

Penyuaran Pesan Teks Media Sosial Pada Perangkat Mobile Menggunakan Text To Speech

Michael Alexander Rustan¹, Anita Nathania P.², Liliana³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

Email: c14180175@john.petra.com¹, anita.nathania@petra.ac.id³, lilian@petra.ac.id²

ABSTRAK

Penggunaan *smartphone* sebagai alat berkomunikasi akan semakin meningkat bersamaan dengan meningkatnya pengguna *smartphone*. Berdasarkan data yang didapat, pada awal tahun 2021 terdapat sekitar 167 juta pengguna *smartphone* di Indonesia. Penggunaan *smartphone* sebagai alat berkomunikasi untuk mengirim pesan dalam penggunaannya juga dapat mulai mengganggu masyarakat apabila mereka sedang beraktivitas yang membutuhkan konsentrasi, contohnya seperti berkendara. Pada tahun 2017, terdapat sekitar 15,341 kasus kecelakaan di Amerika yang disebabkan karena pengendara menggunakan *smartphone*. Untuk mengatasi masalah tersebut, fitur *text to speech* akan digunakan untuk menyuarakan pesan-pesan yang masuk tersebut sehingga pengguna *smartphone* tidak perlu membuka *smartphone* mereka untuk mengetahui pengirim dan isi pesan tersebut.

Hasil dari pengujian yang dilakukan terhadap penggunaan fitur *text to speech* menunjukkan bahwa sistem dapat menyuarakan pesan-pesan yang masuk dengan baik. Untuk pesan-pesan yang memiliki kata singkatan, fitur *text to speech* tidak dapat menyuarakannya dengan baik. Sedangkan untuk pengujian yang dilakukan terhadap fitur untuk mendeteksi pesannya, sistem dapat mendeteksi beberapa data pesan seperti nama *package* dari aplikasi, nama pengirim, dan juga isi pesan dengan baik. Untuk pesan yang diterima melalui obrolan grup, hasil deteksi dari aplikasi *line*, dan aplikasi *whatsapp* memiliki kekurangan, sehingga terdapat kendala dalam proses penyuaran pesan obrolan grup pada aplikasi *line* dan *whatsapp*.

Kata Kunci: *Text-To-Speech, Notifikasi, Mobile Application, Notification Listener Service, Broadcast Receiver, Kotlin*

ABSTRACT

The use of smartphones as a communication tool will increase along with the increase of smartphone users. Based on the data obtained, in early 2021 there are around 167 million smartphone users in Indonesia. The use of smartphone as a communication tool to send messages in its use can also start to annoy people when they are doing activities that require concentration, for example, such as driving. In 2017, there were about 15,341 cases of accidents in America caused by drivers using smartphone. To overcome this problem, the text to speech feature will be used to voicing the incoming messages so smartphone users do not need to open their smartphone to find out the sender and the contents of the message.

The results of the tests that carried out on the usage on text to speech feature showed that the system can voiced the incoming

messages well. For messages that have abbreviated words, the text to speech feature cannot voiced them properly. As for the tests carried out on the feature to detect the message, the system can detect some message data such as the package name of the application, the sender's name, and also the message content properly. For messages received through group chat, the detection results from the line application, and the whatsapp application have problems, so there are obstacles in the process of voicing group chat messages on the line application and whatsapp applications.

Keywords: *Text-To-Speech, Notification, Mobile Application, Notification Listener Service, Broadcast Receiver, Kotlin.*

1. INTRODUCTION

Berdasarkan data yang didapat, pada awal tahun 2021, terdapat sekitar 167 juta pengguna *smartphone* di Indonesia. Dengan banyaknya pengguna *smartphone* ini, penggunaan *smartphone* sebagai sarana komunikasi pun semakin meningkat baik berupa pesan teks ataupun telepon. Tetapi pesan-pesan teks yang masuk itu pun dapat mulai mengganggu masyarakat apabila mereka sedang beraktivitas yang membutuhkan konsentrasi, contohnya seperti berkendara. Penggunaan *smartphone* waktu berkendara dapat merugikan pengendara tersebut seperti mendapatkan hukum pidana, kehilangan konsentrasi, menghambat lalu lintas dan juga yang terburuk adalah mengalami kecelakaan. Terdapat sekitar 34,247 kasus kecelakaan di Amerika pada tahun 2017 yang disebabkan oleh pengendara yang tidak fokus dalam mengendarai kendaraannya. Dan dari kasus tersebut terdapat 15,341 kasus kecelakaan yang disebabkan karena pengendara menggunakan *smartphone*.

Di Indonesia, ada kebijakan yang ditetapkan oleh pemerintah yaitu mengganggu konsentrasi saat berkendara adalah karena sakit, lelah, mengantuk, menggunakan handphone, menonton televisi atau video yang terpasang di kendaraan, hingga pengaruh minuman beralkohol dan obat-obatan. Dengan adanya kebijakan ini, menunjukkan bahwa kasus ini juga serius dan banyak yang melakukan pelanggaran ini dan juga banyak mencelakai orang lain. Oleh karena itu, untuk mengatasi masalah ini tanpa melanggar kebijakan dari pemerintah, akan dibuat suatu aplikasi dimana aplikasi tersebut nantinya dapat mendeteksi apabila ada pesan yang masuk, lalu pesan tersebut akan langsung disuarakan sehingga pengendara tidak perlu membuka *smartphone* untuk mengetahui isi pesan.

Ada beberapa aplikasi yang dapat meminimalisir masalah ini, antara lain aplikasi *Android based voice assistant for blind people*, aplikasi *Audify*, dan aplikasi *Shouter*.

Untuk Aplikasi *voice assistant* memiliki keunggulan seperti dapat menyuarakan notifikasi dari sms yang masuk, dapat melakukan panggilan telepon, dapat membuka aplikasi lain, dan dapat mengirimkan suatu pesan sms. Semua keunggulan tersebut dilakukan dengan menggunakan *voice command* dari pengguna. Untuk kekurangan dari aplikasi ini adalah terbatasnya aplikasi yang dapat disuarakan yaitu hanya aplikasi sms saja [2].

Aplikasi Audify merupakan aplikasi yang dapat menyuarakan pesan dari suatu notifikasi dan memiliki beberapa fitur, seperti fitur untuk mengganti bahasa penyuaran ke beberapa bahasa, dan Audify *locate* untuk menyimpan posisi mobil yang sedang terparkir. Untuk kekurangan dari aplikasi ini adalah apabila ada pesan-pesan yang masuk hampir di waktu yang bersamaan, maka pesan yang pertama akan tertutup dan digantikan oleh pesan yang baru,

Aplikasi Shouter merupakan aplikasi yang dapat menyuarakan pesan dari suatu notifikasi dan memiliki beberapa fitur seperti mengganti bahasa penyuaran ke beberapa bahasa, dapat mengubah beberapa pengaturan suara seperti *pitch* suara, volume, dan terdapat isi *history* notifikasi. Untuk kekurangan dari aplikasi ini adalah apabila ada pesan-pesan yang masuk hampir di waktu yang bersamaan, maka pesan yang pertama akan tertutup dan digantikan oleh pesan yang baru.

2. LANDASAN TEORI

2.1 Notification Listener

Notification Listener Service merupakan suatu *service* yang menerima panggilan apabila ada suatu notifikasi yang baru masuk ke dalam *smartphone* secara *detail*. Pertama kali, *Notification Listener* dirilis pada *Android 4.3*. Secara khusus, *service* ini mendeteksi aplikasi mana yang memicu atau menghapus notifikasi, waktu notifikasi, tingkat prioritas, pola getaran, dan suara di antara beberapa atribut tambahan. Pengguna harus memberikan akses notifikasi terlebih dahulu agar aplikasi yang menggunakan *service* tersebut dapat mengakses notifikasi yang masuk [6], [10].

Untuk menggunakan *Notification Listener*, yang pertama adalah harus membuat *class NotificationListenerService()*. Kemudian, menentukan nama-nama *list package* dari aplikasi yang ingin dideteksi notifikasinya. Untuk mendeteksi apakah ada notifikasi yang masuk adalah menggunakan fungsi *onNotificationPosted()*. Lalu di dalam fungsi *onNotificationPosted()*, notifikasi yang masuk tersebut dapat diterima dan digunakan.

2.2 Sensor Event Listener

SensorEventListener digunakan untuk menerima sensor pada *smartphone* apabila terdeteksi data sensor yang baru [9]. Pada skripsi ini, sensor yang digunakan adalah sensor gerak untuk menghitung akselerasi pada perangkat *smartphone*. Secara konsep, sensor akselerasi menentukan akselerasi yang diterapkan pada perangkat A_D dengan mengukur gaya yang diterapkan pada sensor itu sendiri (F_G). Tetapi, seperti pada Persamaan (1), gaya gravitasi akan selalu memengaruhi akselerasi yang diukur dengan persamaan:

$$A_D = -g - \left(\frac{1}{mass}\right) \sum F_S \quad (1)$$

Untuk mengukur akselerasi nyata dari *smartphone*, kontribusi gaya gravitasi harus dihapus terlebih dahulu dengan menerapkan *filter high-pass*, dan *filter low-pass* yang digunakan untuk mengisolasi gaya gravitasi [1]

2.3 Broadcast Receiver

BroadcastReceiver merupakan suatu cara yang digunakan untuk mengirimkan suatu *intent* ke seluruh komponen dan aplikasi. *BroadcastReceiver* juga dapat digunakan untuk menerima *broadcast* baik dari *android runtime* atau aplikasi lain [3], [4]. Contoh penggunaan *BroadcastReceiver* adalah pada saat aplikasi ingin menerima *broadcast* dari SMS. Apabila ada pesan SMS yang masuk, sistem akan secara otomatis mengirimkan *broadcast* pesan SMS tersebut ke aplikasi yang mendaftarkan untuk menerima *broadcast* dari pesan SMS. Cara mendaftarkan aplikasi terhadap *broadcast*, dapat dilakukan dengan 2 cara, yang pertama adalah dengan mendeklarasikan *receiver* serta *IntentFilter* pada *manifest*. Dan cara kedua adalah dengan mendaftarkan *receiver* pada *context*.

2.4 Text To Speech

Text to Speech merupakan suatu teknologi yang berguna untuk mengkonversi atau mengubah suatu teks tertulis menjadi suatu ucapan atau suara [7], [8]. *Text to Speech* sangat berguna untuk orang yang memang memiliki kekurangan dalam penglihatan dan juga bagi orang-orang tertentu yang memang lebih suka mendengar daripada melihat.

Dalam hal *computer science*, ada beberapa metode untuk melakukan konversi pada *text to speech*, yaitu *Concatenative TTS*, *Statistical Parametric TTS*, *Hybrid Approaches* [5]. Pada aplikasi yang akan dibuat, metode *text to speech* yang digunakan adalah *library* dari *Android* yang menggunakan *Parametric text to speech*.

2.5 Google Text To Speech

Google Text To Speech yang dibuat oleh *Google* terdiri dari 2 jenis, yaitu *Google Text To Speech Android Application* dan *Google Cloud Text To Speech*.

Google Text to Speech Android Application merupakan suatu fitur yang terdapat pada aplikasi *Speech Services* yang dibuat oleh *Google* dan dapat digunakan secara gratis. *Google Text to Speech* dapat membantu pengguna untuk membaca suatu teks yang terdapat pada layar *smartphone* dengan suara. Sebagai contohnya, fitur ini dapat diimplementasikan pada Aplikasi yang berhubungan dengan *talkback* dan *accessibility* untuk menyuarakan *feedback* di seluruh perangkat pengguna.

Google Cloud Text to Speech merupakan suatu *engine text to speech* berbayar yang memungkinkan pengembang untuk membuat penyuaran suara sintetis yang natural, yang terdengar alami sebagai audio. Pengembang dapat menggunakan data audio yang dibuat pengembang sendiri untuk aplikasi yang di kembangkan.

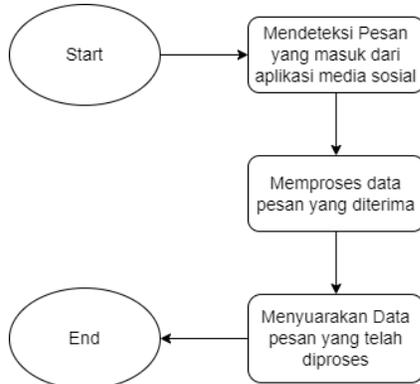
3. DESAIN SISTEM

Untuk mengatasi permasalahan dalam penggunaan *smartphone* sebagai alat komunikasi pesan teks, maka akan di desain suatu aplikasi yang dapat memberi informasi kepada pengguna mengenai isi pesan yang masuk dengan memberikan penyuaran terhadap isi pesan tersebut.

3.1 Alur Utama Program

Alur utama program seperti yang terdapat pada Gambar 1, sistem pertama-tama akan mulai dengan mendeteksi pesan yang masuk dari setiap aplikasi yang terdaftar pada sistem, seperti Instagram, Line, Whatsapp, Telegram, Facebook, dan SMS. Pada proses ini, sistem akan menggunakan *Notification Listener Service* dan *Broadcast Receiver* untuk memperoleh data pesan

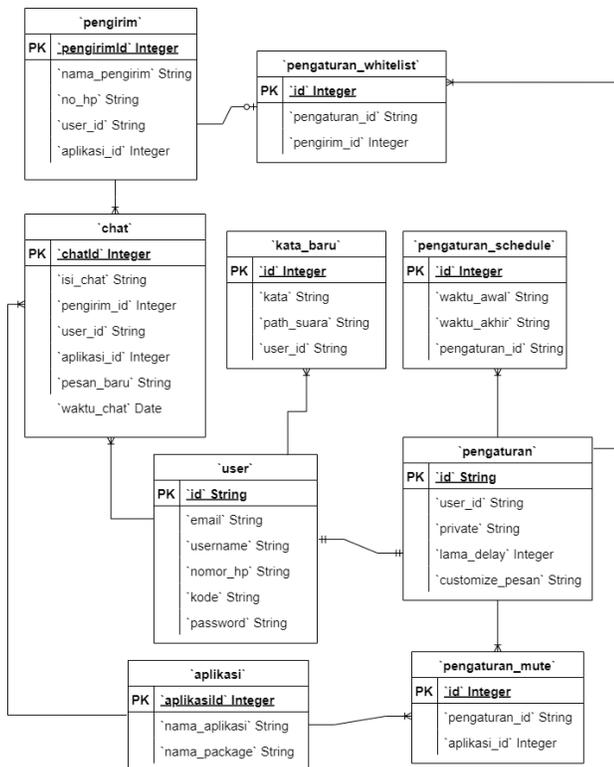
yang dikirimkan. Kemudian, sistem akan memproses data pesan tersebut dan mengambil data-data yang diperlukan seperti, nama pengirim, isi pesan, waktu pesan dikirimkan, dan nama aplikasi. Dan proses terakhir adalah sistem akan menyuarkan data pesan yang telah di proses sebelumnya dengan mengecek data pengaturan seperti data *whitelist*, *mute*, *schedule*, dan *customize* pesan pembuka.



Gambar 1. Flowchart alur utama program

3.2 Entity Relationship Diagram

Entity Relationship Diagram digunakan untuk menjelaskan mengenai relasi-relasi pada database untuk setiap entitas yang terdapat pada sistem. Entitas yang terdapat pada aplikasi ini antara lain user, aplikasi, pengirim, chat, kata_baru, pengaturan, pengaturan_whitelist, pengaturan_mute_aplikasi, dan pengaturan_schedule. Berikut pada Gambar 2, merupakan Entity Relationship Diagram pada sistem



Gambar 2. Entity Relationship Diagram pada sistem

4. PENGUJIAN SISTEM

4.1 Sistem Mendeteksi Pesan Masuk

Pengujian ini dilakukan untuk memastikan bahwa setiap aplikasi yang terdaftar dapat menerima pesan dari semua aplikasi yang terdaftar. Pengujian ini dilakukan dengan menggunakan 2 *smartphone* yaitu *smartphone A* untuk menerima pesan, dan *smartphone B* untuk mengirim pesan ke *smartphone A*. Pada pengujian ini model *smartphone A* adalah *Xiaomi Mi10T Pro*. Berikut hasil percobaan untuk pesan yang dikirimkan pada obrolan personal.

Tabel 1. Pengujian Sistem mendeteksi pesan masuk obrolan personal

Aplikasi	Pesan yang dikirim	Hasil deteksi pesan
Aplikasi lain	“Halo tes”	“Halo tes”
Instagram	“Halo tes”	“<nama_pengirim atau username_pengirim>: Halo test”

Tabel 2. Pengujian sistem mendeteksi pesan masuk obrolan personal

Aplikasi	Nama pengirim	Hasil deteksi nama pengirim
Aplikasi lain	“Nama Pengirim”	“Nama Pengirim”
SMS	“Nama Pengirim”	“Nama Pengirim”
SMS (Tidak tersimpan)	“Nama Pengirim”	“Nomor Pengirim”

Membahas mengenai hasil pengujian penerimaan pesan untuk obrolan grup pada Tabel 1 dan Tabel 2, data isi pesan yang dideteksi pada aplikasi *whatsapp*, SMS, *facebook*, *line*, dan *telegram*, sudah sesuai dengan pesan yang dikirimkan. Sedangkan untuk aplikasi *Instagram*, *username Instagram* dari pengirim juga dimasukkan kedalam data isi pesan sehingga hasil deteksi pesan tidak sesuai dengan pesan yang dikirimkan. Sedangkan untuk data nama pengguna, pada aplikasi *whatsapp*, *facebook*, *line*, *telegram*, *instagram* dan SMS hasil deteksi nama pengirim sudah sesuai dengan nama pengirim. Tetapi untuk aplikasi SMS, apabila nomor pengirim tidak tersimpan pada kontak, data nama pengirim yang dideteksi adalah nomor hp dari pengirim.

Berikut hasil percobaan untuk pesan yang dikirimkan pada obrolan grup.

Tabel 3. Tabel pengujian sistem mendeteksi pesan obrolan grup

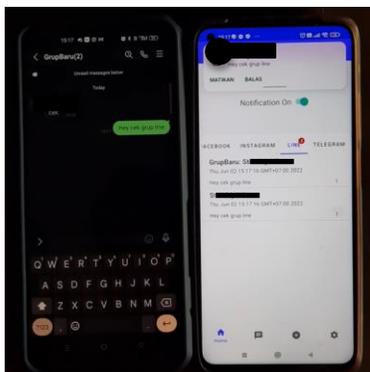
Aplikasi	Pesan yang dikirim	Hasil Deteksi Pesan
Aplikasi lain	“Halo cek grup”	“Halo cek grup”
Instagram	“Halo cek grup”	“<nama_pengirim atau username_pengirim> ke “Grup baru”: Halo cek grup”

Tabel 4. Tabel pengujian sistem mendeteksi pesan obrolan grup

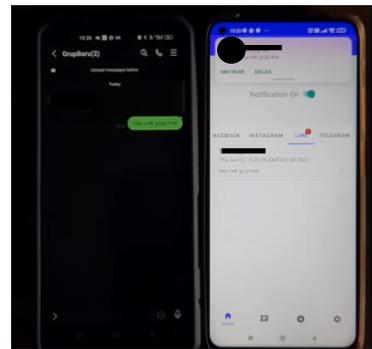
Aplikasi	Nama Grup dan Pengirim	Hasil deteksi nama pengirim
Whatsapp, Facebook & Telegram	“Grup baru” dan “Nama Pengirim”	“Grup baru: Nama Pengirim”
Line	“GrupBaru” dan “Nama Pengirim”	“GrupBaru: Nama Pengirim” & “Nama Pengirim”
Instagram	“Grup Baru” dan “Nama Pengirim”	“Grup Baru”

Membahas mengenai hasil pengujian penerimaan pesan untuk obrolan grup pada Tabel 3 dan Tabel 4, data isi pesan yang dideteksi pada aplikasi *whatsapp*, *facebook*, *line*, dan *telegram*, sudah sesuai dengan pesan yang dikirimkan. Sedangkan untuk aplikasi *Instagram*, nama pengirim atau *username Instagram* dari pengirim dan juga nama grup akan dimasukkan ke dalam data isi pesan sehingga hasil deteksi pesan tidak sesuai dengan pesan yang dikirimkan. Sedangkan untuk data nama pengguna, pada aplikasi *whatsapp*, *facebook*, *line*, dan *telegram* hasil deteksi nama pengirim adalah gabungan antara nama grup dan nama pengguna. Sedangkan untuk aplikasi *instagram*, data nama pengirim yang dideteksi adalah nama grup.

Pada pengujian pesan obrolan grup pada aplikasi *line* akan mengirimkan 2 cara pengiriman. Cara pertama adalah aplikasi *line* mengirimkan 2 notifikasi, notifikasi pertama berisi isi pesan dan nama akun *line* dari pengirim yang di gabungkan dengan nama grup, untuk notifikasi kedua berisi isi pesan dan nama akun *line* dari pengirim (Gambar 3). Cara pengiriman kedua, aplikasi *line* hanya mengirimkan 1 notifikasi saja, dimana data pesannya adalah isi pesan, dan nama akun *line* dari pengirim (Gambar 4)

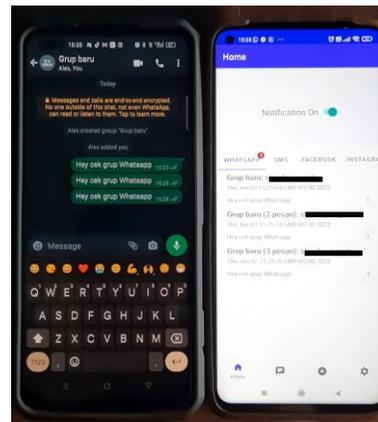


Gambar 3. Data pesan dari grup pada aplikasi line (Cara pengiriman pertama)



Gambar 4. Data pesan dari grup pada aplikasi line (Cara pengiriman kedua)

Pada pengujian pesan obrolan grup pada aplikasi *whatsapp*, apabila pada grup tersebut terdapat lebih dari 1 pesan yang belum dibaca, dan ada pesan yang masuk, maka nama pengirim yang terdeteksi adalah gabungan dari nama grup, jumlah pesan yang belum di baca, nama akun *whatsapp* dari pengirim (Gambar 5) .



Gambar 5. Data pesan dari grup pada aplikasi whatsapp

4.2 Sistem mengimplementasi *text to speech* dan *speech to text*

Pengujian ini dilakukan untuk memastikan bahwa fitur *text to speech* dan *speech to text* dapat berjalan dengan baik. Pengujian ini dibagi menjadi 2 bagian yaitu *text to speech* dan *speech to text*. Pada pengujian ini model *smartphone A* adalah *Xiaomi Mi10T Pro*

4.2.1 Pengujian *Text to Speech*

Pada pengujian *text to speech* dibagi menjadi 7 pengujian antara lain *text to speech* dengan bahasa baku, *text to speech* dengan bahasa baku dalam beberapa kalimat, *text to speech* dengan bahasa tidak baku, *text to speech* dengan bahasa gaul, *text to speech* dengan bahasa indonesia singkatan, *text to speech* bahasa baku, tidak baku, dan singkatan dan pengujian yang terakhir adalah *text to speech* dengan tambahan kata.

Tabel 5. Pengujian *text to speech*

Isi pesan dalam bentuk	Pesan yang dikirim	Hasil Penyuaraan pesan dengan <i>text to speech</i>

Bahasa baku	“halo, merek baterai itu berapa miliar?”	“halo, merek baterai itu berapa miliar?”
Bahasa baku dalam beberapa kalimat	“Halo, merek baterai itu berapa miliar? Saya mendengar bahwa harga baterai itu satu miliar. Apakah benar?”	“Halo, merek baterai itu berapa miliar? Saya mendengar bahwa harga baterai itu satu miliar. Apakah benar?”
Bahasa tidak baku	“halo, merk batere itu berapa milyar?”	“halo, merk batere itu berapa milyar?”
Bahasa gaul	“hei, lu kenal kagak sama gue?”	“hei, lu kenal kagak sama gue?”
Bahasa indonesia singkatan,	“halo, km lg apa?”	“halo, kem eleg apa?”
Bahasa baku, tidak baku, gaul, singkatan	“halo, ak mendengar bahwa lu lagi gabut, gue mau mengajak km jalan. Loe mau tdk ntar malem?”	“halo, ak mendengar bahwa lu lagi gabut, gue mau mengajak km jalan lu mau tedeka?”
Bahasa baku, tidak baku, gaul, singkatan, dengan tambahan kata	“halo, ak mendengar bahwa lu lagi gabut. Gue mau mengajak km jalan, loe mau tdk ntar malem?”.	“halo, aku mendengar bahwa lu lagi gabut. Gue mau mengajak kamu jalan, loe mau tidak ntar malem?”

Dari hasil pengujian pada Tabel 5, penyuaaran yang tidak sesuai atau tidak lancar adalah saat isi pesan yang dikirimkan terdapat kata singkatan.

4.2.2 Pengujian Speech to Text

Pada pengujian *speech to text* ini dilakukan dengan menggunakan 1 *smartphone* yaitu *smartphone* A untuk menerima pesan. Pengujian ini dibagi menjadi 3 percobaan, antara lain *speech to text* untuk mengirim pesan menggunakan aplikasi *whatsapp*, *speech to text* untuk mengirim pesan menggunakan aplikasi *line* dan *speech to text* untuk membuka aplikasi. Setiap pengujian *speech to text*, untuk mengaktifkan fitur *speech to text*, *smartphone* harus di *shake* terlebih dahulu.

Pada pengujian *speech to text* untuk mengirim pesan menggunakan aplikasi *whatsapp*, *smartphone* A akan memberikan *input* yang berisi isi pesan, dan nama atau nomor tujuan. Sistem akan memproses data pesan yang di *input* *smartphone* A. Setelah mengimkan data pesan *smartphone* A, sistem akan mengarahkan halaman ke aplikasi *whatsapp* dan tujuan.

Pada pengujian *speech to text* untuk mengirim pesan menggunakan aplikasi *line*, *smartphone* A akan memberikan *input* suara yang berisi isi pesan, dan aplikasi *line*. Sistem akan memproses data pesan yang di *input* *smartphone* A. Kemudian sistem akan mengarahkan halaman ke aplikasi *line* dan *user* dapat memilih tujuan pengguna *line* yang ingin dikirimkan pesannya

Pada pengujian *speech to text* untuk membuka aplikasi, *smartphone* A akan memberikan *input* suara yang berisi nama aplikasi. Kemudian sistem akan memproses data yang di *input* *smartphone* A. Setelah selesai diproses, sistem akan mengarahkan halaman ke aplikasi *line* dan *user* dapat memilih

4.3 Sistem mengatasi beberapa pesan yang masuk dalam waktu yang bersamaan

Pengujian ini dilakukan untuk memastikan bahwa setiap pesan yang masuk dalam waktu yang bersamaan dapat diterima dan disuarakan dengan baik. Pada pengujian ini, digunakan 3 *smartphone* yang 2 diantaranya digunakan untuk mengirim pesan yaitu *smartphone* B melalui aplikasi *facebook* dan C melalui aplikasi *line* dan 1 lagi, *smartphone* A digunakan untuk menerima pesan tersebut. Model *smartphone* dari *smartphone* A adalah *Xiaomi Mi10T Pro*. Pada pengujian ini pada *smartphone* B dan C dilakukan pengiriman pesan secara bersamaan ke *smartphone* A. Pada hasil pengujian yang dilakukan, pesan dari *smartphone* B yang diterima pertama oleh *smartphone* A. Jadi, pesan dari *smartphone* B akan di suarakan terlebih dahulu hingga selesai lalu akan dilanjutkan oleh pesan dari *smartphone* C

4.4 Pengujian performa

Pengujian performa dilakukan dengan tiga cara, antara lain sistem dijalankan dalam beberapa jam untuk menguji penggunaan baterai, sistem dijalankan pada *smartphone* yang berbeda, sistem dijalankan selama 10 jam 47 menit untuk menguji apakah *service* tetap berjalan. Pada pengujian ini model *smartphone* A adalah *Xiaomi Mi10T Pro*

4.4.1 Sistem dijalankan dalam beberapa jam untuk menguji penggunaan baterai

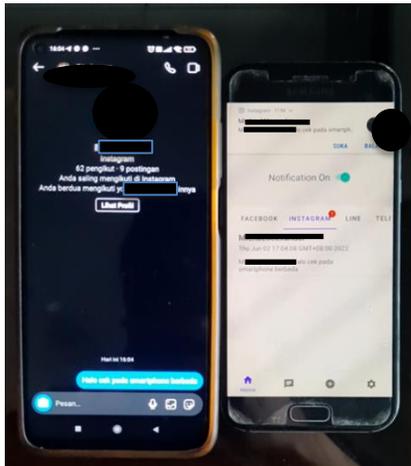
Pengujian ini dilakukan dengan menggunakan 1 *smartphone* A dimana *smartphone* A ini digunakan untuk menjalankan sistem selama 3 jam. Nama aplikasi dari sistem ini adalah “Shout it”. Pengujian ini dimulai pada tanggal 2 Juni 2022 jam 09:00 dan berakhir pada jam 12:00. Dari pengujian yang dilakukan selama 3 jam, sistem mengkonsumsi baterai sebesar 23,86 mAh. Kapasitas baterai dari *smartphone* A adalah sebesar 5000 mAh (Gambar 6)



Gambar 6. Konsumsi sistem terhadap baterai *smartphone*

4.4.2 Sistem dijalankan pada smartphone yang berbeda

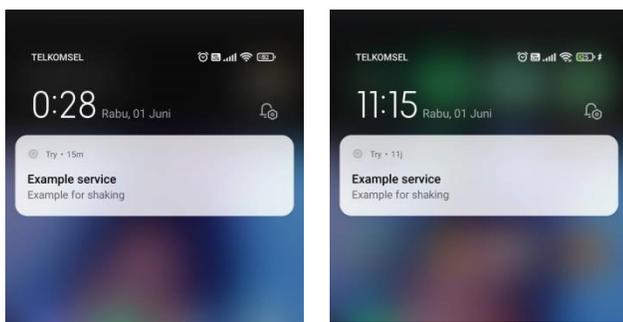
Pengujian ini dilakukan dengan menggunakan 2 *smartphone*, dimana *smartphone* A digunakan untuk mengirim pesan dan *smartphone* C digunakan untuk menerima pesan. Model *smartphone* C Samsung A5 (2017). *Smartphone* A mengirimkan pesan ke *smartphone* C dengan isi pesan “Halo cek pada *smartphone* berbeda”. Sistem pada *smartphone* C berhasil mendeteksi pesan dan menyuarakan pesan “Halo cek pada *smartphone* berbeda” (Gambar 7)



Gambar 7. Pengujian menerima pesan pada *smartphone* berbeda

4.4.3 Sistem dijalankan selama 10 jam 47 menit untuk menguji apakah service tetap berjalan

Pengujian ini dilakukan dengan menggunakan 1 *smartphone* A dimana *smartphone* A ini digunakan untuk menjalankan sistem selama 10 jam 47 menit. Pengujian ini dimulai pada tanggal 1 Juni 2022 jam 0:28 dan berakhir pada jam 11:15 (Gambar 8). Dari pengujian yang dilakukan, *service* tetap berjalan sehingga pesan-pesan yang masuk pun dapat diterima



Gambar 8. Sistem pada proses pengujian

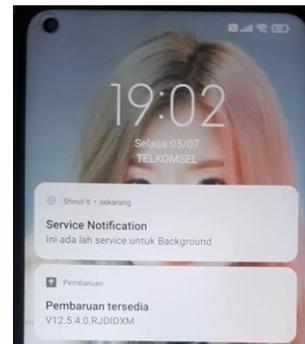
4.4.4 Sistem dapat aktif secara otomatis setelah *smartphone* di nyalakan

Pengujian ini dilakukan dengan menggunakan 1 *smartphone* A dimana *smartphone* A ini akan dilakukan restart. Setelah *smartphone* A selesai melakukan restart perangkat seperti pada Gambar 9, sistem akan secara otomatis aktif seperti pada

Gambar 10 dan dapat langsung menerima pesan yang dikirimkan ke *smartphone* A



Gambar 9. Sistem pada awal aktif



Gambar 10. Sistem setelah mendeteksi *Boot Up* dari *smartphone*

Berikut pada Tabel 6 merupakan perbandingan beberapa *smartphone* untuk menguji konsumsi baterai dan juga fitur utama dari sistem.

Tabel 6. Perbandingan beberapa *smartphone*

Smartphone	RAM	Versi Android	Konsumsi / Jumlah Baterai	Penerimaan pesan dan Penyuaran
Xiaomi Mi 10T Pro	8 GB	11	23,86 mAh / 5000 mAh	Berhasil
Samsung A5 (2017)	4 GB	8	12,05 mAh / 3000 mAh	Berhasil
Redmi Mi Note 5 Plus	4 GB	8	7,05 mAh / 4000 mAh	Berhasil

5. KESIMPULAN DAN SARAN

Berikut adalah kesimpulan dan saran dari hasil pengujian pada sistem.

5.1 Kesimpulan

Berdasarkan keseluruhan sistem yang telah dibuat dan hasil pengujian yang dilakukan, dapat disimpulkan bahwa.

1. Sistem penyuaran pesan teks menggunakan *text to speech* mampu menyuarakan pesan-pesan teks dari aplikasi media sosial. Sistem ini juga mampu mengatasi apabila ada 2 atau lebih pesan yang masuk secara bersamaan. Sistem dapat mengatasinya dengan menggunakan metode *queue* dalam proses penyuarannya. Sehingga pesan yang pertama masuk akan disuarakan terlebih dahulu hingga selesai dan akan dilanjutkan ke pesan selanjutnya

2. Sistem *speech to text* mampu digunakan untuk membantu sistem dalam mengirimkan pesan sesuai dengan tujuan dan isi pesan yang diinginkan dan juga membantu sistem untuk membuka aplikasi media sosial yang terdaftar.

3. Berdasarkan pengujian performa, keseluruhan sistem yang telah dibuat tidak menguras baterai banyak, dapat dijalankan pada *smartphone* yang berbeda, sistem *service* dapat berjalan dengan baik dalam penggunaan selama 10 jam 47 menit, dan *boot-up* service dapat berjalan dengan baik

4. Data pesan yang diterima dari obrolan grup pada aplikasi *line* memiliki permasalahan. Terdapat 2 cara pengiriman notifikasi yang dilakukan oleh aplikasi *line* untuk obrolan grup. Cara pertama, notifikasi yang akan dikirimkan oleh *line* terdiri dari 2 notifikasi, yang secara bersamaan dikirimkan ke sistem sehingga data pesan akan disuarakan sebanyak 2 kali. Data pesan pertama berisi nama grup, nama akun *line* dari pengirim dan isi pesan sedangkan data pesan kedua berisi nama akun *line* dari pengirim, dan isi pesan. Cara kedua, aplikasi *line* akan mengirimkan notifikasi ke sistem dengan data pesan yang berisi nama akun *line* dari pengirim dan isi pesan.

5. Data pesan yang diterima dari obrolan grup pada aplikasi *whatsapp* memiliki permasalahan. Untuk permasalahan pada aplikasi *whatsapp*, apabila pada obrolan grup ada pesan yang belum dibaca atau belum dibuka pada aplikasi *whatsapp*, dan ada pesan baru yang dikirimkan pada grup tersebut, nama pengirim dari data pesan tersebut yang terdeteksi adalah nama grup, jumlah pesan yang belum di baca, nama akun *whatsapp* dari pengirim.

5.2 Saran

Berikut merupakan saran-saran yang diberikan untuk meningkatkan kinerja dari sistem dan pengembangan untuk penelitian kedepannya

1. Dapat meningkatkan kinerja pengiriman pesan dengan mengimplementasi secara menyeluruh sistem *speech to text* pada sistem sehingga *user* tidak perlu berinteraksi dengan aplikasi selain menggunakan suara.

2. Dapat mengimplementasikan keseluruhan sistem pada sistem mobil

3. Dapat memperluas cakupan bahasa yang dapat digunakan

6. DAFTAR PUSTAKA

- [1] Ali Sarker, M. R., Hassanuzzaman, M., Biswas, P., Dadon, S. H., Imam, T., & Rahman, T. (2021). An efficient surface map creation and tracking using smartphone sensors and crowdsourcing. *Sensors*, 21(21), 6969. doi: 10.3390/s21216969
- [2] Barata, M., Galih Salman, A., Faahakhododo, I., & Kanigoro, B. (2018). *Android based voice assistant for blind people*. *Library Hi Tech News*, 35(6), 9–11. doi:10.1108/lhtn-11-2017-0083
- [3] Chen, Y. (2021). Research on Android Architecture and Application Development. *Journal of Physics: Conference Series*, 1992(2), 022168. doi:10.1088/1742-6596/1992/2/022168
- [4] Jovanovic, M., Babic, I., Cabarkapa, M., Mistic, J., Mijalkovic, S., Nikolic, V., & Randjelovic, D. (2018). SOSerbia: *Android-Based Software Platform for Sending Emergency Messages*. *Complexity*, 2018, 1–9. doi:10.1155/2018/828391
- [5] Kuligowska, K., Kisielewicz, P., & Włodarz, A. (2018). Speech synthesis systems: disadvantages and limitations. *International Journal of Engineering & Technology*, 7(2.28), 234. doi:10.14419/ijet.v7i2.28.12933
- [6] Sawano, R., & Murao, K. (2020). Annotation Method for Human Activity and Device State Recognition Based on Smartphone Notification Removals. *Journal of Information Processing*, 28(0), 679–688. doi: 10.2197/ipsjip.28.679
- [7] Sawant, N. K., & Borkar, S. (2018). Devanagari Printed Text to Speech Conversion using OCR. *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference On*. doi:10.1109/i-smac.2018.865368
- [8] Selvaraj, C., & Bhalaji, N. (2018). Enhanced portable text to speech converter for visually impaired. *International Journal of Intelligent Systems Technologies and Applications*, 17(1/2), 42. doi:10.1504/ijista.2018.10012881
- [9] Sun, X., Chen, X., Liu, K., Wen, S., Li, L., & Grundy, J. (2021). Characterizing Sensor Leaks in Android Apps. *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*. doi:10.1109/issre52982.2021.00058
- [10] Weber, D., Voit, A., & Henze, N. (2018). Notification Log. *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers - UbiComp '18*. doi:10.1145/3267305.327