oleh pengguna melalui VUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	Sistem berhasil menyimpan data nama dan nomor kontak yang telah di masukkan oleh pengguna melalui VUI dan memberikan umpan balik suara berupa "Berhasil menyimpan kontak"	Valid

Berdasarkan data yang diperoleh pada Tabel 7.16 yang dilakukan pada setiap fungsionalitas yang ada pada sistem dan diuji berdasarkan hasil yang diharapkan dan dibandingkan dengan hasil yang didapatkan, seperti yang dapat dilihat pada Tabel 7.16 bahwa semua kasus uji untuk pengujian validitas bernilai valid.

7.1.1 Analisis Pengujian Usability

Berdasarkan hasil pengujian yang diperoleh pada Tabel 7.13 dapat dilakukan perhitungan hasil dari jawaban tiap pertanyaan dengan cara sebagai berikut:

1. Untuk setiap pertanyaan yang ada pada nomor ganjil, maka nilai/skor dari jawaban dihitung dengan cara <nilai jawaban> - 1. Misalnya untuk pernyataan pada nomor 1 diperoleh skor 5 maka akan diperoleh skor untuk pernyataan pada nomor 1 yaitu 5-1=4.
2. Untuk setiap pernyataan yang ada pada nomor ganjil dilakukan perhitungan dengan cara 5 dikurangi skor yang diperoleh. Misalnya untuk pernyataan pada nomor 2 diperoleh skor sebesar 2, maka skor akhir dari pernyataan ini adalah 5-2=3.
3. Selanjutnya pada setiap nilai skor setelah perhitungan dilakukan akan dikalikan dengan nilai 2.5. Misalnya untuk pernyataan pada nomor 1 diperoleh nilai perhitungan 4 maka akan dikalikan dengan 2.5 sehingga memperoleh hasil 4*2,5=10. Begitu pula pada pernyataan selanjutnya.
4. Tahap Berikutnya adalah meghitung total skor dari 10 pernyataan yang diperoleh pada masing-masing responden, total nilai skor yang akan diperoleh berkisar 1-100.

5. Tahap terakhir adalah melakukan rata-rata nilai SUS dengan cara menjumlahkan seluruh total nilai pada tiap responden dan hasilnya akan dibagi dengan jumlah responden.

Tabel 7.17 Hasil Analisis Pengujian SUS

Responden	Pernyataan										
	1	2	3	4	5	6	7	8	9	10	Total SUS
1	10	10	10	7.5	10	10	7.5	7.5	10	7.5	90
2	7.5	10	7.5	10	5	7.5	10	7.5	10	7.5	82.5
3	10	10	10	7.5	7.5	10	10	5	7.5	10	87.5
4	7.5	5	10	10	10	7.5	7.5	10	10	7.5	85
5	10	7.5	7.5	10	7.5	7.5	10	7.5	10	7.5	85
Rata-rata Nilai Pengujian SUS											86

Berdasarkan hasil pengujian yang telah di peroleh pada Tabel 7.17 dengan menggunakan perhitungan yang telah dijabarkan sebelumnya, maka dapat dilihat bahwa nilai rata-rata dari pengujian SUS yang dilakukan mendapatkan hasil sebesar 86, dan dapat disimpulkan mendapatkan kategori *grade scale "excellent"* dan dapat dikatakan bahwa hasil pengujian *usability* pada aplikasi *voice call* bernilai sangat baik.

Untuk mengkaji secara lebih dalam mengenai hasil dari SUS dapat dilakukan dengan menggunakan perhitungan *Relative Importance Indeks* (RII) seperti pada Persamaan 7.1 yang digunakan untuk mendapatkan hasil persentase nilai dari tiap pernyataan yang diperoleh dari kelima responden, semakin besar persentase yang diperoleh maka dapat disimpulkan dari pernyataan tersebut memperoleh nilai sangat setuju begitu pula sebaliknya semakin kecil nilai persentase yang diperoleh maka rata-rata responden semakin tidak setuju dengan pernyataan tersebut.

$$RII\ (\%) = \left(\frac{n1+2n2+3n3+4n4+5n5}{5(n1+n2+n3+n4+n5)}\right) x\ 100 \qquad (7.1)$$

Hasil dari persentase RII dapat dilihat pada Tabel 7.18 Cara untuk menghitung RII dapat dilakukan sebagai berikut:

1. Dari 10 pernyataan yang diberikan kepada responden hitung jumlah setiap pilihan jawaban yang diperoleh. Contohnya dalam pengujian yang dilakukan terhadap 5 orang responden di pernyataan nomor 1 ada sebanyak 0 orang yang memilih pilihan jawaban nomor 1, 2 dan 3. Sedangkaan ada sebanyak 2 orang yang menjawab pilihan jawaban nomor 4 dan ada 3 orang.
2. Selanjutnya hitung setiap pilihan jawaban yang diperoleh dari tiap pernyataan dan tuliskan dalam sebuah tabel untuk mempermudah perhitungan selanjutnya.
3. Setelah itu tentukan nilai dari n (1,2,3,4,5) yang menunjukkan jumlah responden yang menjawab pilihan jawaban tersebut misalnya untuk pernyataan nomor 1 jumlah responden yang menjawab pilihan jawabab nomor 1 adalah sebanyak 0 orang maka n1=0, begitu pula untuk n2,n3,n4 dan n5.

4. Tahap berikutnya adalah hitung nilai RII berdasarkan rumus yang ada pada Persamaan 7.1. Contohnya untuk pernyataan nomor 1

$$RII\ (\%) = \left(\frac{0 + 2(0) + 3(0) + 4(2) + 5(3)}{5(0 + 0 + 0 + 2 + 3)}\right) x\ 100$$

$$RII\ (\%) = \left(\frac{23}{25}\right) x\ 100$$

$$RII\ (\%) = 0.92\ x\ 100$$

$$RII = 92\%$$

Tabel 7.18 Hasil Persentase RII Pada Pengujian SUS

Pilihan Jawaban dari 5 Responden							
Pernyataan		1	2	3	4	5	Hasil Persentase
	1	0	0	0	2	3	92%
	2	3	1	1	0	0	32%
	3	0	0	0	2	3	92%
	4	3	2	0	0	0	28%
	5	0	0	1	2	2	84%
	6	2	3	0	0	0	32%
	7	0	0	0	2	3	92%
	8	1	3	1	0	0	40%
	9	0	0	0	1	3	76%
	10	1	4	0	0	0	36%

Berdasarkan hasil yang diperoleh pada Tabel 7.18 yang menunjukkan hasil persentase RII yang diperoleh dari tiap pernyatan. Dapat dilihat pada Tabel 7.18 pernyataan yang memperoleh persentase paling besar adalah pernyataan nomor 1 dan 7 hal ini menujukkan bahwa pernyataan nomor 1 dan 7 merupakan pernyataan yang memperoleh hasil paling mendekati sangat setuju dari responden. Sedangkan untuk pernytaan dengan persentase paling kecil adalah pernyataan nomor 4 yaitu sebesar 28% hal ini menunjukkan bahwa pernyataan nomor 4 yang memperoleh hasil mendekati paling tidak setuju dari responden.

Analisis Pengujian MOS *testing*

Analisis pengujian yang dilakukan terhadap MOS *testing* diperoleh berdasarkan dari hasil perhitungan nilai rata-rata yang dilakukan terhadap nilai yang diperoleh pada pengujian MOS *testing* pada Tabel 7.15. Perhitungan nilai rata-rata skor pada MOS *testing* dapat dilakukan dengan menggunakan rumus yang ada pada Persamaan 7.2.

$$MOS = \frac{1}{N}\sum_{i=1}^{N} yi \qquad (7.2)$$

$$MOS = \frac{5 + 4 + 4 + 5 + 5 + 3 + 4 + 5 + 3 + 4}{10}$$

$$MOS = \frac{42}{10}$$

$$MOS = 4{,}2$$

Berdasarkan hasil pehitungan nilai rata-rata akhir dari MOS *testing* diperoleh hasil dari pengujian MOS *testing* pada aplikasi ini mendapatka nilai 4,2. Hal ini berarti bahwa layanan interaksi suara pada komunikasi yang disediakan aplikasi *voice call* ini memiliki fungsi yang baik (*Good*) karena mendapatkan nilai akhir diatas 4.

BAB 8 KESIMPULAN DAN SARAN

8.1. Kesimpulan

Berdasarkan seluruh rangkaian proses yang dilakukan pada penelitian ini, maka diperoleh kesimpulan sebagai berikut.

1. Aplikasi *voice call* dengan manajemen kontak berbasis suara bagi pengguna tuna netra telah dirancang dalam beberapa tahapan perancangan dengan menggunakan pendekatan berorientasi objek yang dimulai dari perancangan *sequence diagram, class diagram,* perancangan antarmuka dan *screenflow* yang mengacu pada tahap analisis kebutuhan yang telah dilakukan pada tahap sebelumnya. Dan Aplikasi ini juga berhasil diimplementasikan dengan bahasa pemrograman Typescript dan HTML dengan memanfaatkan *Framework* Ionic, serta modul Google *Speech Recogrnition* API dan Google *Text-toSpeech* sebagai modul pemrosesan suara.
2. Berdasakan hasil pengujian *usability* yang dilakukan dengan menggunakan metode SUS, maka diperoleh hasil sebesar 86 yang menunjukkan bahwa aplikasi sudah berada pada skala "*excelent*" sehingga dapat disimpulkan aplikasi mudah digunakan oleh pengguna.
3. Nilai rata-rata dari pengujian MOS *testing* yang diperoleh adalah sebesar 4,2 sehingga hal ini menunjukkan bahwa kinerja dari pemrosesan suara Google *Text-to-Speech* dan Google *Speech Recognition* memiliki nilai yang baik (*good*).

8.2. Saran

Setelah serangkaian proses pada penelitian ini berhasil dilakukan, maka diberikan saran yang bertujuan untuk memberikan masukan bagi pengembangan sistem sejenis yang ingin dilakukan. Adapun saran yang diberikan adalah sebagai berikut.

1. Pengembang aplikasi dapat menyediakan layanan *voice call* tersendiri sehingga pengguna dapat terhubung dengan panggilan suara melalui jaringan internet.
2. Pengembang aplikasi dapat meneliti lebih lanjut mengenai pemanfaatan modul suara yang dapat digunakan tanpa memerlukan koneksi internet.
3. Pengembang aplikasi dapat memanfaatkan layanan *voice intercation* dengan menggunakan Google API dalam pengembangan aplikasi selain untuk melakukan panggilan telepon (*voice call*).

DAFTAR PUSTAKA

Alauddin, M. W., Kurniawan, W. & Setiawan, B. D., 2017. Rancang Bangun Alat Pendeteksi Suara Panggilan Manusia Berbahasa Indonesia Untuk Tunarungu Menggunakan Library PocketSphinx Berbasis Embedded system. *Proceedings of the 4th IEEE Global Humanitarian Technology Conference, GHTC 2014,* [online] Tersedia di < https://www.researchgate.net/publication/320345197_Rancang_Bangun_Alat_Pendeteksi_Suara_Panggilan_Manusia_Berbahasa_Indonesia_Untuk_Tunarungu_Menggunakan_Library_PocketSphinx_Berbasis_Embedded_system> [Diakses 10 Mei 2018]

Bangor, A., Kortum, P. & Miller, J., 2009. Determining What Individual SUS Scores Mean. *Journal of usability studies,* 4(3), pp. 114-123.

BPS, 2018. *Persentase Penduduk yang Memiliki/Menguasai Telepon Seluler Menurut Provinsi dan Klasifikasi Daerah, 2012-2016.* [online] Tersedia di: <https://www.bps.go.id/dynamictable/2015/11/10/985/persentase-penduduk-yang-memiliki-menguasai-telepon-seluler-menurut-provinsi-dan-klasifikasi-daerah-2012-2016.html>
[Diakses 9 November 2018]

Brooke, J., 1996. SUS - A quick and dirty usability scale. *Usability evaluation in industry. CRC Press,* 2(1), pp. 189-194.

Colenbrander, A., 2007. Visual standards: Aspects and ranges of vision loss. *International Council of Ophtalmology,* 14(3), pp. 199-203.

Garibay, F. R., Olivarria, C. M., Eufracio Aguilera, A. F. & Huegel , J. C., 2014. MyVox—Device for the communication between people: blind, deaf, deaf-blind and unimpaired. *Proceedings of the 4th IEEE Global Humanitarian Technology Conference, GHTC 2014,* [online] Tersedia di: < https://ieeexplore.ieee.org/document/6970330/> [Diakses 04 April 2018]

Garrett, J. J., 2003. *The elements of user experience: user-centered design fo the web and beyond.* 2nd ed. Berkeley: Jesse James Garrett.

Gladden , D., 2018. *The Effects of Smartphones on Social Lives : How They Affect Our Social Interactions and Attitudes.* Norfolk, OTS Master's Level Projects & Papers, p. 32.

GlobalStats, 2018. *Operating System Market Share Indonesia | StatCounter Global Sta.* [online]
Tersedia di: <http://gs.statcounter.com/os-market-share/all/indonesia>
[Diakses 3 November 2018].

Hansen, G. C., Falkenbach, K. H. & Yaghmai, I., 2010. Voice Recognition System. *Radiology,* 169(1), p. 580.

IDC, 2015. *Worldwide Smartphone OS Market Share.* [online] Tersedia di: <https://www.idc.com/promo/smartphone-market-share/os> [Diakses 23 Agustus 2018]

Indrawati, 2016. *Angka Kebutaan di Indonesia Capai 3.750.000 Orang.* [online] Tersedia di: <http://www.aktual.com/220713-2/> [Diakses 20 April 2018].

ISO, 2010. *ISO 9241-210 Ergonomics of human-system interaction.* [online] Tersedia di: <https://www.iso.org/standard/52075.html> [Diakses 28 Mei 2018].

Johnsen, A. D., Grønli, T.M. & Bygstad, B., 2010. MAKING TOUCH-BASED MOBILE PHONES ACCESSIBLE FOR THE VISUALLY IMPAIRED, [online] Tersedia di: <http://www.nik.no/2012/4-2-johnsen12MakingTouch-BasedMobilePhonesAccessibleForTheVisuallyImpaired.pdf> [Diakses 20 Mei 2018]

Justicia, L. T., 2017. Rancang Bangun Aplikasi Messaging Berbasis Voice Interaction Bagi Penderita Tunanetra Pada Sistem Operasi Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 1(7), pp. 620-627.

Justicia, L. T., 2017. *Rancang Bangun Aplikasi Messaging Berbasis Voice Intercation Bagi Penderita Tunanetra Pada Sistem Operasi Android.* S1. Universitas Brawijaya.

Kardyś, P., Dąbrowski, A., Iwanowski, M. & Huderek, D., 2016. *A new Android application for blind and visually impaired people.* Poznan, Poznan University of Technology, [online] Tersedia di: < http://ieeexplore.ieee.org/document/7763604/> [Diakses 4 April 2018]

Këpuska, V. Z. & Bohouta, G., 2017. Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *International Journal of Engineering Research and Applications,* 7(3), pp. 20-24.

Kominfo, 2015. *Indonesia Raksasa Teknologi Digital Asia.* [online] Tersedia di: <https://www.kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media> [Diakses 9 November 2018]

Martz, E., ed.,2018. *Promoting Self-Management of Chronic Health Conditions: Theories and Practice.* New York: Oxford University Press.

Pagliari, C., 2007. Design and Evaluation in eHealth: Challenges and Implications for an Interdisciplinary Field. *JOURNAL OF MEDICAL INTERNET RESEARCH,* 9(2), pp. 1-10.

Popović, G. & Pale, U., 2016. *Audio phonebook for the blind people Goran,* Croatia: Faculty of Electrical Engineering and Computing, [online] Tersedia di: < https://ieeexplore.ieee.org/document/7522402/> [Diakses 4 April 2018]

Rachma, H. D., Rugmiaga, Z. & Huda , M., 2011. *Pembuatan Text-To-Speech Synthesis System Untuk Penutur Berbahasa Indonesia.* Surabaya, Politeknik

Elektronika Negeri Surabaya, Institut Teknologi Sepuluh November Surabaya Kampus ITS Sukolilo.

Ribeiro, F., Florêncio, D., Zhang, C. & Seltzer, M., 2011. CROWDMOS: An Approach for Crowdsourcing Mean Opinion Score Studies. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings,* pp. 2416-2419.

Rofiq, M. & Putri, S. I., 2017. Perancangan Sistem Pemesanan Rumah Sakit di Kota Malang Menggunakan Ionic Framework Berbasis Mobile Phone. *Jurnal Ilmiah Teknologi Informasi Asia,* I(2), pp. 171-178.

Samsudin & Putra, R. Y., 2016. PERANCANGAN APLIKASI TEXT TO SPEECH PENGENALAN. *Konferensi Nasional Pengembangan Teknologi Informasi dan Komunikasi,* 4(1), pp. 391-398.

Sianturi, A. H., 2014. *Implementasi Speech Recognition Pada Pembelajaran Dalam Bentuk Permainan Menebak Kata Baku Bahasa Indonesia.* S1. Universitas Sumatera Utara.

Stackshare, 2018. *Ionic vs Xamarin vs PhoneGap 2018 Comparison of Cross-Platform Mobile Development.* [online] Tersedia di: <https://stackshare.io/stackups/ionic-vs-phonegap-vs-xamarin> [Diakses 29 November 2018].

Tolle, H., Pinandito, A., Kharisma, A. P. & Dewi, R. K., 2017. *Pengembangan Aplikasi Perangkat Bergerak (Konsep&Implementasi).* 1st ed. Malang: UB Press.

Usability.gov, 2017. *System Usability Scale (SUS).* [online] Available at: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> [Diakses 12 September 2018].

W3C, 2015. *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile,* [online] Tersedia di: <https://www.w3.org/TR/mobile-accessibility-mapping/> [Diakses 12 Agustus 2018]

WHO, 2017. *Blindness and visual impairment.* [online] Tersedia di: <http://www.who.int/mediacentre/factsheets/fs282/en/> [Diakses 20 April 2018]