

Pemanfaatan

## Google API Text-to-Speech

Dalam Otomatisasi Pembacaan Pesan Masuk  
Perangkat Android

Aplikasi *text-to-speech* sudah cukup banyak muncul di dunia. Namun, sebagian besar aplikasi tersebut menggunakan bahasa Inggris sebagai dasar utama. Sementara aplikasi *text-to-speech* yang dikhususkan untuk bahasa Indonesia masih kurang diminati untuk diteliti, padahal banyak hal yang bisa dipelajari dari aplikasi *text-to-speech*.

Teknologi *Text-to-speech* memungkinkan suatu aplikasi dapat mengubah deretan kata-kata sebagai *input* dan diproses menjadi ucapan sebagai *output*. *Text-to-speech* juga bisa mengkonversi teks pesan ke dalam bentuk ucapan sehingga isi pesan tertentu dapat didengarkan secara *real time*.

Buku ini khusus membahas pengembangan perangkat lunak *android* menggunakan *Google Cloud Text-to-Speech* untuk mengubah pesan teks yang masuk pada perangkat *android* menjadi suara sehingga memberikan kemudahan bagi pengguna yang mengalami gangguan penglihatan. Buku ini juga akan sangat bermanfaat bagi pembaca, khususnya khususnya pengembang aplikasi berbasis *android* baik dari sisi pengembangan aplikasi maupun pemanfaatan *Google Aps*.

**BINTANG  
SEMESTA MEDIA**

Jl. Karangasem, Gg. Nakhuda, Slemah, Yogyakarta 55772  
Telepon: 02741 4256388 WA: 0858 4634 2317  
Email: info@bintangsemestamedia.com  
Website: bintangsemestamedia.com



ISBN 978-625-5361-17-8



9 786235 361178

Fiqhi Maulana, dkk

Pemanfaatan Google API Text-to-Speech Dalam Otomatisasi Pembacaan Pesan Masuk Perangkat Android



Pemanfaatan



## Google API Text-to-Speech

Dalam Otomatisasi Pembacaan Pesan Masuk  
Perangkat Android

Fiqhi Maulana, dkk



**Pemanfaatan Google API  
Text-to-Speech  
Dalam Otomatisasi Pembacaan Pesan  
Masuk Perangkat Android**

**UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 28 TAHUN 2014**  
**TENTANG**  
**HAK CIPTA**  
**Lingkup Hak Cipta**

**Pasal 1 Ayat 1 :**

1. Hak Cipta adalah hak eksklusif pencipta yang timbul secara otomatis berdasarkan prinsip deklaratif setelah suatu ciptaan diwujudkan dalam bentuk nyata tanpa mengurangi pembatasan sesuai dengan ketentuan peraturan perundang-undangan.

**Ketentuan Pidana:**

**Pasal 113**

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

**Pasal 114**

Setiap Orang yang mengelola tempat perdagangan dalam segala bentuknya yang dengan sengaja dan mengetahui membiarkan penjualan dan/atau pengandaan barang hasil pelanggaran Hak Cipta dan/atau Hak Terkait di tempat perdagangan yang dikelolanya sebagaimana dimaksud dalam Pasal 10, dipidana dengan pidana denda paling banyak Rp100.000.000,00 (seratus juta rupiah).

Fiqhi Maulana  
Najirah Umar  
M. Adnan Nur

**Pemanfaatan Google API  
Text-to-Speech  
Dalam Otomatisasi Pembacaan Pesan  
Masuk Perangkat Android**

Diterbitkan Oleh



# **Pemanfaatan Google API Text-to-Speech Dalam Otomatisasi Pembacaan Pesan Masuk Perangkat Android**

Penulis : Fiqhi Maulana  
Najirah Umar  
M. Adnan Nur  
Tata Letak : Riza Ardyanto  
Desain Cover : Ridwan Nur M

## **Penerbit:**

### **CV. Bintang Semesta Media**

Anggota IKAPI Nomor 147/DIY/2021

Jl. Karang Sari, Gang Nakula, RT 005, RW 031,

Sendangtirto, Berbah, Sleman, Yogyakarta 57773

Telp: 4358369. Hp: 085865342317

Facebook: Penerbit Bintang Madani

Instagram: @bintangpustaka

Website: [www.bintangpustaka.com](http://www.bintangpustaka.com)

Email: [bintangsemestamedia@gmail.com](mailto:bintangsemestamedia@gmail.com)

[redaksibintangpustaka@gmail.com](mailto:redaksibintangpustaka@gmail.com)

Cetakan Pertama, Maret 2022

Bintang Semesta Media Yogyakarta

xii + 128 hal : 14.5 x 20.5 cm

ISBN : 978-623-5361-17-8

Dicetak Oleh:

Percetakan Bintang 085865342319

Hak cipta dilindungi undang-undang

*All right reserved*

Isi di luar tanggung jawab percetakan

## Prakata

Puji dan Syukur penulis panjatkan kehadirat Allah SWT atas berkah dan karunia-Nya sehingga penulis dapat menyelesaikan buku dengan judul “Pemanfaatan *Google Text-To-Speech* Dalam Otomatisasi Pembacaan Pesan Masuk Perangkat *Android*.” Buku ini membahas pengembangan perangkat lunak *android* yang menggunakan *Google Cloud Text-to-Speech* untuk mengubah pesan teks yang masuk pada perangkat *android* menjadi suara sehingga memberikan kemudahan bagi pengguna yang mengalami gangguan penglihatan.

Pengembangan perangkat lunak yang disajikan pada buku ini meliputi pemodelan dan perancangan aplikasi dengan *flowchart* dan diagram *use case*, perancangan dan pembuatan basis data pada *SQLite*, perancangan antarmuka, implementasi dan pembuatan aplikasi pada *Android Studio* serta pengujian pada beberapa perangkat dan versi *android*. Landasan dan kebutuhan dalam pengembangan aplikasi *android text-to-speech* juga diberikan untuk menggambarkan secara lengkap kepada pembaca terkait *resource* dan *tools* yang digunakan.

Pemanfaatan *Google Cloud Text-to-Speech* pada pengembangan aplikasi android yang dibahas dalam buku ini diharapkan dapat memberikan pengetahuan tambahan bagi pembaca khususnya pengembang aplikasi berbasis android baik dari sisi pengembangan aplikasi maupun pemanfaatan Google Apis. Penulis menyadari masih banyak kekurangan pada buku ini sehingga untuk penyempurnaan, penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Makassar, Maret 2022

Penulis

## Daftar isi

Prakata .....	v
Daftar isi .....	vii
Daftar tabel.....	ix
Daftar gambar.....	x
Daftar lampiran .....	xii

### BAGIAN PERTAMA

Pengantar Aplikasi Pembaca Pesan .....	1
A. Aplikasi Text-to-Speech Berbahasa Indonesia	
Sepi Peminat.....	1
B. Gambaran Aplikasi .....	6
C. Kebutuhan Aplikasi .....	8

### BAGIAN KEDUA

Landasan Pengembangan Aplikasi Pembaca Pesan.....	11
A. Kaidah Bahasa Indonesia .....	11
B. Teknologi Pemrosesan Bahasa.....	12
C. Text-to-Speech.....	13
D. Google API.....	15
E. Implementasi <i>Google Text-to-Speech API</i> .....	16



F. Android.....	21
G. Flowchart .....	29
H. Unified Modeling Language (UML).....	30
I. SQLite .....	40

### BAGIAN KETIGA

Penerapan Aplikasi Pembaca Pesan.....	41
A. Permodelan Aplikasi yang Diusulkan .....	41
B. Perancangan Fitur.....	50
C. Perancangan <i>Database</i> .....	53
D. Perancangan Antar Muka .....	55
E. Implementasi Antarmuka .....	61
F. Pengujian Aplikasi.....	69

### BAGIAN KEEMPAT

Kesimpulan .....	77
Daftar Pustaka .....	79
Lampiran .....	82
Panduan Penggunaan Aplikasi Pembaca Pesan .....	103
Biodata Penulis.....	127

## DAFTAR TABEL

Tabel 2. 1	
<i>Use Case Diagram</i> .....	32
Tabel 2. 2	
<i>Class Diagram</i> .....	35
Tabel 2. 3	
<i>Sequence Diagram</i> .....	37
Tabel 2. 4	
<i>Activity Diagram</i> .....	38
Tabel 4.1	
Tabel <i>Chat</i> .....	54
Tabel 4. 3	
Tabel Rencana Pengujian Fitur .....	70
Tabel 4. 4	
Tabel Hasil Pengujian Fitur .....	71

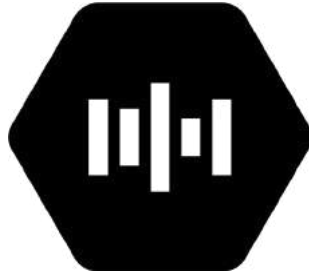
## DAFTAR GAMBAR

Gambar 1.	
<i>Use Case</i> Aplikasi yang diusulkan .....	6
Gambar 3.	
Diagram Blok Aplikasi <i>Text-To-Speech</i> Klien/Server.....	19
Gambar 4. 1	
<i>Flowchart</i> Aplikasi yang diusulkan.....	42
Gambar 4. 2	
Inisialisasi <i>Notification Listener Service</i> .....	43
Gambar 4.3	
Inisialisasi TTS.....	44
Gambar 4. 4	
Objek ke string.....	45
Gambar 4. 5	
<i>Flowchart</i> Parsing Kalimat.....	47
Gambar 4. 7	
<i>Fragment</i> dalam <i>activity</i> .....	50
Gambar 4. 8	
<i>Flowchart</i> Input Singkatan .....	51
Gambar 4. 9	
<i>Option Menu</i> .....	52

Gambar 4. 10	
Rancangan <i>Welcome Activity</i> .....	56
Gambar 4. 11	
Rancangan Halaman Utama.....	57
Gambar 4. 12	
Rancangan <i>Context Menu</i> .....	58
Gambar 4. 13	
Rancangan Tambah Kata Singkatan .....	59
Gambar 4. 14	
Rancangan Hapus Pesan Telegram.....	60
Gambar 4. 15	
Rancangan Hapus Pesan WhatsApp .....	61
Gambar 4. 16	
Ikon Aplikasi Pembaca Pesan .....	61
Gambar 4. 17	
Tampilan <i>Welcome Activity</i> .....	62
Gambar 4. 18	
Tampilan Halaman Utama .....	63
Gambar 4. 19	
Tampilan <i>Action Context Menu</i> .....	64
Gambar 4. 20	
Tampilan Tambah Kata Singkatan .....	65
Gambar 4. 21	
Tampilan Hapus Pesan Telegram.....	66
Gambar 4. 22	
Tampilan Hapus Pesan WhatsApp .....	67
Gambar 4. 23	
Tampilan Hasil Penerimaan Pesan.....	68

## DAFTAR LAMPIRAN

Tabel Lampiran 1	
Tabel Hasil Membaca Pesan di Xiaomi MI A2 .....	83
Tabel Lampiran 2	
Tabel Hasil Membaca Pesan di Samsung A7 (A750) .....	87
Tabel Lampiran 3	
Tabel Hasil Membaca Pesan di OPPO A7 .....	90
Tabel Lampiran 4	
Tabel Hasil Membaca Pesan di Realme 6 Pro .....	94
Tabel Lampiran 5	
Tabel Hasil Membaca Pesan di Nokia 3 .....	97
Tabel Lampiran 6	
Pengujian berdasarkan nama dan gelar .....	100
Tabel Lampiran 7	
Pengujian berdasarkan singkatan huruf kapital .....	101
Tabel Lampiran 8	
Pengujian berdasarkan akronim .....	102
Source Code .....	109



## **BAGIAN PERTAMA**

### **Pengantar Aplikasi Pembaca Pesan**

#### **A. Aplikasi Text-to-Speech Berbahasa Indonesia Sepi Peminat**

Penggunaan ponsel pintar (*smartphone*) di Indonesia diperkirakan akan terus bertambah. Pada tahun 2015, penggunaan ponsel pintar sebanyak 28,6% dari populasi penduduk di Indonesia. Seiring dengan berjalannya waktu, *smartphone* semakin terjangkau sehingga penggunaannya semakin meningkat. Angka yang berlaku pada setiap orang yang mengakses internet setidaknya satu kali setiap bulan mendudukkan Indonesia di peringkat ke-6 terbesar di dunia dalam hal jumlah penggunaan internet. 56,2% dari populasi penduduk di Indonesia telah menggunakan *smartphone* pada tahun 2018. Pada tahun 2019, sebanyak 63,3% dari populasi penduduk di Indonesia menggunakan *smartphone*. Hingga pada tahun 2025, diperkirakan 89,2% populasi penduduk di Indonesia, dengan kata lain bertambah 25,9% dari data yang ada akan menggunakan *smartphone* (databoks.katadata, 2020).

Menurut Marius dan Sapto, ada tiga alasan orang Indonesia menggunakan internet, di antaranya adalah untuk mengakses media sosial atau komunikasi sebanyak 72%, memperoleh informasi harian sebanyak 65%, dan mengikuti tren perkembangan zaman sebanyak 51%. Tiga alasan mengakses internet tersebut diimplementasikan melalui empat kegiatan pokok, yaitu menggunakan jejaring sosial sebanyak 87%, sumber informasi sebanyak 69%, berkirim pesan instan sebanyak 60% dan mencari berita terbaru sebanyak 60% (Marius, Sapto, 2014).

Dalam artikel yang dibahas oleh *Scott Jeffrey* ada dua *instant messaging* yang memiliki kelebihan yang berbeda menjadi daya tarik bagi pengguna ponsel, yaitu Telegram dengan keunggulan lebih ringan dan berbasis *cloud* yang mempunyai proses sinkronisasi yang mulus dan WhatsApp dengan kelebihan tampilan yang mudah dengan aplikasi enkripsi dan menjadi *instant messaging* terpopuler (Jeffrey, 2014).

Berkomunikasi menggunakan *instant messaging* biasanya pengguna mengalami beberapa keterbatasan dalam membaca pesan yang diterima sehingga menghambat informasi dan tidak tersampaikan dengan cepat. Beberapa keterbatasan tersebut seperti jarak ponsel dengan pengguna, penyandang tunanetra, berkendara, dan keterbatasan aktivitas lainnya. Dengan teknologi *Text-To-Speech* (TTS) keterbatasan tersebut menjadi solusi yang baik untuk mempermudah pengguna.

*Text-To-Speech* adalah suatu aplikasi yang memproses dua konversi, yaitu konversi teks ke *phonem* dan konversi *phonem* ke ucapan melalui tahap secara berurutan dengan masukan dalam

bentuk teks dan menghasilkan keluaran dalam bentuk ucapan. Namun, dalam mengkonversi teks menjadi suara dibutuhkan fonem dan prosodi untuk menghasilkan pengucapan yang dapat dimengerti sehingga membutuhkan penyimpanan yang sangat besar untuk data fonem dan prosodi dalam bahasa Indonesia.

Aplikasi *text-to-speech* memang sudah cukup banyak muncul di dunia. Namun, sebagian besar aplikasi-aplikasi tersebut menggunakan bahasa Inggris sebagai dasar utama. Sementara itu, aplikasi *text-to-speech* yang dikhususkan untuk bahasa Indonesia masih kurang diminati untuk diteliti. Hal tersebut tentunya membuat perkembangan aplikasi *text-to-speech* berbahasa Indonesia belum banyak beredar luas, padahal banyak hal yang bisa dipelajari dari aplikasi *text-to-speech*.

Teknologi *Text-to-speech* memungkinkan suatu aplikasi dapat mengubah deretan kata-kata sebagai *input* dan diproses menjadi ucapan sebagai *output*. *Text-to-speech* juga bisa mengkonversi teks pesan ke dalam bentuk ucapan sehingga isi pesan tertentu dapat didengarkan secara *real time*.

Namun, dari aplikasi tersebut masih terdapat penulisan teks pesan yang belum bisa dibaca dengan jelas. Penyingkatan kata dalam penulisan pesan tersebut belum dapat dikenali oleh aplikasi *text-to-speech* dengan baik sehingga aplikasi mengeluarkan ucapan yang kurang jelas dan bermakna ambigu yang menimbulkan terjadinya kesalahpahaman pada penerima pesan.

Langkah awal dalam membentuk sintesis ucapan adalah dengan merekam berbagai macam kata dalam Bahasa Indonesia



dan disimpan pada *database*. Kemudian, hasil rekaman tersebut disegmentasi menjadi unit-unit *diphone*. Perangkaian unit-unit *diphone* dilakukan dengan algoritma *Time Domain Pitch Synchronous Overlap-Add (TD PSOLA)*.

Algoritma *TD-PSOLA* memperkenankan menggunakan suara rekaman untuk dirangkai dengan lancar dan halus, juga memberikan kemudahan dalam mengendalikan *pitch* dan durasi. Dengan menggunakan algoritma tersebut, perangkaian antar unit-unit *diphone* yang memiliki transisi antar dua bunyi berdekatan diharapkan akan lebih halus dan dapat mengurangi bunyi yang mempunyai sifat letupan sehingga sintesis ucapan yang dihasilkan mampu memenuhi tiga kriteria, yaitu terdengar jelas dan dapat dimengerti, lancar dan alami (*intelligibility, fluidity, naturalness*).

*Google Text-to-Speech API* memiliki tingkat keakuratan sintesis ucapan tinggi dan memungkinkan *user* dapat menyintesis ucapan sehingga terdengar lebih alami, serta menawarkan pilihan sebanyak 30 suara yang tentunya tersedia dalam berbagai bahasa yang mudah diintegrasikan dengan aplikasi dan perangkat yang ada.

Sebagai *API (Application Programming Interface)* yang gampang digunakan, *Google Text-to-Speech* juga mendukung aplikasi serta perangkat jenis apa pun yang mampu melakukan transmisi *REST* atau *gRPC*, di antaranya *smartphone*, komputer, tablet, dan *IoT*. Beberapa pemanfaatan *Google Text-to-Speech* antara lain mekanisasi secara otomatis pusat panggilan dan umpan balik interaktif dari jenis perangkat *IoT*.

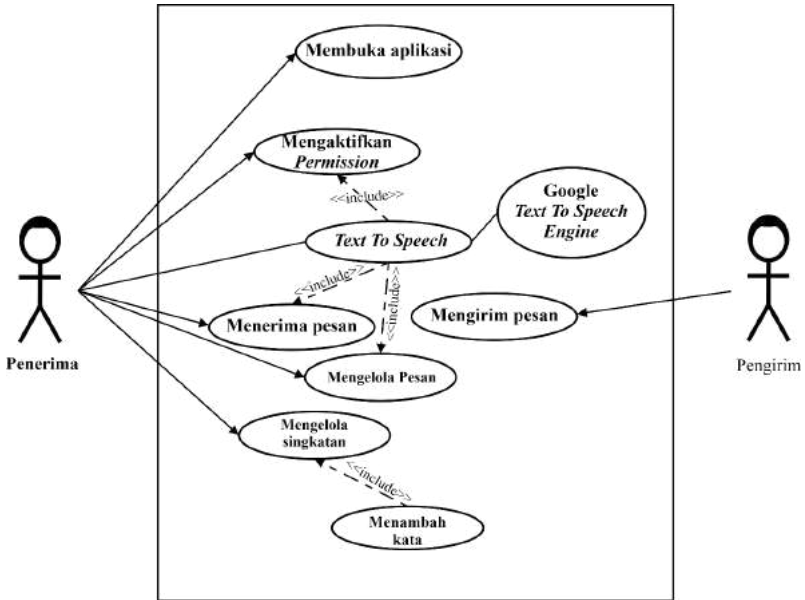
Pada dasarnya, aplikasi *text-to-speech* dari Google ini adalah solusi yang serba guna untuk menghasilkan suatu *natural experience* untuk beberapa jenis kasus pemanfaatan. Fitur yang canggih dari Google tersebut mampu meringankan pekerjaan, terutama pada saat membutuhkan fitur yang mampu memproses perekaman suara. Karena adanya aplikasi *text-to-speech* tersebut, maka *user* dapat mempunyai *device* yang mampu berinteraksi kembali kepada *user*. Fitur teknologi canggih pada *text-to-speech* ini bahkan mampu digunakan untuk memfasilitasi penyandang tunanetra agar dapat mendengarkan dan memahami suatu pesan teks.

Berdasarkan uraian tersebut, penulis mengusulkan aplikasi pembaca pesan yang akan menerima teks pesan dari Telegram dan WhatsApp kemudian mengubahnya menjadi ucapan menggunakan *Google API text-to-speech voice* berbahasa Indonesia. Aplikasi *Text-To-Speech* Bahasa Indonesia ini berbasis android yang mampu mengucapkan kata singkatan dengan jelas.

Aplikasi tersebut juga melakukan interaksi pada *user* dalam proses penyempurnaan *database* kata singkatan sehingga aplikasi dapat mengenali kata singkatan lebih banyak. Selain itu, aplikasi ini dilengkapi dengan fitur manajemen pesan seperti menghapus dan mendengarkan kembali pesan yang telah diterima serta memasukkan singkatan untuk menghasilkan keluaran yang baik.

## B. Gambaran Aplikasi

Aplikasi pembaca pesan masuk pada perangkat *android* dapat digambarkan dalam diagram *use case* berikut:



**Gambar 1.** *Use Case* Aplikasi yang diusulkan

Berikut adalah aktor-aktor yang terdapat pada aplikasi.

a. Penerima

Merupakan aktor yang berperan menerima pesan.

b. Pengirim

Aktor yang berperan untuk mengirim pesan.

*Use case* adalah suatu fungsi yang disiapkan pada aplikasi sebagai unit-unit yang saling bertransmisi terhadap unit lain atau aktor. *Use case* yang terdapat pada aplikasi adalah sebagai berikut.

- a. Mengirim Pesan  
Fungsionalitas untuk mengirim pesan.
- b. Menerima Pesan  
Fungsionalitas untuk menerima pesan.
- c. Membuka Aplikasi  
Fungsionalitas untuk membuka aplikasi Pembaca Pesan
- d. Mengaktifkan Permission  
Fungsionalitas untuk mengaktifkan izin aplikasi agar pesan baru dapat dibaca secara otomatis
- e. *Text to Speech*  
Fungsionalitas untuk mengkonversi pesan teks ke dalam bentuk ucapan (suara) yang akan dituturkan oleh Google *text-to-speech engine*.
- f. Mengelola pesan  
Fungsionalitas untuk mengelola pesan masuk yang tersimpan untuk dibacakan kembali atau ingin dihapus.
- g. Mengelola singkatan  
Fungsionalitas untuk mengelola data singkatan yang tersimpan di *database*
- h. Menambah kata  
Fungsionalitas untuk menambah kata singkatan pada *database* singkatan

## C. Kebutuhan Aplikasi

### 1. Kebutuhan *Functional*

Kebutuhan fungsional merupakan kebutuhan yang berkaitan secara langsung terhadap aplikasi. Berikut adalah kebutuhan fungsional yang diperlukan pada aplikasi yang ingin dibangun.

- a. *User* bisa dengan gampang memahami *message* atau informasi walaupun tanpa intensitas konsentrasi yang tinggi.
- b. *Message* atau informasi bisa diterima ketika *user* tengah melakukan aktivitas lainnya, contohnya saat berjalan-jalan, menangani atau sedang memperhatikan objek lainnya

### 2. Kebutuhan *Non-Functional*

#### a. *Software*

*Software* atau perangkat lunak yang dibutuhkan dalam mengimplementasikan aplikasi yang dibuat antara lain:

- 1) Android Studio versi 3.5.2,
- 2) Android SDK (*Software Development Kit Manager*),
- 3) Java (*jdk-8u121-windows-x64*),
- 4) SQLite *Manager*.

b. *Hardware*

*Hardware* yang digunakan dalam mengimplementasikan aplikasi antara lain:

1) *Smartphone* Xiaomi MI A2

Spesifikasi *smartphone* yang digunakan antara lain:

- a) Android versi 10.0, Android Pure,
- b) RAM 4 GB, *Internal* 64 GB,
- c) *Processor Qualcomm* SDM660 Snapdragon 660.

2) *Smartphone* Samsung A7 (A750)

Spesifikasi *smartphone* yang digunakan antara lain:

- a) Android versi 10.0, One UI 2.0,
- b) RAM 4 GB, *Internal* 64 GB,
- c) *Processor Exynos* 7885

3) *Smartphone* OPPO A7

Spesifikasi *smartphone* yang digunakan antara lain:

- a) Android versi 8.1, ColorOS 5.2,
- b) RAM 4 GB, *Internal* 64 GB,
- c) *Processor Mediatek* MT6771 Helio P60.

4) *Smartphone* Realme 6 Pro

Spesifikasi *smartphone* yang digunakan antara lain:

- a) Android versi 10.0, Realme UI,
- b) RAM 6 GB, *Internal* 64 GB,

c) *Processor Qualcomm SM7125 Snapdragon 720G.*

5) *Smartphone Nokia 3*

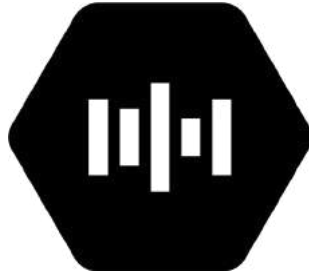
Spesifikasi *smartphone* yang digunakan antara lain:

- a) *Android versi 8.0,*
- b) *RAM 2 GB, Internal 16 GB,*
- c) *Processor Mediatek MT6737.*

6) *Desktop*

Spesifikasi *desktop* yang digunakan antara lain:

- a) *Laptop DELL Inspiron,*
- b) *Processor Intel Core i3-5005U @ 2.0GHz,*
- c) *RAM 8 GB,*
- d) *SSD 256 GB, Harddisk 500 GB,*



## **BAGIAN KEDUA**

### **Landasan Pengembangan Aplikasi Pembaca Pesan**

#### **A. Kaidah Bahasa Indonesia**

Bahasa Indonesia terdiri dari bahasa tulisan dan bahasa lisan. Terkadang ada beberapa perbedaan pada kedua jenis bahasa ini. Pada bahasa lisan, terdapat istilah *phonem*, yaitu suatu kesatuan bahasa terkecil yang dapat membedakan arti. Pada bahasa tulisan, *phonem* disimbolkan dengan huruf. Maka huruf merupakan tulisan dari *phonem*. Sering kali istilah *phonem* disamaartikan dengan huruf, padahal tidak selalu berlaku demikian. Berikut adalah konsep bahasa Indonesia berdasarkan Pedoman Umum Ejaan Bahasa Indonesia yang disempurnakan.

##### **1. Abjad**

Abjad yang ada dalam bahasa Indonesia terdiri atas 52 huruf, yaitu 26 huruf besar (A-Z) dan 26 huruf kecil (a-z).



## 2. *Phonem*

*Phonem* merupakan istilah linguistik yang berarti satuan terkecil dalam sebuah bahasa yang masih bisa menunjukkan perbedaan makna.

## 3. *Diphone*

*Diphone* adalah gabungan dari dua buah fonem bahasa Indonesia (Arman, 2004).

### **B. Teknologi Pemrosesan Bahasa**

Bahasa terdiri atas dua bagian, yaitu bahasa alami dan bahasa buatan. Bahasa alami merupakan bahasa yang biasa digunakan untuk berkomunikasi antar manusia, seperti bahasa Indonesia, bahasa daerah, dan sebagainya. Bahasa buatan merupakan bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu, misalnya bahasa pemrograman komputer.

Suatu aplikasi pemrosesan bahasa alami secara lisan dapat dibentuk dari tiga sub-aplikasi (Pramono, 2013):

1. Sub-Aplikasi *Natural Language Processing* (NLP), mempunyai fungsi dapat melakukan pemrosesan secara simbolik terhadap bahasa tulisan. Beberapa bentuk aplikasi sub-aplikasi ini adalah penerjemah bahasa alami, aplikasi pemeriksaan sintak bahasa, aplikasi yang dapat menyimpulkan suatu narasi dan sebagainya.
2. Sub-Aplikasi *Text-to-Speech* (TTS), mempunyai fungsi dapat mengkonversi teks (tulisan) menjadi ucapan (bahasa lisan).

3. Sub-Aplikasi *Speech Recognition* (SR), merupakan aplikasi yang dapat mengubah ataupun mengenali suatu ucapan (bahasa lisan) menjadi teks (bahasa tulisan).

### C. Text-to-Speech

*Text-to-speech* adalah suatu aplikasi yang dapat mengubah deretan kata-kata sebagai *input* menjadi ucapan sebagai *output* secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan). Tujuan dari penggunaan *text-to-speech* (TTS) adalah untuk mengkonversi masukan berbentuk teks menjadi ucapan agar dapat dipahami oleh pengguna. Maka dari itu, *text-to-speech* mengirimkan informasi dari mesin kepada manusia. Aplikasi *text-to-speech* sering kali disebut sebagai aplikasi pensintesis ucapan. Aplikasi ini bisa dimanfaatkan untuk berbagai macam aplikasi (Wundari, 2009).

Membaca merupakan suatu proses men-*decode*-kan sinyal tulisan menjadi pesan agar dapat memahami pesan tersebut supaya didapatkan suatu pengertian (pemahaman). Membaca dengan bersuara (*reading aloud*) adalah suatu tindakan mengubah sinyal tulisan menjadi sinyal berbentuk ucapan (*speech*).

Dalam tulisan ada satu hal yang paling penting yang diabaikan, yaitu prosodi. Prosodi berguna untuk membentuk emosi dari pembicara. Pada bahasa tulisan, prosodi ini mempunyai komponen yang terbatas karena tidak adanya cara agar dapat mengekspresikan sejumlah fitur pada prosodi tersebut. Memang bisa diberikan tanda baca seperti tanda

seru (!), tanda tanya (?) dan semacamnya agar dapat memberi penekanan, tetapi hal-hal lain, misal kemarahan, kesakitan, ataupun sarkasme, tidak mempunyai metode khusus untuk mengutarakannya.

Dalam tulisan termaksud bahasa yang bukan suatu bagian dari linguistik, seperti bahasa matematika. Saat menemui suatu bahasa matematika, mesin dituntut dapat memberikan perbedaan dengan bahasa natural (*natural language*).

Kendala tersebut dikategorikan masalah semiotik. Bahasa natural diistilahkan seperti aplikasi semiotik. Mesin dituntut menginterpretasikan arti yang memiliki lambang semiotik saat menemukan kalimat yang mengandung lambang tersebut. Supaya mesin bisa memproses tulisan tersebut, mesin dituntut bisa mengenali bagian mana pada sebuah teks yang memiliki keterkaitan dengan masalah semiotik.

Kendala-kendala yang diungkap tersebut biasa dijumpai pada proses membaca sehingga dapat diidentifikasi model dari *text-to-speech* yang mampu melakukan proses yang diinginkan (Wundari, 2009).

Metode sintesis ucapan memungkinkan mesin bisa mengabaikan perintah atau informasi pada *user* melalui "ucapan". Proses tersebut melibatkan pemecahan kata menjadi *phonem*, menganalisis agar penanganan khusus pada *text* seperti angka, jumlah mata uang, perubahan *pitch* suara, dan juga pemberian tanda-tanda baca (Arman, 2004).

*Message* atau informasi yang dikirimkan melalui ucapan mempunyai kelebihan, di antaranya (Dutoit, 1997):

1. *User* bisa dengan gampang memahami *message* atau informasi walaupun tanpa intensitas konsentrasi yang tinggi.
2. *Message* atau informasi bisa diterima ketika *user* tengah melakukan aktivitas lainnya, contohnya saat berjalan-jalan, menangani atau sedang memperhatikan objek lainnya.

#### D. Google API

*Google API* merupakan bagian dari *Framework* Google. Beberapa *API* (*Application Programming Interface*) yang tersedia di Google yang memiliki manfaat bagi *developer* aplikasi seperti *web*, *mobile*, ataupun *desktop* dapat menggunakan sejumlah fitur yang sudah disediakan Google, contohnya: *AdSense*, *Search Engine*, *Translation*, *YouTube* dan lain sebagainya.

Secara sederhana *API* dapat dimaksudkan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi ataupun *web* yang dikembangkan dengan fungsi-fungsi yang dikerjakan. Contohnya dalam hal ini *Google API* merupakan kode program yang bisa ditambahkan ke aplikasi untuk memanfaatkan fungsi atau *feature* yang telah disiapkan oleh Google. Contohnya *developer* bisa menambahkan fitur *Google Map* pada aplikasi yang sedang dibangun. *Google API* bisa dipelajari secara langsung lewat *Google Code*. Dengan menggunakan *Google Code*, *developer* bisa belajar mengenai *Google API* dan bisa mengimplementasikan ke aplikasi yang sedang dibangun. Beberapa *API* yang disiapkan oleh Google antara lain (Irawan, 2014):

1. *Language API*: menggunakan fitur *translation* milik Google.
2. *Earth API*: menggunakan fitur yang terdapat pada *Google Earth*.
3. *Maps API*: menggunakan fitur yang terdapat pada *Google Maps*.
4. *Search API*: menggunakan fitur pencarian milik Google *Search*.
5. *Visualization API*: membuat grafik maupun diagram dengan *Google API*.
6. *YouTube API*: menggunakan fitur yang terdapat pada YouTube misalnya untuk pencarian video.
7. *Text to Speech API*: mengubah *text* ke bentuk ucapan.

#### E. Implementasi *Google Text-to-Speech API*

Pemanfaatan *Google Text-to-Speech API* memungkinkan pengembang agar dapat mengkonversi teks ke dalam ucapan (suara). Untuk memanfaatkan fitur *API* ini, pengembang android bisa memakai *interface* dan *class* yang telah disediakan *Google API* (*package android.speech.tts*) (Irawan, 2014).

##### 1. *Interface*

Terdapat beberapa *interface* yang terdapat pada *package android.speech.tts*, di antaranya:

- a. *SynthesisCallback*
- b. *TextToSpeech.OnInitListener*
- c. *TextToSpeech.OnUtteranceCompletedListener*

## 2. *Classes*

Terdapat beberapa *class* yang terdapat pada *package android.speech.tts*, di antaranya:

- a. *SynthesisRequest*, berisi data yang dibutuhkan oleh mesin untuk mensintesis ucapan.
- b. *TextToSpeech*, mensintesis ucapan dari teks untuk diputar langsung atau untuk membuat *file* suara,
- c. *TextToSpeech.Engine*, konstanta dan nama parameter untuk mengendalikan *text-to-speech*,
- d. *TextToSpeech.EngineInfo*, informasi tentang terpasang nya mesin *text-to-speech*,
- e. *TextToSpeechService*, kelas abstrak dasar untuk implementasi mesin *Text-to-Speech*.

Untuk dapat memanfaatkan fitur Google *Text-to-Speech API* pada program diperlukan beberapa langkah, di antaranya mengimpor *class* yang berhubungan dengan *text-to-speech*, yaitu

```
android.speech.tts.TextToSpeech;
```

Untuk dapat memanfaatkan kelas *text-to-speech*, diperlukan inisialisasi sebuah objek dari *class text-to-speech* dan menentukan *initListenere*.

```

Private EditText write;

tts = newTextToSpeech(getApplicationContext(),
    new TextToSpeech.OnInitListener() {
        @Override
        public void onInit(int status) {}
    }
);

```

Pada *listener* ini, harus menentukan properti untuk objek *text-to-speech*, seperti bahasa, *pitch* dan lain sebagainya. Bahasa dapat diatur dengan memanggil *method setLanguage()*, yaitu

```
tts.setLanguage(Locale.UK);
```

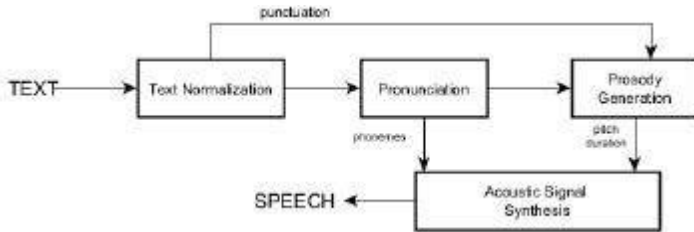
*Method setLanguage* mengambil objek lokal sebagai parameter. Berikut contoh daftar dari beberapa objek lokal (tergantung masing-masing *device*):

**US, CANADA\_FRENCH, GERMANY, ITALY, JAPAN, CHINA.**

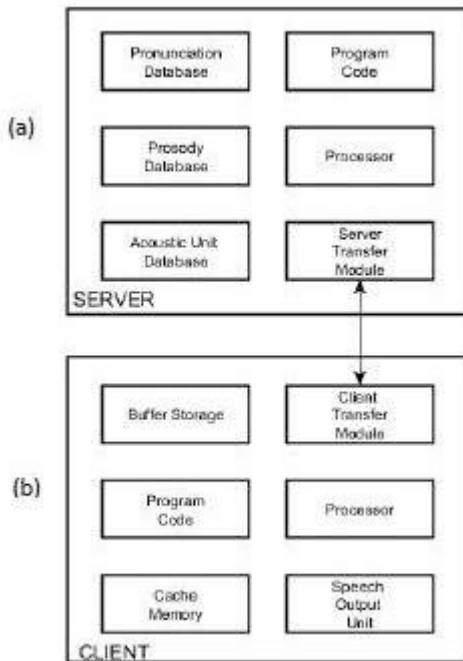
Setelah mengatur bahasa, dapat memanggil *method speak*

```
tts.speak(toSpeak, TextToSpeech.QUEUE_
    FLUSH, null);
```

Terlepas dari *method speak*, ada beberapa *method* lain yang disediakan pada *class text-to-speech* yaitu *addSpeech (String text, String filename)*, *getLanguage()*, *isSpeaking()*, *setPitch (floatpitch)*, *setSpeechRate (float speechRate)*, *shutdown()* dan *stop()*.



Gambar 2. Diagram Blok Aplikasi Sintetis *Text-To-Speech*



Gambar 3. Diagram Blok Aplikasi *Text-To-Speech* Klien/Server

Pada Gambar 2 dapat dilihat teks dikonversi ke ucapan melalui sejumlah unit pemrosesan. Teks masukan biasanya berisi tanda baca, singkatan, akronim dan simbol tanpa kata.



Pada unit normalisasi teks masukan semua teks kalimat akan diubah menjadi teks yang hanya berisi urutan kata-kata tanpa disingkat. Sebagian besar tanda baca berguna untuk menunjukkan keakuratan yang sesuai sehingga teks normalisasi tanda baca digunakan sebagai masukan pada unit *prosody*. Singkatan dan akronim diubah menjadi urutan kata yang mungkin atau tidak bergantung pada teks.

Fungsi utama dari normalisasi teks adalah mengkonversi simbol ke kata. Misalnya, angka, jumlah mata uang, tanggal, waktu, dan alamat *email* terdeteksi, diklasifikasikan dan kemudian dikonversi menjadi teks yang bergantung pada posisi simbol dalam kalimat. Teks dari normalisasi dikirim pada unit *pronunciation* sebagai awal analisis setiap kata untuk menentukan representasi morfologisnya. Kata-kata yang dihasilkan kemudian diubah menjadi urutan fonem. Terdapat tiga aturan yang digunakan oleh unit *pronunciation* untuk melakukan konversi di antaranya *letter-to-sound*, representasi statistik yang mengubah urutan huruf ke urutan fonem berdasarkan bahasa yang ada dan kamus. Kumpulan aturan ini akan membentuk *database* yang besar.

Fonem yang dihasilkan akan dikirim ke unit *prosody* bersama dengan tanda baca yang diambil dari unit *normalization*. Unit *prosody* akan menghasilkan informasi *timing* dan *pitch* yang dibutuhkan untuk *speech synthesis* dari struktur kalimat, tanda baca, kata-kata tertentu dan beberapa kalimat. Tanggal, waktu dan mata uang adalah contoh bagian kalimat yang didefinisikan khusus, nada masing-masing ditentukan dari

seperangkat aturan model statistik yang dibuat untuk jenis informasi tersebut. Ritme, atau durasi fonem, tanggal dan nomor telepon biasanya berbeda satu sama lain. Kumpulan aturan atau model statistik ini juga membutuhkan *database* cukup besar

*Unit acoustic signal synthesis*, menggabungkan *pitch*, durasi dan fonem dari unit *pronunciation* dan unit *prosody* untuk menghasilkan *signal* akustik yang baik. Ada beberapa metode dalam *speech synthesize*, yang pertama adalah *formant synthesize*, merupakan pemodelan suara manusia dan fonem *synthesize*, untuk menghasilkan kata yang diperlukan teknik *synthesize* berkualitas tinggi yang digunakan secara luas adalah *synthesize* gabungan, di mana suara seorang artis direkam untuk menghasilkan *database*, menentukan urutan unit yang digunakan dan menggabungkannya sedemikian rupa sehingga durasi dan *pitch* dimodifikasi untuk mendapatkan *prosody* yang diinginkan (Hillsboro, P.V., Sunnyvale, T.F.M, 2004).

## F. Android

Android merupakan sebuah *operating aplikasi* untuk *device mobile* berbasis *linux* yang meliputi *operating aplikasi*, *middleware* dan *application*. Android menyiapkan *platform open source* untuk para *developer* agar dapat mengembangkan aplikasi mereka.

Sejumlah fitur yang dimiliki pada *device smartphone* berbasis Android (Salman, W., Puspita, B., 2017) di antaranya:

1. *Dalvik Virtual Machine*, yaitu *Java Runtime Environment* yang sudah melalui proses optimasi untuk *device*

dengan *size* memori yang relatif kecil. Fitur *machine* tersebut membuat aplikasi dapat *running* dengan lancar pada *device* berbasis android.

2. *Touch-screen* atau layar sentuh. Fitur ini terbilang fenomenal dikarenakan belum pernah diaplikasikan pada perangkat ponsel model lama. Dengan memanfaatkan fitur ini maka proses *navigation* menu menjadi lebih mudah dikarenakan *user* tinggal memilih dan menekan menu yang ingin dijalankan tanpa perlu harus melakukan *scroll* ke atas-bawah atau samping kiri-kanan.
3. *Multipage*. Pengguna bisa menambahkan halaman baru ke layar yang membuat tampilan ikon aplikasi di layar menjadi banyak. Fitur ini berguna untuk menambah efektivitas dan mempercepat akses ke berbagai aplikasi.
4. Bersifat terbuka (*open source*) agar *user* bisa mempelajari, membuat serta memodifikasi berdasarkan keinginan sendiri.
5. Kualitas suara dan grafik yang mumpuni karena pada aplikasi Android sudah disediakan standar suara dan video seperti MP3, mkv.
6. SQLite, sebagai basis data yang berguna sebagai media penyimpanan berbagai macam aplikasi *smartphone*.
7. Adanya tersedia berbagai jenis *library/services* yang bisa langsung dimanfaatkan, misalnya *browser*, GPS, kamera, *Bluetooth*, dan *Wifi*.
8. *Miracast*, merupakan bentuk protokol yang mengizinkan perangkat baru seperti *Samsung Galaxy Note 20* agar

dapat menikmati layanan *streaming* audio dan video pada TV yang memiliki fitur *Miracast*. Fitur ini memiliki kemiripan dengan *Airplay* yang telah lama ada di sejumlah perangkat keluaran Apple.

9. *Gesture Typing Keyboard*. Fitur yang mempunyai kemiripan dengan *swipe keyboard* yang telah lama diperkenalkan, namun google menyempurnakannya di Android 4.2 dengan memberikan akurasi yang lebih baik dan respons yang lebih cepat.

#### a. Versi Android

Ponsel pertama yang menggunakan *operating aplikasi* Android adalah HTC *Dream* yang dirilis pada 22 Oktober 2008. Pada tahun 2010, diperkirakan hampir semua *vendor* seluler di dunia memakai Android sebagai aplikasi operasinya. Berikut merupakan versi-versi Android yang sudah dirilis:

##### 1) Android versi 1.1

Google merilis Android versi 1.1 pada 9 Maret 2009, versi ini memiliki pembaruan estetika pada alarm, jam, aplikasi, pencarian suara, mengirimkan pesan dengan gmail, dan pembaruan aplikasi email.

##### 2) Android versi 1.5 (*Cupcake*)

Dirilis di pertengahan bulan Mei 2009 membawa sejumlah pembaruan dan menambahkan sejumlah fitur pada android versi ini yaitu mampu *me-record* dan menonton video dengan mode kamera, mengunggah

video ke *YouTube* dan *image* ke *Picasa* secara *real time* dari ponsel, *support Bluetooth A2DP*, memungkinkan terhubung secara *automatic* pada *headseat Bluetooth*, *screen animation*, dan *keyboard* pada layar yang bisa menyesuaikan dengan aplikasi.

### 3) Android versi 1.6 (*Donut*)

Versi ini lebih mengunggulkan aplikasi-aplikasi standar salah satunya yakni proses pencarian yang semakin baik, fitur galeri menjadi semakin rapi dan ramah pengguna, *support* resolusi layar WVGA, meningkatkan *Android market* dan memungkinkan *upgrade OS* ke versi 2.1.

### 4) Android versi 2.0/2.1 (*Éclair*)

Versi OS *Android 4<sup>th</sup> generation* yang mendapat tambahan *google map* pada versi ini, memiliki sejumlah tambahan fitur yang lebih meningkat dan canggih misalnya pada kamera 3,2 mega *pixel* telah *support flash*, tampilan kontak lebih istimewa, *HTML5* yang diusung pada perubahan tampilan antarmuka dengan peramban baru dan penambahan terakhir yaitu fitur *Bluetooth 2.1* yang memiliki kecepatan lebih cepat.

### 5) Android versi 2.2 (*Froyo: Frozen Yoghurt*)

Versi ini dirilis pada tahun 2010 yang telah dilengkapi dengan sejumlah fitur baru dan di klaim lebih cepat hingga lima kali dari versi sebelumnya, kemampuan *record video HD*, bisa meng-*install* aplikasi di kartu memori dan memungkinkan pembaruan otomatis di *Android market*.

6) Android versi 2.3 (*Gingerbread*)

Kelebihan dan peningkatan di versi ini yakni, performa peningkatan dalam *game*, suara, kualitas video, kamera, dan sebagainya. Juga dilengkapi papan ketikan virtual yang memiliki penyeleksi kata, manajemen *power*, *control* aplikasi dan *support* kamera yang meningkat serta membawa tampilan antarmuka hemat energi.

7) Android versi 3.1 (*Honeycomb*)

Versi ini khusus diciptakan untuk *device* tablet dan tidak *support* di ponsel. Desain khusus agar kecanggihan tablet bisa bekerja dengan maksimal.

8) Android versi 4.0 (*Ice Cream Sandwich*)

Versi pertama yang memakai fitur pembuka kunci dengan *face recognition*. Versi ini menampilkan antarmuka yang lebih halus jika dibandingkan versi lamanya serta memiliki jenis *font* baru.

9) Android versi 4.1 (*Jelly Bean*)

Versi ini memiliki penambahan peningkatan di antaranya, yaitu *voice search* yang semakin cepat, ramalan cuaca, info kemacetan, dan sebagainya. Peningkatan *keyboard* virtual.

10) Android versi 4.4 (*KitKat*)

Versi ini membawa sejumlah fitur yang terbilang baru di antaranya, SMS yang mempunyai integrasi langsung dengan google *hangouts*. Penambahan fitur *Cloud printing* serta dapat menerima perintah suara

dari Google *now* yang lebih hemat baterai dan bisa langsung menuju kamera walaupun layar terkunci.

#### 11) Android versi 5.0 (*Lollipop*)

Versi ini mengusung *user interface* yang dinamakan “*Material Design*”. 64-bit ART *compiler*. *Project volta*, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.

#### 12) Android versi 6.0 (*Marshmallow*)

Versi ini membawa *power management* baru bernama *Doze*. *Support fingerprint*, perizinan yang dirancang ulang, aplikasi pada waktu instalasi tidak lagi diberikan semua hak akses secara otomatis.

#### 13) Android versi 7.0 (*Nougat*)

Versi ini membawa sejumlah fitur baru yakni Fitur *Daydream* untuk menggunakan *Virtual Reality*, *Multitasking* ditingkatkan lebih baik, *Multi Windows* dan *Double tab chrome*, Penghemat data untuk meminimalisir data internet, Editor *Quick Settings*, Emoji baru, serta tombol *cancel* saat *men-download*

#### 14) Android versi 8.0 (*Oreo*)

Versi ini memiliki sejumlah fitur baru adapun fiturnya adalah Tampilan *interface homescreen*, *notification*, *app drawer*, *setting menu storage dan battery management*, *Support autofill*, *Smart Text Selection*, *Picture-In-Picture*, *Wifi Aware* atau *Neighbor Awareness Networking (NAN)*, *Background Limit*, *Bluetooth Audio*.

#### 15) Android versi 8.1 (*Oreo*)

Versi ini memiliki sejumlah fitur baru adapun fiturnya adalah penyempurnaan *App Shortcut*, Menu pengaturan lebih sederhana.

#### 16) Android versi 9.0 (*Pie*)

Versi ini memiliki sejumlah fitur baru meliputi *user interface* dan *notification* yang lebih *fresh*, *dark mode*, fitur *Screenshot* dengan banyak kelebihan, *Dashboard*, penguncian dengan keamanan ekstra, satu tombol *Home* untuk mengontrol navigasi, status baterai pada layar *ambient*.

#### 17) Android versi 10.0

Versi ini memiliki sejumlah fitur baru adapun fiturnya adalah Fitur *Smart reply*, *Navigation Gesture*, *Live caption*, *Digital Wellbeing*, Mode Fokus (Dicoding, 2020).

### **b. Android Studio**

Android Studio merupakan sebuah IDE untuk *Android Development* yang diperkenalkan Google pada acara Google I/O 2013. Android Studio adalah pengembangan dari *Eclipse IDE*, dan dibuat berdasarkan IDE Java populer, yaitu *IntelliJ IDEA*. Android Studio adalah IDE resmi untuk pengembangan aplikasi Android.

Android Studio di-*launching* pada 16 Mei 2013 saat Konferensi Google I/O oleh Produk Manajer Google, *Ellie Powers*. Android studio merupakan aplikasi gratis di bawah *Apache License 2.0*. Android Studio versi 0.1 dirilis pada bulan Mei 2013, lalu di luncurkan versi beta



0.8 pada bulan Juni 2014. Versi paling stabil di-*launching* di Desember 2014, berbasiskan *JetBrainns' IntelliJ IDEA*, Android Studio di rancang khusus untuk pengembangan android. Aplikasi ini sudah *support* Windows, Mac OS X, dan Linux (Syaputra, 2017).

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

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai pengembangan di *platform* Android menggunakan *Java Programming Language*. Android adalah *subset software* untuk *smartphone* yang meliputi *operating aplikasi, middleware* dan *key app* yang diluncurkan google. Saat ini disediakan Android SDK (*Software Development Kit*) untuk alat bantu dan *Application Programming Interface* sebagai awal pengembangan aplikasi di *platform* Android menggunakan *Java Programming Language*. Karena *platform* aplikasi netral, Android memungkinkan *user* untuk mengembangkan aplikasi yang diinginkan dan bukan suatu aplikasi standar *Smartphone* (Safaat, H., N., 2012).

**d. Java**

Java adalah bahasa pemrograman yang dibuat oleh James Gosling dan dibantu oleh rekan-rekan perusahaan *software*-nya yang bernama *Sun Microaplikasis*, di tahun 1991. Bahasa pemrograman ini mulanya diinisialisasi dengan nama "*Oak*", akan tetapi di tahun 1995 namanya diganti menjadi "*Java*".

Java berjalan di atas sebuah mesin penerjemah (*interpreter*) dengan nama *Java Virtual Machine* (JVM). JVM tersebut lah yang akan menerjemah kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin sehingga Bahasa Pemrograman *Java* disebut sebagai *portable programming language* karena bisa *running* di berbagai *operating aplikasi*, dengan ketentuan terdapat JVM di *operating aplikasi* tersebut.

Alasan terbesar penyusunan *Java Programming Language* yaitu dapat mengembangkan aplikasi-aplikasi yang bisa dipasang di bermacam *device* elektronik. Oleh karena itu, *Java* tidak boleh bergantung pada *platform*. Sebab itu *Java Programming Language* dikenal dengan istilah “*write once, run everywhere*” dalam dunia pemrograman sehingga *program code* hanya ditulis satu kali, tetapi bisa *running* di *platform* mana pun (Shalahudin, 2010).

## G. Flowchart

*Flowchart* atau bagan alur adalah bagan (diagram) alir yang merupakan sekumpulan simbol-simbol atau skema yang menunjukkan kegiatan-kegiatan program dari awal sampai akhir (Andika, 2017). Penggunaan *flowchart* yaitu agar dapat menyederhanakan rangkaian proses atau tata cara atau prosedur agar pemahaman user terhadap informasi menjadi lebih mudah.

*Flowchart* dapat digunakan pada segala sesuatu yang mengandung rangkaian kegiatan dan aliran data. Lima jenis *Flowchart* adalah sebagai berikut:

1. *Flowchart Aplikasi*
2. *Flowchart Document*
3. *Flowchart Schematic*
4. *Flowchart Program*
5. *Flowchart Process*

## H. Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan suatu bahasa standar industri untuk visualisasi, perancangan dan dokumentasi *software aplikasi*. UML memberi penawaran suatu pedoman baku yang diperuntukkan pada perancangan model aplikasi. Pemanfaatan UML *user* bisa merancang model untuk berbagai macam *software application*, dan dapat membuat *hardware* dapat menjalankan aplikasi tersebut, jaringan apa pun dan *operating aplikasi*, serta di-*code* dalam bahasa pemrograman apa pun.

Namun, karena UML dalam hal penggunaan juga memakai kelas dan operasi pada konsep dasarnya sehingga lebih pas untuk pengembangan *software* dalam bahasa-bahasa *object oriented*, misalnya C++, Java, C# atau VB.NET. Oleh karena itu, UML tetap bisa dipakai pada pemodelan *procedural application* pada Visual Basic ataupun C (Herlawati, Widodo Pudjo, P., 2011).

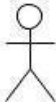




Layaknya bahasa-bahasa pada umumnya, UML mendefinisikan notasi dan *syntax* atau semantik. Notasi UML adalah kumpulan bentuk khusus untuk menggambarkan bermacam diagram *software*. Bentuk setiap diagram mempunyai





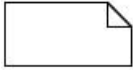
arti tertentu, dan *syntax* UML mendefinisikan bagaimana bentuk dari diagram tersebut bisa dikombinasi.

### 1. Use Case

*Use Case* Diagram adalah suatu jenis pemodelan yang dapat menggambarkan sifat aplikasi yang akan dibuat. Di *use case* pendeskripsian suatu interaksi antar satu ataupun lebih *actor* dengan *aplikasi* yang akan dirancang. Sehingga diagram *use case* dapat dipakai untuk mengetahui fungsionalitas apa saja yang terdapat di dalam sebuah *aplikasi* serta siapa yang mempunyai hak menggunakan fungsionalitas tersebut. Ada beberapa *symbol* dalam mendeskripsikan diagram *use case*, yakni *use case*, *actor* dan relasi. Yang menjadi poin utama dalam hal diagram *use case* adalah bukan mendeskripsikan *user interface* arsitektur dari aplikasi, kebutuhan *non-fungsional* dan tujuan performasi. Namun demikian penamaan *use case* adalah nama dituliskan sesederhana mungkin, bisa dimengerti dan memakai kata kerja (Herlawati, Widodo Pudjo, P., 2011).

Tabel 2. 1 Use Case Diagram

No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Spesifikasi himpunan peran yang <i>user</i> gunakan saat berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Relasi perubahan yang terjadi pada satu elemen <i>independent</i> dapat mempengaruhi elemen yang sedang bergantung padanya elemen <i>non-independent</i> .
3.		<i>Generalization</i>	Relasi <i>descendent</i> berbagi sifat dan struktur data terhadap objek di atasnya yaitu <i>ancestor</i> .
4.		<i>Include</i>	Mendefinisikan bahwa <i>use case</i> adalah sumber <i>eksplisit</i> .
5.		<i>Extend</i>	Mendefinisikan bahwa <i>use case</i> target memperluas sifat dari <i>use case</i> sumber.

6.		<i>Association</i>	Penghubung antara objek.
7.		<i>Aplikasi</i>	Mendefinisikan paket bahwa aplikasi memiliki batasan.
8.		<i>Use Case</i>	Pendeskripsian urutan suatu aksi yang menampilkan aplikasi dan memberi hasil yang terukur untuk <i>actor</i> .
9.		<i>Collaboration</i>	Aturan yang berinteraksi dan elemen lain yang sedang bekerja sama menyediakan perilaku yang lebih besar dari kuantitas dan elemennya.
10.		<i>Note</i>	Elemen fisik yang aktif ketika aplikasi <i>running</i> dan menunjukkan cerminan sumber daya komputasi.

Sumber: (Herlawati, Widodo Pudjo, P., 2011)

## 2. *Class Diagram*

*Class Diagram* mendeskripsikan struktur *aplikasi* dari segi pengertian *classes* yang akan dirancang agar dapat membangun *aplikasi*. Berikut merupakan bagian dari *Class Diagram*.






- a. Atribut adalah variabel-variabel yang dipunyai suatu kelas dan mendeskripsikan properti dengan sebaris teks di dalam *class* tersebut
- b. Operasi atau metode merupakan fungsionalitas yang dimiliki suatu kelas.

*Class Diagram* menggambarkan macam-macam objek pada *aplikasi* dan bermacam relasi statik yang terlibat antar diagram. *Class Diagram* juga menunjukkan properti dan operasi sebuah *class* dan keterbatasan yang ada dalam relasi objek tersebut, *Class Diagram* mendeskripsikan *structure* dan deskripsi *class*, *package*, dan objek serta relasi satu sama lain seperti *containment*, pewaris, asosiasi dan sebagainya.



Kelas memiliki tiga area pokok:

- a. Nama, yakni nama terhadap sebuah *class*.
- b. Atribut, yakni karakter dan sifat data terhadap objek dalam *class*.
- c. Operasi, yakni fungsionalitas atau perubahan yang memungkinkan bisa di implementasikan ke atau oleh suatu objek dalam *class* (Herlawati, Widodo Pudjo, P., 2011).

Tabel 2. 2 Class Diagram

No.	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Relasi antar <i>descendent</i> berbagi sifat dan <i>structure</i> data terhadap objek di atasnya <i>ancestor</i> .
2.		<i>Nary Association</i>	Usaha agar terhindar dari asosiasi terhadap lebih dari 2 objek.
3.		<i>Class</i>	Kumpulan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Penggambaran terhadap urutan aksi yang ditampilkan <i>aplikasi</i> yang memberi hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang dilakukan oleh suatu objek.



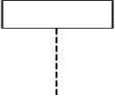

6.		<i>Dependency</i>	Relasi perubahan yang terjadi pada satu elemen <i>independent</i> dapat mempengaruhi elemen yang sedang bergantung padanya elemen <i>non-independent</i> .
7.		<i>Association</i>	Penghubung antara objek.

Sumber: (Herlawati, Widodo Pudjo, P., 2011)

a. *Sequence Diagram*

*Sequence Diagram* mendeskripsikan perilaku objek pada *use case* dengan menggambarkan keterangan waktu dan pesan yang bertransmisi antara objek. Sehingga agar dapat menggambarkan *sequence diagram* diperlukan pengetahuan objek-objek yang dipakai pada suatu *use case* serta metode-metode yang terlibat dalam kelas yang diinisiasi menjadi objek tersebut (Herlawati, Widodo Pudjo, P., 2011).

Tabel 2. 3 *Sequence Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , interaksi antar <i>user interface</i> .
2.		<i>Message</i>	Spesifikasi terhadap interaksi antara objek yang berisi informasi mengenai aktivitas yang berlangsung.

Sumber: (Herlawati, Widodo Pudjo, P., 2011)

#### d. *Activity Diagram*




*Activity Diagram* mendeskripsikan aliran kerja terhadap *activity* pada *aplikasi*. Perlu diketahui di sini bahwa *activity diagram* mendeskripsikan *aplikasi activity* bukan apa yang dikerjakan oleh *actor*, sehingga *aplikasi* dapat melakukan aktivitas. *Activity Diagram* memungkinkan perilaku paralel.

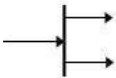
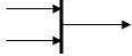

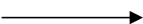

*Activity Diagram* juga sering dipakai untuk mengartikan hal-hal berikut:

- 1) Perancangan proses bisnis dalam hal tiap urutan *activity* yang dideskripsikan adalah proses bisnis *aplikasi* yang didefinisikan.

- 2) Pengurutan atau pengelompokan *interface* terhadap *aplikasi* bahwa tiap *activity* dianggap mempunyai suatu desain *user interface*.
- 3) Perancangan pengujian bahwa tiap *activity* dianggap membutuhkan suatu pengujian dan diperlukan definisi studi kasusnya (Herlawati, Widodo Pudjo, P., 2011).

**Tabel 2. 4** *Activity Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Start Point</i>	<i>Start point</i> menunjukkan awal dimulainya alur kerja dalam <i>activity diagram</i> . Di UML, <i>start Point</i> disimbolkan lingkaran hitam.
2.		<i>End Point</i>	<i>End point</i> menunjukkan berakhirnya suatu <i>activity diagram</i> . Kadang kala terdapat lebih dari satu <i>end point</i> di suatu <i>activity diagram</i> . <i>End point</i> disimbolkan dengan simbol lingkaran hitam yang dikelilingi oleh lingkaran.
3.		<i>Activities</i>	<i>Activities</i> menunjukkan suatu <i>task</i> dalam alur kerja.

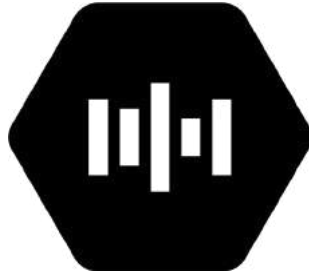
4.		<i>Fork</i> (Pencabangan)	<i>Activity Diagram</i> memungkinkan terjadinya pengelompokan jalur yang dapat menunjukkan objek mana yang terlibat pada suatu aktivitas. Percabangan tunggal terjadi karena adanya yang saling terkoneksi dengan aktivitas berikutnya.
5.		<i>Joint</i> (Penggabungan / Merge)	Transisi yang bisa menciptakan cabang ke dua atau lebih percabangan
6.		<i>Decision</i> (Kondisi)	<i>Decision</i> merupakan <i>activity diagram</i> yang menjadi indikasi satu kondisi yang memungkinkan terjadinya perbedaan transisi.
7.		<i>State Transition</i> (Alur)	<i>State transition</i> mendeskripsikan <i>activity</i> setelah <i>activity</i> sebelumnya.
8.		<i>Swimlane</i> (Kelompok)	<i>Swimlane</i> adalah penggambaran objek mana yang memiliki peranan terhadap aktivitas tertentu.

Sumber: (Herlawati, Widodo Pudjo, P., 2011)

## I. SQLite

SQLite adalah sebuah basis data yang bersifat ACID-compliant dan memiliki *library* yang relatif kecil. SQLite adalah *project* yang bersifat *public domain* yang disusun oleh D. Richard Hipp. SQLite merupakan *open source database* yang telah cukup lama ada, terbilang stabil, dan sangat populer pada *device* kecil, termasuk Android. SQLite terbilang ringan untuk setiap aplikasi yang menggunakannya. Aplikasi memungkinkan mengatur *relational database engine* untuk men-secure data dengan aman dan efisien. Pada Android, SQLite dijadikan satu di dalam *Android runtime*, oleh karena itu tiap aplikasi Android bisa membuat *database* SQLite. Dikarenakan SQLite memakai *user interface* SQL, sehingga penggunaan untuk orang awam pun menjadi cukup mudah untuk digunakan (Djuandi, 2011).

Adapun *tools* yang cukup populer adalah *SQLite Manager*. *SQLite Manager* merupakan suatu program yang terbilang cukup membantu *programmer* untuk pengelolaan basis data SQLite, contohnya mendesain tabel, *index* dan menjalankan perintah-perintah SQL. Program ini terbilang mudah untuk dipakai dan sangat bermanfaat untuk keperluan yang praktis (Djuandi, 2011).

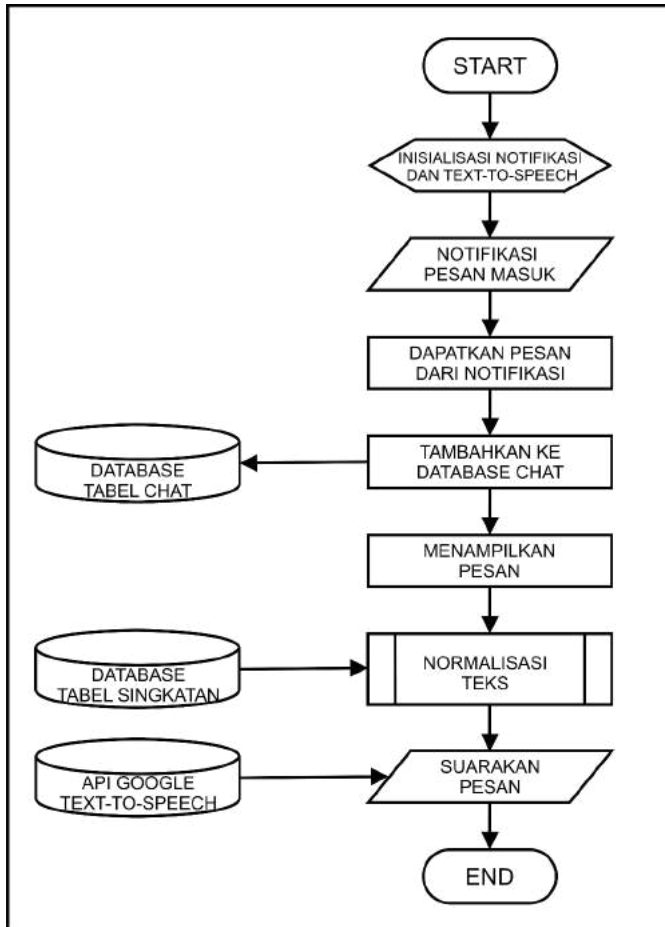


## BAGIAN KETIGA

### Penerapan Aplikasi Pembaca Pesan

#### A. Permodelan Aplikasi yang Diusulkan

Permodelan aplikasi adalah sekumpulan *activity* dalam perancangan model perwakilan atau gagasan terhadap objek ataupun *actual situational* ke bentuk sederhana dari sesuatu fakta yang rumit. Pada permodelan atau rancangan desain *aplikasi, item, dan component* dibutuhkan perancangan agar tercapai tujuan yang dapat memudahkan dan menambah efektivitas kinerja dan *activity* serta mendeskripsikan tentang suatu mekanisme yang benar dalam merancang suatu *aplikasi* yang sesuai dengan kebutuhan. Permodelan yang digunakan yaitu *flowchart* pada gambar 4.1 berikut ini.



**Gambar 4. 1** Flowchart Aplikasi yang diusulkan

Menurut Gambar 4.1, *Flowchart* aplikasi dapat dijelaskan sebagai berikut:

### 1. Inisialisasi Notifikasi dan *Text to Speech*

Pada tahap ini, sebelum aplikasi dapat mengkonversi teks menjadi suara, diperlukan inisialisasi. Inisialisasi ada dua macam, yaitu inisialisasi *Notification Listener Service* dan *Text to Speech Google API*. Aplikasi ini membutuhkan izin dari user untuk menjalankan *Notification Listener Service* di *Smartphone*.

```

1 <service
2 android:name=".NotificationService"
3     android:label="@string/app_name"
4
5 android:permission="android.permission.BIND_NOTIFICATION_LISTENER_SERVICE">
6     <intent-filter>
7         <action
8 android:name="android.service.notification.NotificationListenerService" />
9     </intent-filter>
10 </service>

```

**Gambar 4. 2 Inisialisasi *Notification Listener Service***

Gambar 4.2 merupakan *source code* untuk mendapatkan izin menerapkan *Notification Listener Service* yang ditempatkan pada *file Android Manifest*.

Inisialisasi *Notification Listener Service* merupakan proses yang berjalan di belakang (*background thread*) berfungsi untuk mengambil segala informasi dari pemberitahuan atau notifikasi yang masuk ke *smartphone* Android. Inisialisasi ini dilakukan ketika aplikasi pertama kali dijalankan.

Pada saat *Notification Listener Service* terinisialisasi, selanjutnya aplikasi melakukan inisialisasi *Text to Speech* (TTS).



```
public class NotificationService
extends NotificationListenerService
implements TextToSpeech.OnInitListener {
```

**Gambar 4.3 Inisialisasi TTS**

Gambar 4.3 merupakan *source code* untuk menerapkan *Text to Speech* ke dalam *class Notification Service* dengan turunan dari *Notification Listener Service*. *Class Notification Service* akan dibahas pada tahap selanjutnya. Kemudian mengatur bahasa yang digunakan:

```
tts.setLanguage(new Locale("id","ID"));
```

Namun inisialisasi ini membutuhkan *voice data* yang diunduh dari *Google Text-to-Speech*. Saat inisialisasi berhasil aplikasi akan mengatur bahasa yang digunakan bahasa Indonesia.

## 2. Notifikasi Pesan Masuk

Pada Gambar 4.3 *class Notification Service* digunakan untuk membaca pemberitahuan pada *Status Bar*. Dengan menggunakan *method onNotificationPosted* dari *Notification Listener Service* memungkinkan pemberitahuan yang diposting pada *Status Bar* akan dijadikan sebagai objek dengan *StatusBarNotification*. Kemudian, objek yang diterima akan diproses pada tahap selanjutnya. Ini merupakan tahap untuk *input* pesan yang diambil dari notifikasi pesan instan seperti Telegram ataupun WhatsApp.

### 3. Dapatkan Pesan dari Notifikasi

Tahap ini merupakan proses dari *on Notification Posted* yang memungkinkan objek yang telah diterima diproses dengan memisahkan string menjadi bagian-bagian pesan yang terdiri dari *pack*, *title*, dan *text*. Proses pertama yang dilakukan adalah mendapatkan *pack* atau asal aplikasi dengan *method get Package Name*, kemudian mendapatkan *title* atau judul dengan nama variabel "*android.title*" dan mendapatkan *text* dengan nama variabel "*android.text*".

```
Log.i("NotifyService", "got notification");
String pack = sbn.getPackageName();
Bundle extras = sbn.getNotification().extras;
String title = extras.getString("android.title");
String text = extras.getString("android.text");
mPreviousNotificationKey = sbn.getKey();
mPreviousNotification = sbn.getNotification();
```

**Gambar 4. 4 Objek ke string**

Jika *pack* yang diterima bernilai "*org.telegram.messenger*" dan "*com.whatsapp*". Kemudian membuat *intent* baru menggunakan *intent filter "Msg"* untuk mengirim nilai yang diperoleh sebagai pesan *Broadcast* menggunakan *method*:

```
LocalBroadcastManager.getInstance(context).send  
Broadcast(msgrcv);
```

Pesan Broadcast yang diterima yaitu *string pack*, *title*, *text* dan juga *date and time*. String *Date and time* didapatkan dari hasil ketika pesan ditampilkan pada *Status Bar* sesuai waktu menerima pesan menggunakan *method Simple Date Format* ("*dd-MM-yyyy HH:mm:ss*"). Kemudian, nilai dari

*send Broadcast (msgrcv)* akan dikirim dan diterima sesuai *fragment* yang dibuat.

#### 4. Tambahkan ke *Database Chat*

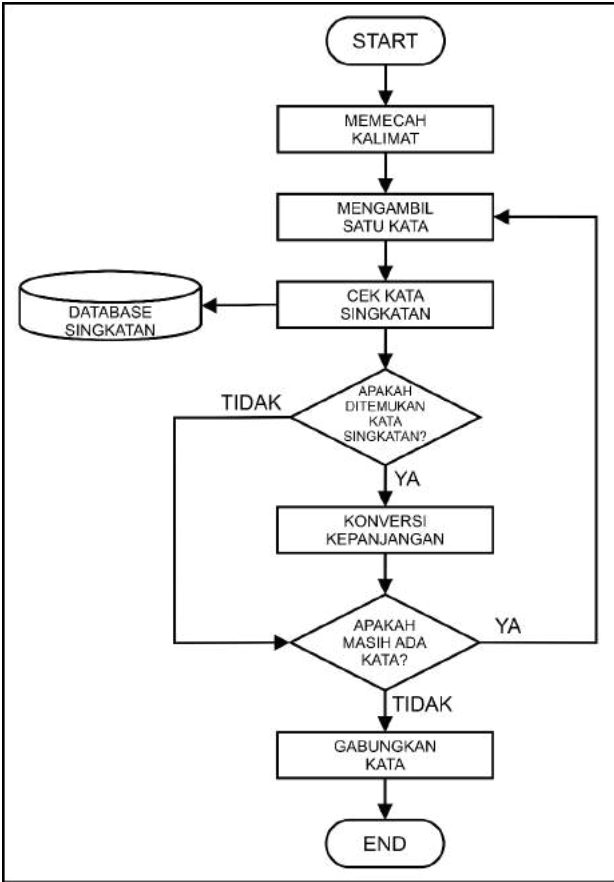
Ketika *pack* yang diterima sesuai dengan nilai yang ditentukan, maka *pack*, *title*, *text*, *date* dan *time* pesan akan disimpan ke dalam tabel *chat* yang berada pada *database* SQLite. Pesan yang tersimpan diperlukan untuk mendukung fasilitas pengolahan data ataupun validasi hasil konversi teks ke ucapan.

#### 5. Menampilkan Pesan

Pesan yang telah disimpan pada *database* akan ditampilkan pada menu utama aplikasi.

#### 6. Normalisasi Pesan

Proses pemecahan kalimat kemudian menggabungkan kembali menjadi satu kalimat dengan membuat fungsi *parse Singkatan* berlangsung pada proses ini. *Flowchart* parsing kata bisa dilihat pada Gambar 4.5.



**Gambar 4.5** *Flowchart* Parsing Kalimat

Berikut adalah penjelasan Gambar 4.5 Parsing Kalimat:

- a. Memecah kalimat

Pada tahap ini string isi pesan yang telah diterima dari proses sebelumnya akan dipecah menjadi beberapa kata dengan karakter spasi sebagai pembatas. Dengan fungsi `words.split("\\s+").` Sebagai contoh teks

pesan “Bsk/ mau/ ka/ ke/ kmpps/ cma/ tkt/ ka/ hjn/”.

b. Mengambil satu kata

Tahap ini akan memproses kata pertama, yaitu “Bsk” yang diterima dari pemecahan pada tahap sebelumnya sampai dengan kata terakhir pada kalimat yang didapatkan.

c. Cek kata singkatan

Tahap cek kata singkatan ini akan memproses kata pertama kemudian membandingkan kata tersebut dengan yang terdapat di basis data tabel singkatan. Apabila terdapat kesamaan kata dengan kata pada tabel singkatan, aplikasi akan melakukan tahap konversi kepanjangan. Akan tetapi, apabila tidak terdapat kesamaan kata terhadap tabel singkatan, aplikasi akan mengecek apakah masih ada kata saat proses pemecahan.

d. Konversi kepanjangan

Tahap ini merupakan tahap kondisi apabila terdapat kesamaan kata terhadap basis data tabel singkatan, singkatan tersebut akan dikonversi berdasarkan kepanjangan yang dimaksud. Sebagai contoh kata singkatan “kmpps” sama dengan kata singkatan yang terdapat dalam database “kmpps” dengan kepanjangan kata “kampus” sehingga aplikasi akan mengkonversi “kmpps” menjadi “kampus”.

e. Apakah masih terdapat kata?

Tahap ini merupakan tahap pengulangan. Jika masih ada kata dalam kalimat yang telah dipecah pada tahap memecah kalimat, aplikasi akan kembali ke tahap mengambil satu kata sesuai banyaknya panjang kata yang telah dipecah.

f. Gabungkan kata

Proses pada tahap ini adalah menggabungkan kembali semua kata dari tahap-tahap sebelumnya sehingga membuat kalimat yang utuh.

## 7. Suarakan pesan

```
if(tts != null && prefManager.isTtsOn()){
    ParseSingkatan parser = new ParseSingkatan(context);
    text = parser.parseSingkatan(text);
    text = parser.parseDatabase(text);
    tts_speak("Pengirim,"+title+" "+text,TextToSpeech.QUEUE_ADD,null);
}
```

**Gambar 4. 6 Output suara setelah normalisasi**

Pada tahap ini isi teks yang telah melalui proses normalisasi pada pesan yang diterima, selanjutnya dikonversi menjadi suara oleh *text-to-speech* dengan menyebutkan [Pengirim], [title] atau yang mengirimkan dan [text] yang merupakan isi pesan masuk. *API Google Text-to-Speech* memungkinkan untuk mengatur antrean pemrosesan konversi teks menjadi suara. Terdapat dua jenis pengaturan antrean, yaitu *QUEUE\_ADD* yang akan menyelesaikan proses konversi satu per satu sebelum memproses pesan baru yang masuk pada antrean. Selanjutnya adalah antrean *QUEUE\_FLUSH* yang akan memproses pesan baru pada

antrean dan menghentikan proses konversi pesan yang sedang berjalan.

## B. Perancangan Fitur

Pada rancangan fitur digunakan *Method setup View Pager* untuk menampilkan *fragment* ke dalam *activity* sehingga satu tampilan akan menampilkan dua buah *tab* untuk melihat pesan dari Telegram dan WhatsApp.

```
private void setupViewPager(ViewPager viewPager) {
    ViewPagerAdapter adapter = new
    ViewPagerAdapter (getSupportFragmentManager());
    adapter.addFragment(new Telegram(), "Telegram");
    adapter.addFragment(new Whatsapp(), "Whatsapp");
    viewPager.setAdapter(adapter);
}
```

**Gambar 4.7** *Fragment dalam activity*

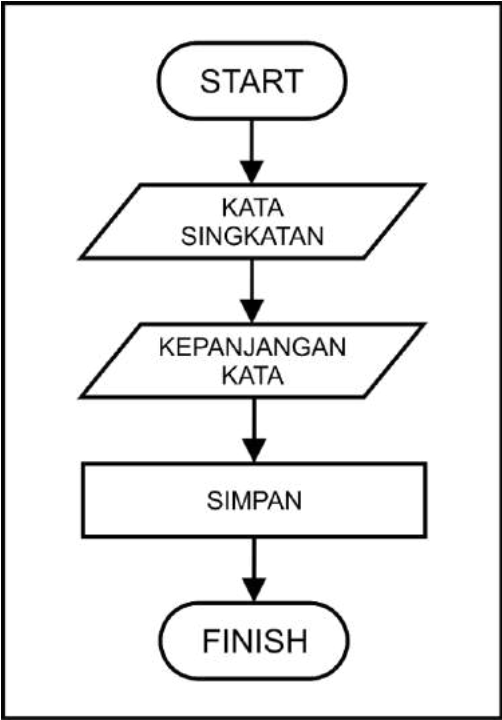
Pada class *Chat Adapter* extends *android.support.v7.widget.RecyclerView* memungkinkan mengatur tata letak ketika menaruh *package, tittle, text, date and time* serta dapat menampung lebih banyak data.

Adapun beberapa fitur pada *activity* ini di antaranya:

### 1. Menu *Input* Singkatan

*Input* singkatan dibuat untuk membenarkan *output* suara yang sesuai dengan tata bahasa yang benar. *Input* Singkatan menggunakan *method on Options Item Selected* yang terdapat pada *options* menu kemudian ditampilkan menggunakan *Alert Dialog* yang di dalamnya terdapat dua *Edit Text* untuk proses *input*. Proses *input* singkatan dimulai dari *input* kata singkatan, kemudian *input* kata

yang dimaksud. Sebagai contoh singkatan “Kmps” dan kepanjangan “Kampus”. Setelah proses *input* selesai *user* dapat menekan tombol simpan dan tidak. Jika pilihan simpan, singkatan dan juga kepanjangan akan tersimpan ke dalam *database* pada tabel singkatan. *Flowchart input* kata singkatan bisa dilihat pada gambar 4.8:



**Gambar 4. 8** *Flowchart Input Singkatan*

2. Hapus Pesan

Hapus pesan dibuat untuk memudahkan dalam manajemen pesan ketika data yang diterima terlalu banyak. Menghapus pesan terbagi menjadi dua pilihan,



di antaranya menghapus berdasarkan *Id* dan *pack* pada tabel *Chat*. Menghapus berdasarkan *Id* memungkinkan menghapus satu pesan, dengan menekan lama pesan yang ingin dihapus kemudian menekan pilihan “Ya” pada *Alert Dialog* yang muncul maka pesan telah terhapus. Menghapus berdasarkan *pack* dari Telegram dan WhatsApp. Pilihan menghapus ini terdapat pada *options menu* kemudian menekan pilihan “Ya” ketika *Alert Dialog* tampil.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/notifsetting"
    android:icon="@drawable/ic_activate"
    android:visible="true"
    android:title="Aktivasi"
    app:showAsAction="always" />
  <item
    android:id="@+id/exitg"
    android:icon="@drawable/icexit"
    android:title="Keluar"
    android:visible="true"
    app:showAsAction="always" />
  <item
    android:id="@+id/add_database_singkatan"
    android:icon="@drawable/ic_add_circle_outline_black_24dp"
    android:title="Tambah Singkatan"
    app:showAsAction="never" />
  <item
    android:title="Hapus Pesan"
    android:icon="@drawable/ic_baseline_delete_forever_24"
    app:showAsAction="never">
    <menu>
      <item
        android:id="@+id/hapus_pesani"
        android:title="Hapus Pesan Whatsapp"
        android:icon="@drawable/ic_baseline_delete_forever_24"
        app:showAsAction="never" />
      <item
        android:id="@+id/hapus_pesant"
        android:title="Hapus Pesan Telegram"
        android:icon="@drawable/ic_baseline_delete_forever_24"
        app:showAsAction="never" />
    </menu>
  </item>
  <item
    android:title="Kontrol Ucapan"
    app:showAsAction="never">
    <menu>
      <item
        android:id="@+id/ttson"
        android:title="Aktifkan Ucapan"
        android:icon="@drawable/ic_up"
        app:showAsAction="never" />
      <item
        android:id="@+id/ttsoff"
        android:title="Matikan Ucapan"
        android:icon="@drawable/ic_off"
        app:showAsAction="never" />
    </menu>
  </item>
</menu>
```

Gambar 4. 9 Option Menu

### 3. Suarakan

Tahap ini memungkinkan *user* dapat mendengarkan kembali pesan yang diterima. Pilihan ini terdapat pada *class Chat Adapter* yang akan ditampilkan terus-menerus sebanyak jumlah *List Chat* yang diterima dengan *icon button voice*. Ketika tombol ditekan *method set On Click Listener* diaktifkan, maka *event on Click* akan menjalankan fungsi *text-to-speech* pada pesan yang dipilih.

### C. Perancangan *Database*

Rancangan basis data pada aplikasi diberi nama *autospeech.db* dan terdiri dari dua tabel. Tabel 4.1 dibuat untuk mendapatkan *package*, *title*, *text* dan *datetime* ketika menerima pesan serta berfungsi sebagai pengelola dalam manajemen pesan nantinya. Tabel 4.2 dibuat untuk menyimpan kata singkatan dan kepanjangan yang nantinya digunakan sebagai pembanding dan hasil keluaran. Aplikasi *database* yang digunakan adalah SQLite dengan menggunakan turunan dari *SQLite Asset Helper* untuk mendapatkan akses penyimpanan *database* baca dan tulis ke *database* yang memungkinkan dalam manipulasi data seperti mencari, menambah, mengubah atau menghapus satu atau lebih *record* pada tabel. SQLite ini telah tersedia *built-in* pada Android SDK.

Berikut adalah tabel yang digunakan dalam basis data:

**Tabel 4.1 Tabel Chat**

Field	Type	Range Data	Atribut	Fungsi
id	Integer	-	<i>Primary key, auto increment</i>	Sebagai indeks untuk pengurutan pesan
Pengirim	Text	1024	-	Identitas pengirim pesan
Pesan	Text	1024	-	Isi Pesan
Pack	Text	-		<i>package whatsapp atau telegram</i>
DateTime	Text	-	-	Waktu penerimaan

**Tabel 4. 2 Tabel Singkatan**

Field	Type	Atribut	Fungsi
Id	Integer	<i>Primary key, auto increment</i>	Sebagai indeks untuk pengurutan
Word	Text	-	Sebagai masukan kata
Meaning	Text	-	Kata yang akan diucapkan

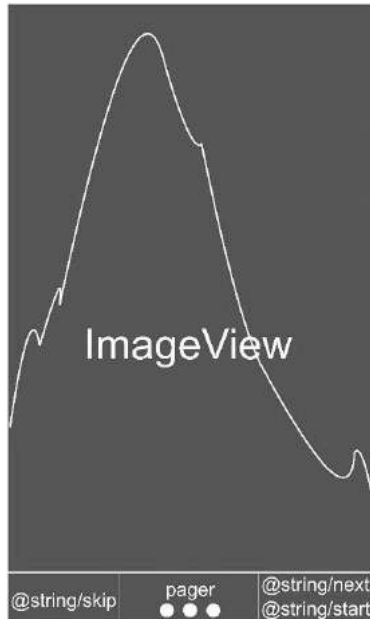
## D. Perancangan Antar Muka

Antarmuka merupakan aspek aplikasi komputer atau program yang dapat dilihat, didengar, berinteraksi atau dipersepsikan oleh *user*, serta perintah atau aplikasi mekanik yang dipakai oleh *user* agar dapat memegang kendali operasi dan pemasukan data. Rancangan *user interface* bisa dilihat dalam uraian sebagai berikut:

### 1. Halaman *Welcome Activity*

Rancangan berikut merupakan rancangan halaman *welcome activity*, antara lain:

- a. `ImageView`, Panduan singkat aplikasi
- b. `@String/skip`, `String` perintah lewatkan halaman
- c. `Pager`, `dots` navigasi halaman
- d. `@String/next`, `String` perintah ke halaman selanjutnya
- e. `@String/start`, `String` perintah memulai aplikasi



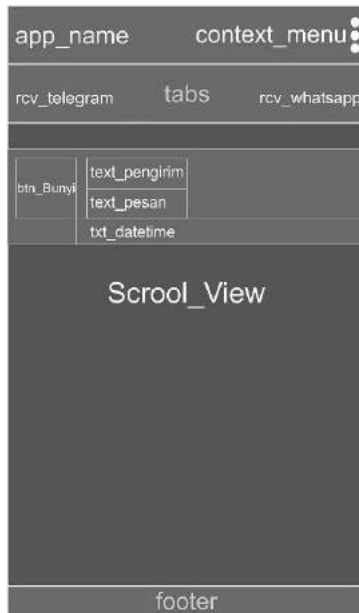
**Gambar 4. 10 Rancangan** *Welcome Activity*

## 2. Rancangan Halaman Utama

Berikut merupakan rancangan halaman utama, antara lain:

- a. `app_name`, Nama Aplikasi
- b. `context_menu`, pilihan fitur menu aplikasi
- c. `tabs rcv_telegram`, tab menampilkan pesan telegram
- d. `tabs rcv_whatsapp`, tab menampilkan pesan WhatsApp
- e. `Scrool_View`, tampilan halaman yang dapat di-*scroll*
- f. `btn_Bunyi`, *button* untuk mendengarkan pesan
- g. `text_pengirim`, berisikan nama pengirim pesan

- h. `text_pesana`, berisikan isi pesan yang diterima
- i. `txt_datetime`, tanggal dan waktu pesan diterima

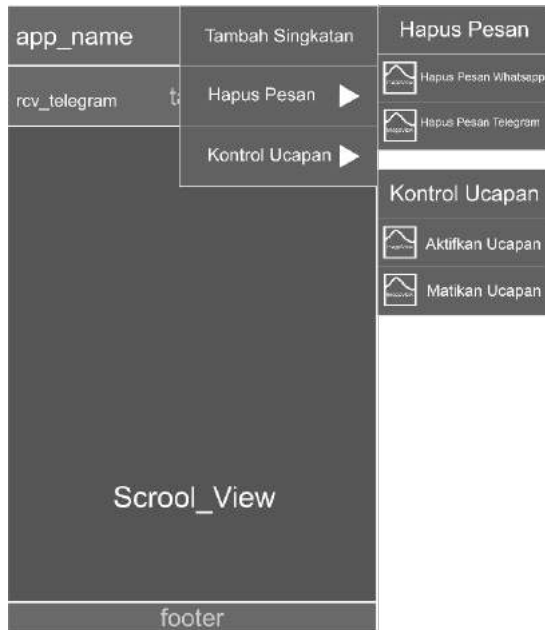


**Gambar 4. 11 Rancangan Halaman Utama**

### 3. Rancangan *Context Menu*

Rancangan berikut merupakan rancangan fitur-fitur aplikasi, antara lain:

- a. Tambah Singkatan, fitur untuk menambah kata singkatan
- b. Hapus Pesan WhatsApp, fitur untuk menghapus semua pesan WhatsApp yang tampil di aplikasi
- c. Hapus Pesan Telegram, fitur untuk menghapus semua pesan telegram yang tampil di aplikasi
- d. Aktifkan Ucapan, fitur untuk mengaktifkan ucapan
- e. Matikan Ucapan, fitur untuk menonaktifkan ucapan



**Gambar 4. 12 Rancangan Context Menu**

#### 4. Rancangan Tambah Kata Singkatan

Rancangan berikut merupakan rancangan untuk menambah kata singkatan, antara lain:

- add\_database\_singkatan, kotak dialog untuk memasukkan kata singkatan
- Kata Singkatan, *form* untuk mengisi kata singkatan
- Kata yang Dimaksud, *form* untuk mengisi kepanjangan kata
- Negative Button, perintah untuk membatalkan
- Positive Button, perintah untuk menyetujui



**Gambar 4. 13 Rancangan Tambah Kata Singkatan**

#### 5. Rancangan Hapus Pesan Telegram

Rancangan berikut merupakan rancangan untuk menghapus semua pesan telegram, antara lain:

- hapus\_pesant, kotak dialog untuk menghapus pesan telegram
- ImageView, Simbol peringatan
- TextView, berupa teks Konfirmasi dan deskripsi perintah
- NegativeButton, perintah untuk membatalkan
- PositiveButton, perintah untuk menyetujui





**Gambar 4. 14 Rancangan Hapus Pesan Telegram**

6. Rancangan Hapus Pesan WhatsApp

Rancangan berikut merupakan rancangan untuk menghapus semua pesan WhatsApp, antara lain:

- a. hapus\_pesannya, kotak dialog untuk menghapus pesan WhatsApp
- b. ImageView, Simbol peringatan
- c. TextView, berupa teks Konfirmasi dan deskripsi perintah
- d. NegativeButton, perintah untuk membatalkan
- e. PositiveButton, perintah untuk menyetujui



**Gambar 4. 15 Rancangan Hapus Pesan WhatsApp**

## E. Implementasi Antarmuka

Implementasi *user interface* dikembangkan memakai *layout* dengan ekstensi *.xml*. Implementasi yang dikembangkan adalah sebagai berikut.

### 1. Icon Aplikasi



**Gambar 4. 16 Ikon Aplikasi Pembaca Pesan**

Untuk menjalankan aplikasi, klik ikon aplikasi Pembaca Pesan hingga muncul halaman *Welcome Activity*.

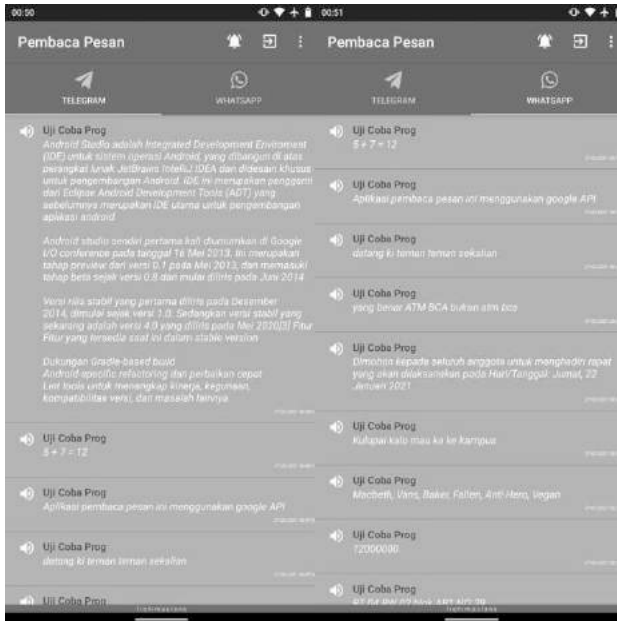
## 2. Tampilan Halaman *Welcome Activity*



**Gambar 4. 17** Tampilan *Welcome Activity*

Pada saat masuk di halaman *Welcome Activity*, user akan melihat panduan singkat dan diperkenalkan untuk mengaktifkan *Permission* Akses Notifikasi. Kemudian klik SELANJUTNYA.

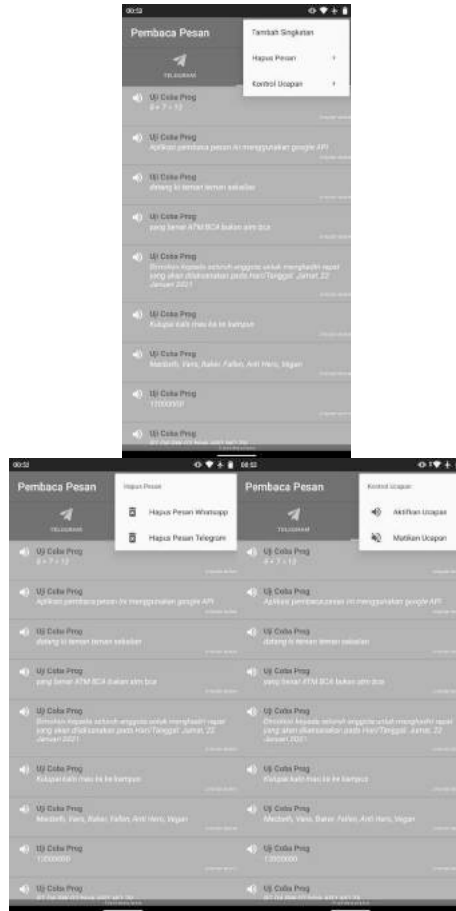
### 3. Tampilan Halaman Utama



**Gambar 4. 18 Tampilan Halaman Utama**

Tampilan berikutnya yaitu tampilan antarmuka pada aplikasi saat *user* menjalankan aplikasi. Dalam satu tampilan terdapat dua *tabLayout* yang dibagi menjadi *fragment* halaman Telegram dan *fragment* halaman WhatsApp sehingga tampilan pada *layout activity* menjadi lebih atraktif.

#### 4. Tampilan *Action Context Menu*



**Gambar 4. 19 Tampilan *Action Context Menu***

Pada halaman antarmuka ini terdapat fitur *action bar* dan *options menu*. *Action bar* adalah elemen UI yang terletak di layar bagian kanan atas sedangkan *options menu* ditampilkan ketika pengguna menekan tombol atau ikon menu untuk menampilkan perintah-perintah umum yang tersedia.

## 5. Tampilan Tambah Kata Singkatan



**Gambar 4. 20 Tampilan Tambah Kata Singkatan**

Tambah singkatan merupakan menu untuk *input* singkatan dan *input* kepanjangan kata yang nantinya sebagai *output* ucapan yang benar. Ketika memilih *option* tambah singkatan *Alert Dialog* akan ditampilkan yang di dalamnya terdapat dua *edittext* untuk melakukan *input* kata singkatan dan kepanjangan kata yang dimaksud serta dua pilihan simpan untuk menyimpan *input* singkatan dan kepanjangan, dan tidak untuk membatalkan pilihan.

## 6. Tampilan Hapus Pesan Telegram



**Gambar 4. 21 Tampilan Hapus Pesan Telegram**

Hapus pesan Telegram merupakan pilihan untuk menghapus seluruh pesan berdasarkan *package* yang dipilih dengan menekan tombol pilihan "HAPUS" untuk menghapus dan tombol pilihan "TIDAK" untuk membatalkan yang muncul sebagai *alert dialog* pada layar aplikasi.

## 7. Tampilan Hapus Pesan WhatsApp



**Gambar 4. 22 Tampilan Hapus Pesan WhatsApp**

Hapus pesan WhatsApp merupakan pilihan untuk menghapus seluruh pesan berdasarkan *package* yang dipilih dengan menekan tombol pilihan “HAPUS” untuk menghapus dan tombol pilihan “TIDAK” untuk membatalkan yang muncul sebagai *alert dialog* pada layar aplikasi.



## 8. Tampilan Hasil Penerimaan Pesan



**Gambar 4. 23 Tampilan Hasil Penerimaan Pesan**

Ketika mendapatkan pemberitahuan pesan masuk, maka aplikasi akan mengkonversi teks isi pesan yang diterima menjadi ucapan, kemudian *activity* akan menerima (*BroadcastReceiver*) dari *sendBroadcast* sehingga membuat pesan yang diterima ditampilkan secara *real time* pada tiap *fragment* yang dituju. Gambar 4.23 menunjukkan pesan baru telah diterima dari *package* WhatsApp, *tittle* (pengirim) dan *text* (isi pesan). Selanjutnya, aplikasi *text-to-speech* menerjemahkan isi pesan tersebut menjadi suara [ap-li-ka-si pem-ba-ca pe-san i-ni meng- gu-na-kan goo-gle A-P-I],

kemudian pesan secara *real time* akan tampil dengan nama pengirim “Uji Coba Prog”, isi pesan “Aplikasi pembaca pesan ini menggunakan google API” dan waktu terima “27-02-2021 00:39:30”.

Pesan yang diterima akan tersimpan ke dalam *database* pada tabel *Chat* kemudian ditampilkan pada setiap *fragment* Telegram dan *fragment* WhatsApp. Ucapan juga dapat diaktifkan dan dinonaktifkan melalui kontrol ucapan, yang dalam hal ini tidak akan mengganggu proses menampilkan pesan di setiap *fragment*. Setiap pesan yang ditampilkan dapat didengarkan kembali ketika *button* suara ditekan dan apabila pesan ditekan lama akan menampilkan menu *pop-up* peringatan untuk memilih menghapus pesan atau tidak.

## F. Pengujian Aplikasi

Pengujian sebuah *aplikasi* adalah pokok utama yang memiliki tujuan agar dapat mengetahui tercapai ataupun tidaknya tujuan dibuatnya *aplikasi*, dan menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada *software* yang dikembangkan sesuai standarisasi dengan tujuan pengembangan *software* tersebut.

Pengujian dilakukan pada setiap *view*, *service* dan *text to speech*. Pengujian ini dilakukan secara *Black-Box*, yaitu diuji dengan memperhatikan *input* ke aplikasi dan *output* terhadap *aplikasi*. Rencana pengujian direncanakan terbagi dua, yaitu pengujian fitur dan pengujian membaca pesan yang diterima. Rencana pengujian fitur *aplikasi* ini bisa dilihat pada Tabel 4.3.

Tabel 4. 3 Tabel Rencana Pengujian Fitur


No.	Fitur yang diuji	Detail Pengujian	Jenis Uji
1	<i>Service</i>	Menjalankan tugas aplikasi yang bekerja di belakang layar Menerima pesan Telegram dan WhatsApp	<i>Black-box</i>
2	<i>Text-to-Speech</i>	Mengkonversi teks ke ucapan dari pesan yang diterima Mendengarkan pesan yang diterima	<i>Black-box</i>
3	<i>View Message</i>	Menampilkan pesan yang telah diterima Menampilkan pesan per halaman	<i>Black-box</i>
4	<i>Sound</i>	Menekan <i>btn_Bunyi</i> untuk menjalankan fungsi <i>Text-to-Speech</i>	<i>Black-box</i>
5	<i>Delete</i>	Menghapus pesan berdasarkan <i>package</i> atau <i>pop-up</i> Menampilkan peringatan menghapus atau tidak	<i>Black-box</i>
6	<i>Input Singkatan</i>	Menampilkan kotak dialog <i>input</i> singkatan dan kepanjangan Memasukkan singkatan ke dalam <i>database</i>	<i>Black-box</i>



Pengujian aplikasi akan dilakukan di beberapa *brand device* yang berbeda guna mendeteksi kesalahan (*bug*) dan mengetahui kualitas aplikasi yang dibangun, *device* yang akan digunakan antara lain:


1. Xiaomi MI A2
2. Samsung A7 (A750)
3. OPPO A7
4. Realme 6 Pro
5. Nokia 3

Berdasarkan rencana uji terhadap aplikasi bisa dilihat pada tabel 4.4.

**Tabel 4. 4 Tabel Hasil Pengujian Fitur**

Aksi/Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengaktifkan Pengaturan <i>Notification Access</i> pada <i>smartphone</i>	Menerima pesan dari <i>Telegram</i> dan <i>WhatsApp</i> tanpa membuka Aplikasi <i>Pembaca Pesan</i>	Aplikasi menerima pesan dari <i>Telegram</i> dan <i>WhatsApp</i> tanpa membuka Aplikasi <i>Pembaca Pesan</i>  A screenshot of a WhatsApp notification on a smartphone. The notification shows a message from 'Uji Coba Prtg' with the text 'Agihast pembaca pesan ini menggunakan google A7I'. Below the message, there is a 'REPLY' button and the name 'MAAK AS RIGAB'. At the bottom, there is a small icon and the text 'Uji Coba Prtg' and 'Aplikasi pembaca pesan ini menggunakan google A7I'.	<b>Berhasil</b>
Mendownload <i>voice</i> suara Bahasa Indonesia pada <i>Google Text To Speech</i>	Menerima pesan <i>notification</i> dari <i>Telegram</i> dan <i>WhatsApp</i> kemudian mengubah teks pesan menjadi ucapan	Aplikasi menerima pesan <i>notification</i> dari <i>Telegram</i> dan <i>WhatsApp</i> kemudian mengubah teks pesan menjadi ucapan	<b>Berhasil</b>

View Message ketika membuka Aplikasi Pembaca Pesan	Menampilkan pesan yang diterima dari Telegram dan WhatsApp	Aplikasi menampilkan pesan yang diterima dari Telegram dan WhatsApp	<b>Berhasil</b>
			
Menekan tombol Sound pada tiap pesan yang diterima	Pesan teks dikonversi menjadi ucapan berdasarkan teks pilihan pesan	Aplikasi mengkonversi teks pesan berdasarkan teks pesan yang dipilih	<b>Berhasil</b>
Delete Pesan	Memilih pesan yang akan dihapus	Aplikasi memilih pesan yang akan dihapus berdasarkan beberapa pilihan hapus	<b>Berhasil</b>
	berdasarkan beberapa pilihan hapus		

<i>Input Singkatan</i>	Memasukkan kata singkatan dan kepanjangan pada dialog yang ditampilkan dan mendapatkan kata dari database	Aplikasi dapat memasukkan kata singkatan dan kepanjangan pada dialog serta mendapatkan kata dari database 	<b>Berhasil</b>
------------------------	---	--	-----------------

Dari hasil pengamatan pada tabel 4.4, aplikasi menerima pesan dari Telegram dan WhatsApp tanpa membuka aplikasi Pembaca Pesan berjalan dengan baik sehingga *user* akan menerima setiap pesan yang masuk. Aplikasi menerima notifikasi pesan dari Telegram dan WhatsApp kemudian mengubah teks pesan menjadi suara berjalan dengan baik sehingga *user* dapat mendengarkan isi pesan yang telah diterima. Aplikasi menampilkan pesan yang diterima dari Telegram dan WhatsApp berjalan dengan baik.

Aplikasi mengkonversi teks pesan berdasarkan teks pesan yang dipilih berjalan dengan baik sehingga pengguna dapat mendengarkan ulang pesan ketika tidak memiliki waktu untuk membaca dan bagi peneliti sebagai hasil validasi untuk teks yang telah dikonversi. Aplikasi memilih pesan yang akan dihapus berdasarkan pilihan yang ada berjalan dengan baik sehingga *user* dapat manajemen data yang telah diterima tidak terlalu besar. Serta aplikasi dapat memasukkan kata singkatan dan kepanjangan yang nantinya akan digunakan dalam perbandingan kata sebagai *output*.

Pengujian yang telah dilakukan di beberapa *brand* perangkat yang berbeda tidak ditemukan perbedaan *output* yang dihasilkan,

ini dikarenakan aplikasi menggunakan *Text-to-Speech Engine* yang sama yaitu Google. Hanya terdapat perbedaan letak Akses Notifikasi untuk mengaktifkan Aplikasi Pembaca Pesan, ini dikarenakan kustomisasi aplikasi operasi android dari *brand* itu sendiri.

Pengujian hasil dikelompokkan menjadi 4 skenario pengujian. *Pertama*, pengujian pesan yang sama dari setiap *brand smartphone* yang berbeda. *Kedua*, yaitu pengujian berdasarkan nama dan gelar. *Ketiga*, yaitu pengujian berdasarkan singkatan dengan huruf kapital. *Keempat*, yaitu pengujian berdasarkan akronim. Hasilnya, pengujian dinyatakan berhasil.

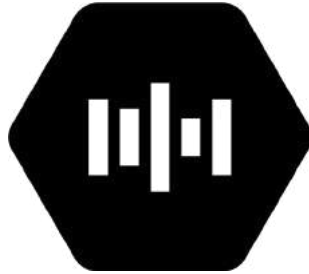
Dalam pengujian yang telah dilakukan di antaranya semua kata baku yang ada di dalam kamus besar bahasa Indonesia (KBBI), angka, simbol, tanggal, waktu, beberapa nama produk, penyebutan satuan bilangan sampai dengan bilangan mencapai satuan kuadtriliun, beberapa singkatan sesuai dengan ejaan yang disempurnakan serta kata singkatan yang terdapat dalam *database* singkatan.

Namun, satu pesan pengujian dinyatakan tidak sesuai karena kata [atm bca] merupakan singkatan dengan menggunakan huruf kecil dan kata tersebut tidak terdapat dalam *database* kata-kata khusus maupun dalam KBBI. Singkatan akan terbaca jika menggunakan huruf kapital [ATM BCA] dengan menyebutkan tiap huruf yang ada sesuai penulisan singkatan nama resmi lembaga. Kata singkatan yang belum dimasukkan ke dalam *database* tidak terbaca baik dikarenakan belum memiliki maksud kata. Setelah melakukan pengujian kembali pada kata yang tidak

terbaca dengan baik seperti kata [a-tm b-ca] menjadi [A-T-M B-C-A] membuat aplikasi dapat mengenali kata sehingga pesan dapat dibaca dengan baik.







## BAGIAN KEEMPAT

### Kesimpulan

Aplikasi pembaca pesan *text-to-speech* berjalan di *operating aplikasi* android yang dibuat memakai Android Studio dengan bahasa pemrograman *Java* dan menggunakan aplikasi manajemen *database SQLite*. Aplikasi pembaca pesan *text to speech* dapat menerima *pack*, *tittle* dan *text* dari Telegram dan WhatsApp sebagai masukan kemudian teks dikonversi menjadi suara sebagai keluaran.

Aplikasi pembaca pesan *text to speech* dengan turunan dari *Notification Listener Service* mampu menerima langsung pesan setiap pemberitahuan dari pesan Telegram dan WhatsApp yang mempunyai batasan 1024 karakter per setiap pesan dan akan ditampilkan pada layar *smartphone* dengan isi pesan sebagai masukan dan langsung dikonversi menjadi suara yang membuatnya bekerja di belakang layar tanpa harus membuka aplikasi utama. Dengan fungsi antrean *QUEUE\_ADD* pada *Google Text-to-Speech API* memungkinkan pesan yang diterima akan diproses satu per satu.

Aplikasi pembaca pesan *text to speech* menyediakan masukan kata singkatan dan kata yang dimaksud jika kata tidak sesuai dengan yang diucapkan. Aplikasi akan membandingkan kata singkatan yang telah dimasukkan ke dalam *database* dan membuat aplikasi mengucapkan kata yang dimaksud sesuai dengan kata singkatan tersebut.

Aplikasi ini bersifat *case sensitive* terhadap singkatan dan akronim sesuai dengan *input* dari pengguna. Aplikasi juga mempunyai batasan dalam membedakan antara singkatan kata maupun akronim kata.

Adapun dalam rangka pengembangan berkelanjutan aplikasi tersebut, penulis memiliki beberapa saran bagi pihak-pihak yang berminat ataupun memang yang *concern* di bidang ini, di antaranya: *Pertama*, aplikasi pembaca pesan dapat dikembangkan agar bisa dijalankan *smartphone* pada *operating aplikasi* selain Android seperti IOS, *Windows Phone*, dan sebagainya. *Kedua*, mengembangkan aplikasi dengan keluaran suara yang berbeda tergantung dengan teks yang tertulis seperti teks dalam bahasa Inggris, bahasa Arab, bahasa Jepang dan beberapa bahasa lainnya. *Ketiga*, mengembangkan aplikasi agar dapat ditambahkan fitur seperti menu *pop-up* untuk menambahkan kata singkatan ke dalam *library* aplikasi pembaca pesan dari sumber aplikasi yang lain, seperti aplikasi Telegram atau WhatsApp. *Keempat*, mengembangkan aplikasi yang sebaliknya, yaitu *Speech-to-Text*.

## DAFTAR PUSTAKA

- Andika, D. 2017. "Pengertian Flowchart". *it-jurnal*. Diakses 5 November 2020. <https://www.it-jurnal.com/pengertian-flowchart/>
- Arman, A. 2004. *Konversi dari Teks ke Ucapan*.
- Pusparisa, Yosepha. 2020. "Pengguna Smartphone Diperkirakan Mencapai 89% Populasi pada 2025". *databoks.katadata*. Diakses 7 November 2020. <https://databoks.katadata.co.id/datapublish/2020/09/15/pengguna-smartphone-diperkirakan-mencapai-89-populasi-pada-2025>.
- Anonim. 2020. "Inilah Urutan Versi Android dari Awal Hingga Terbaru". *decoding*. Diakses 7 November 2020. <http://www.dicoding.com/blog/urutan-versi-android/>.
- Djuandi, F. 2011. "Menggunakan Database di Android.". [www.tobuku.com](http://www.tobuku.com).
- Dutoit, T. 1997. *An Introduction Text-to-Speech Synthesis, Illustrated, ed*, Springer Science & Business Media.
- Herlawati, Widodo Pudjo, P. 2011. *Menggunakan UML*. Bandung: Informatika.
- Hillsboro, P.V., Sunnyvale, T.F.M. 2004. *Client/Server Architecture for Text-to-Speech Synthesis*.
- Irawan, A. 2014. *Implementasi Speech to Text dan Text to Speech Pada Aplikasi Traveling Conservations Assistance Berbasis Mobile*. Skripsi. Universitas Komputer Indonesia.
- Jeffrey, S. 2014. *WhatsApp vs Line vs Telegram: the best instant messenger apps compared - TorrApk [WWW Document]*. <https://www.torrapk.com/en/blog/apps-comparison/whatsapp-vs-line-vs-telegram> (accessed 11.23.17).

- Marius, Sapto. 2014. *Profil Pengguna Internet Indonesia*. PUSKAKOM.
- Mullennix, J. 2010. *Computer Synthesized Speech Technologies*. New York: Medical Information Science Reference.
- Pelton, G. E. 1993. *Voice Processing*. New York: McGraw-Hill Education.
- Anonim. 2015. "Smartphone Lagi, Penyebab Kecelakaan di Jalan". [plazainformasi.jogjaprov.go.id](http://plazainformasi.jogjaprov.go.id).
- Pramono, J. 2013. "Rancang Bangunan Aplikasi Text to Speech Sebagai Alat Bantu Pembelajaran Bahasa Inggris". Naskah Publikasi. AMIKOM.
- Prasetyo, D. 2006. *Aplikasi Pensintesa Ucapan Berbahasa Indonesia Sebagai Pembaca SMS*. Semarang: Fakultas Teknik, Universitas Diponegoro.
- Rahmawati, H. 2011. *Penggunaan Bahasa Indonesia dalam Bahasa SMS*.
- Safaat, H., N. 2012. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Salman, W., Puspita, B. 2017. *Perancangan Aplikasi Pengaduan Masyarakat pada Dashboard Smart City Berbasis Web dan Android*. Makassar: Universitas Hasanuddin.
- Shalahudin, M. 2010. *Java di WEB*. Informatika.
- Syaputra, R. 2017. "Modul Programming Android dengan Android Studio - 1". <http://www.academia.edu/9990291>.
- Taylor, P. 2009. *Text to Speech Synthesis*. Cambridge.
- Toma S,A., Tarsa G.I . 2010. *A TD-PSOLA Based Method for*

*Speech Synthesis and Compression*. IEEE.

Wundari, B. 2009. *Aplikasi Text-to-Speech Dengan Metode Unit Selection Synthesis Untuk Bahasa Indonesia*. Jakarta: Univ. Indonesia.

Yunus, R. D. 2007. *Aplikasi Text-to-Speech Berbahasa Indonesia sebagai Pembaca SMS*. Semarang: Universitas Stikubank.

## LAMPIRAN

1. Pengujian yang dilakukan di *device* Xiaomi MI A2

**Tabel Lampiran 1 Tabel Hasil Membaca Pesan di Xiaomi MI A2**

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	Aplikasi ini menggunakan google text to speech	[ap-li-ka-si i-ni meng-gu-nakan goo-gle text to-speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text-to speech]	Berhasil
2	;	[ti-tik ko-ma]	[ti-tik ko-ma]	[ti-tik ko-ma]	Berhasil
3	Tolong di transfer di ATM BCA	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	Berhasil
4	Tolong di transfer di atm bca	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di A-T-M B-C-A]	Tidak-berhasil
5	Jgn lah prg ke kos	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	Berhasil



6	081242918588	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du- a em-pat du-a sem-bi-lan sa- tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa- tu du-a em-pat du-a sem-bi-lan sa-tu de-la- pan li-ma de-la-pan de-la-pan]	Berhasil
7	RT 04 RW 02 blok  AB 1 no 29	[er-te ko-song em-pat er- we ko-song du-a blok a- be sa-tu no-mor du-a sem-bi-lan]	[er-te ko-song em-pat er-we ko-song du-a blok a-be sa-tu no-mor du-a sem-bi-lan]	[er-te ko-song em-pat er-we nol du-a blok a- be sa-tu no-mor du-a sem-bi-lan]	Berhasil
8	12.000.000 IDR	[du-a be-las ju-ta ru-pi-ah]	[du-a be-las ju-ta ru-pi-ah]	[du-a be-las ju-ta ru-pi-ah]	Berhasil
9	Macbeth, Vans, Baker, Fallen, Vegan	[mac-beth vans, ba-ker, fa-len, ve-gan]	[mac-beth vans, ba-ker, fa-len, ve-gan]	[mac-beth vans, ba-ker,fa-len, ve-gan]	Berhasil

10	Kulupai kalo mau ka ke kampus	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	Berhasil
11	Dimohon kepada seluruh anggota untuk menghadiri rapat yang akan dilaksanakan pada hari Jumat, 22 Januari 2021	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	Berhasil
12	datang ki teman teman sekalian	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	Berhasil

13	Jurnal ini dikonversi ke ucapan menggunakan perangkat lunak penerjemah otomatis bukan penerjemah manusia	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	Berhasil
14	Eh hai apa kabar?	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	Berhasil

2. Pengujian yang dilakukan di *device* Samsung A7 (A750)

**Tabel Lampiran 2 Tabel Hasil Membaca Pesan di  
Samsung A7 (A750)**

No.	Input	Output Telegram	Output WhatsApp	Output yang diing-inkan	Hasil Uji Coba
1	Aplikasi ini menggunakan google text to speech	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	Berhasil
2	;	[ti-tik ko-ma]	[ti-tik ko-ma]	[ti-tik ko-ma]	Berhasil
3	Tolong di transfer di-ATM BCA	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	Berhasil
4	Tolong di transfer di atm bca	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di A-T-M B-C-A]	Tidak berhasil
5	Jgn lah prg ke kos	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	Berhasil
6	081242918588	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	Berhasil

7	RT 04 RW 02 blok AB 1 no 29	[er-te ko- song em- pat er- we ko-song du-a blok a- be sa-tu no-mor du-a sem- bi-lan]	[er-te ko-song em-pat er-we ko-song du-a blok a-be sa- tu no-mor du-a sem-bi- lan]	[er-te ko- song em-pat er-we nol du-a blok a- be sa-tu no-mor du-a sem-bi-lan]	Ber- hasil
---	-----------------------------------	---	--	--	---------------

8	12.000.000 IDR	[du-a be-las ju-ta ru-pi- ah]	[du-a be-las ju-ta ru-pi-ah]	[du-a be-las ju-ta ru-pi- ah]	Berha- sil
9	Macbeth, Vans, Baker, Fallen, Veg- an	[mac-beth vans, ba- ker,fa-len, ve-gan]	[mac-beth vans, ba-ker, fa-len, ve-gan]	[mac-beth vans, ba- ker, fa-len, ve-gan]	Berha- sil
10	Kulupai kalo mau ka ke kampus	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	Berha- sil
11	Dimohon ke- pada seluruh anggota untuk meng- hadiri rapat yang akan dilaksanakan pada hari Ju- mat, 22 Janu- ari 2021	[di-mo-hon ke-pa-da se- lu-ruh ang- go-ta un-tuk meng-ha- di-ri ra-pat yang a-kan di-lak-sa- na-kan pa-da ha-ri jum-at tang- gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se- lu-ruh ang- go-ta un-tuk meng- ha-di- ri ra-pat yang a-kan di-lak- sa-na-kan pa- da ha-ri jum- at tang-gal du-a pu-luh du-a ja-nu-a- ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se- lu-ruh ang- go-ta un-tuk meng-ha- di-ri ra-pat yang a-kan di-lak-sa- na-kan pa-da ha-ri jum- at tang-gal du-a pu-luh du-a ja-nu- a-ri du-a ri-bu du-a pu-luh sa- tu]	Berha- sil

12	datang ki teman teman sekalian	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	Berhasil
13	Jurnal ini dikonversi ke ucapan menggunakan perangkat lunak penerjemah otomatis bukan penerjemah manusia	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	Berhasil
14	Eh hai apa kabar?	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	Berhasil

3. Pengujian yang dilakukan di *device* OPPO A7

Tabel Lampiran 3 Tabel Hasil Membaca Pesan di OPPO A7

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	Aplikasi ini menggunakan google text to speech	[ap-li-ka-si i-ni menggunakan goo-gle text to speech]	[ap-li-ka-si i-ni menggunakan goo-gle text to speech]	[ap-li-ka-si i-ni menggunakan goo-gle text to speech]	Berhasil
2	;	[ti-tik ko-ma]	[ti-tik ko-ma]	[ti-tik ko-ma]	Berhasil
3	Tolong di transfer di-ATM BCA	[to-long di transfer di A-T-M B-C-A]	[to-long di transfer di A-T-M B-C-A]	[to-long di transfer di A-T-M B-C-A]	Berhasil
4	Tolong di transfer di atm bca	[to-long di transfer di a-tm b-ca]	[to-long di transfer di a-tmb-ca]	[to-long di transfer di A-T-M B-C-A]	Tidak berhasil
5	Jgn lah prg ke kos	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	Berhasil

6	081242918588	[ko-song de-la-pan sa-tu du-a empat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a empat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a empat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	Berhasil
7	RT 04 RW 02 blok AB 1 no 29	[er-te ko-song empat er-we ko-song du-a blok a-be sa-tu no-mor du-a sem-bi-lan]	[er-te ko-song empat er-we ko-song du-a blok a-be sa-tu no-mor du-a sem-bi-lan]	[er-te ko-song empat er-we nol du-a blok a-be sa-tu no-mor du-a sem-bi-lan]	Berhasil
8	12.000.000 IDR	[du-a belas ju-ta ru-pi-ah]	[du-a belas ju-ta ru-pi-ah]	[du-a belas ju-ta ru-pi-ah]	Berhasil
9	Macbeth, Vans, Baker, Fallen, Vegan	[mac-beth vans, ba-ker, fa-len, ve-gan]	[mac-beth vans, ba-ker, fa-len, ve-gan]	[mac-beth vans, ba-ker, fa-len, ve-gan]	Berhasil



10	Kulupai kalo mau ka ke kampus	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam-pus]	Berhasil
11	Dimohon kepada seluruh anggota untuk menghadiri rapat yang akan dilaksanakan pada hari Jumat, 22 Januari 2021	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng-ha-di-ri ra-pat yang a-kan di-lak-sa-na-kan pa-da ha-ri jum-at tang-gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	Berhasil
12	datang ki teman teman sekalian	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	[da-tang ki te-man te-man se-ka-li-an]	Berhasil

13	Jurnal ini dikonversi ke ucapan menggunakan perangkat lunak penerjemah otomatis bukan penerjemah manusia	[jurnal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver-si ke u-ca-pan meng-gu-na-kan pe-rang-kat lu-nak pe-ner-je-mah o-to-ma-tis bu-kan pe-ner-je-mah ma-nu-si-a]	Berhasil
14	Eh hai apa kabar?	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	Berhasil

4. Pengujian yang dilakukan di *device* Realme 6 Pro  
**Tabel Lampiran 4 Tabel Hasil Membaca Pesan di Realme 6 Pro**

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	Aplikasi ini menggunakan google text to speech	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text-to speech]	Berhasil
2	;	[ti-tik ko-ma]	[ti-tik ko-ma]	[ti-tik ko-ma]	Berhasil
3	Tolong di transfer di-ATM BCA	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	Berhasil
4	Tolong di transfer di-atm bca	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di A-T-M B-C-A]	Tidak berhasil
5	Jgn lah prg ke kos	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	Berhasil
6	081242918588	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	Berhasil

7	RT 04 RW 02 blok AB 1 no 29	[er-te ko- song empat er- we ko-song du-a blok a- be sa-tu no-mor du-ase- bi-lan]	[er-te ko- song empat er- we k o - s o n g du-a blok a-be sa-tu n o - m o r du-a sem- bi-lan]	[er-te ko- song empat er-we nol du-a blok a- be sa-tu no-mor du-a sem-bi-lan]	Berhasil
---	-----------------------------------	--	--	---	----------

8	12.000.000 IDR	[du-a be- las ju-ta ru-pi-ah]	[du-a be-las ju-ta ru-pi- ah]	[du-a be-las ju-ta ru-pi- ah]	Berhasil
9	Macbeth, Vans, Baker, Fallen, Vegan	[mac-beth vans, ba- ker, fa-len, ve-gan]	[mac-beth vans, ba- ker, fa-len, ve-gan]	[mac-beth vans, ba- ker, fa-len, ve-gan]	Berhasil
10	Kulupai kalo mau ka ke kampus	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	Berhasil
11	Dimohon ke- pada seluruh anggota untuk menghadiri rapat yang akan dilak- sanakan pada hari Jumat, 22 Januari 2021	[di-mo- hon ke- pa-da se-lu-ruh ang-go-ta un-tuk meng-ha- di-ri ra- pat yang a-kan di- lak-sa-na- kan pa-da ha-ri jum- at tang-gal du-a pu- luh du-a ja-nu-a-ri du-a ri-bu du-a pu- luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng- ha- di-ri ra-pat yang a-kan di-lak-sa- na-kan pa-da ha- ri jum-at tang-gal du-a pu- luh du-a ja-nu-a-ri du-a ri-bu du-a pu- luh sa-tu]	[di-mo-hon ke-pa-da se- lu-ruh ang- go-ta un-tuk meng-ha- di-ri ra-pat yang a-kan di-lak-sa-na- kan pa-da ha-ri jum- at tang-gal du-a pu-luh du-a ja-nu- a-ri du-a ri- bu du-a pu- luh sa-tu]	Berhasil

12	datang ki teman teman sekalian	[da-tang ki te-man te-man se- ka-li-an]	[da-tang ki te-man te-man se- ka-li-an]	[da-tang ki te-man te- man se-ka- li-an]	Berhasil
13	Jurnal ini dikonversi ke ucapan menggunakan perangkat lunak pener- jemah otom- atis bukan penerjemah- manusia	[jur-nal i-ni di- kon-ver-si ke u-ca- pan meng- gu-na-kan pe-rang- kat lu-nak pe-ner-je- mah o-to- ma-tis bu-kan pe-ner-je- mah ma- nu-si-a]	[jur-nal i-ni di-kon-ver- si ke u-ca- pan meng- gu-na-kan pe-rang-kat lu-nak pe- ner-je-mah o-to-ma-tis bu-kan pe- ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver- si ke u-ca- pan meng- gu-na-kan pe-rang-kat lu-nak pe- ner-je-mah o-to-ma-tis bu-kan pe- ner-je-mah ma-nu-si-a]	Berhasil
14	Eh hai apa kabar?	[eh hai a-pa ka- bar]	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	Berhasil

5. Pengujian yang dilakukan di *device* Nokia 3

Tabel Lampiran 5 Tabel Hasil Membaca Pesan di Nokia 3

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	Aplikasi ini menggunakan google text to speech	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	[ap-li-ka-si i-ni meng-gu-na-kan goo-gle text to speech]	Berhasil
2	;	[ti-tik ko-ma]	[ti-tik ko-ma]	[ti-tik ko-ma]	Berhasil
3	Tolong di transfer di ATM BCA	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	[to-long di trans-fer di A-T-M B-C-A]	Berhasil
4	Tolong di transfer di atm bca	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di a-tm b-ca]	[to-long di trans-fer di A-T-M B-C-A]	Tidak berhasil
5	Jgn lah prg ke kos	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	[ja-ngan lah per-gi ke kos]	Berhasil
6	081242918588	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[k o - s o n g de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	[ko-song de-la-pan sa-tu du-a em-pat du-a sem-bi-lan sa-tu de-la-pan li-ma de-la-pan de-la-pan]	Berhasil

7	RT 04 RW 02 blok AB 1 no 29	[er-te ko- song empat er- we ko-song du-a blok a- be sa-tu no-mor du-a sem-bi-lan]	[er-te ko- song empat er- we k o - s o n g du-a blok a-be sa- tu no-mor du-a sem- bi-lan]	[er-te ko- song empat er- we nol du-a blok a- be sa-tu no- mor du-a sem-bi-lan]	Berhasil
8	12.000.000 IDR	[du-a be- las ju-ta ru-pi-ah]	[du-a be-las ju-ta ru-pi- ah]	[du-a be-las ju-ta ru-pi- ah]	Berhasil
9	Macbeth, Vans,Baker, Fallen, Vegan	[mac-beth vans, ba- ker, fa-len, ve-gan]	[mac-beth vans, ba- ker, fa- len, ve-gan]	[mac-beth vans, ba- ker,fa-len, ve-gan]	Berhasil
10	Kulupai kalo mau ka ke kampus	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	[ku-lu-pa-i ka-lo ma-u ka ke kam- pus]	Berhasil
11	Dimohon ke- pada seluruh anggota un- tuk menghad- iri rapat yang akan dilak- sanakan pada hari Jumat, 22 Januari 2021	[di-mo-hon ke-pa-da se- lu-ruh ang-go-ta un-tuk meng-ha- di-ri ra-pat yang a-kan di-lak-sa- na- kan pa-da ha- ri jum-at tang-gal du-a pu- luh du-a ja-nu-a-ri du-a ri-bu du-a pu- luh sa-tu]	[di-mo-hon ke-pa-da se-lu-ruh ang-go-ta un-tuk meng- ha- di-ri ra-pat yang a-kan di-lak-sa- na-kan pa-da ha- ri jum-at tang- gal du-a pu-luh du-a ja-nu-a-ri du-a ri-bu du-a pu-luh sa-tu]	[di-mo-hon ke-pa-da se- lu-ruh ang-go-ta un-tuk meng-ha- di-ri ra-pat yang a-kan di-lak-sa- na-kan pa-da ha- ri jum- at tang-gal du-a pu- luh du-a ja-nu-a-ri du-a ri- bu du-a pu- luh sa-tu]	Berhasil

12	datang ki teman teman sekalian	[da-tang ki te-man te- man se-ka- li-an]	[da-tang ki te-man te- man se-ka- li-an]	[da-tang ki te-man te- man se-ka- li-an]	Berhasil
13	Jurnal ini dikonversi ke ucapan meng- gunakan perangkat lunak pener- jemah otom- atis bukan penerjemah manusia	[jur-nal i- ni di-kon- ver-si ke u-ca-pan meng-gu- na-kan pe- rang-kat lu-nak pe- ner-je-mah o-to-ma- tis bu-kan pe- ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver- si ke u-ca- pan meng- gu-na-kan pe-rang-kat lu-nak pe- ner-je-mah o-to-ma-tis bu-kan pe- ner-je-mah ma-nu-si-a]	[jur-nal i-ni di-kon-ver- si ke u-ca- pan meng- gu-na-kan pe-rang-kat lu-nak pe- ner-je-mah o-to-ma-tis bu-kan pe- ner-je-mah ma-nu-si-a]	Berhasil
14	Eh hai apa kabar?	[eh hai a-pa ka- bar]	[eh hai a-pa ka-bar]	[eh hai a-pa ka-bar]	Berhasil



## 6. Pengujian berdasarkan nama dan gelar

Tabel Lampiran 6 Pengujian berdasarkan nama dan gelar

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	W.R. Supratman	[W R sup-rat-man]	[W R sup-rat-man]	[W R sup-rat-man]	Berhasil
2	R.A. Kartini	[R A kar-ti-ni]	[R A kar-ti-ni]	[R A kar-ti-ni]	Berhasil
3	H. Agus	[ha-ji a-gus]	[ha-ji a-gus]	[ha-ji a-gus]	Berhasil
4	M.B.A	[M B A]	[M B A]	[M B A]	Berhasil
5	M.Sos.	[M Sos]	[M Sos]	[M Sos]	Berhasil
6	Jend. Sudirman	[jen-de-ral su-dir-man]	[jen-de-ral su-dir-man]	[jen-de-ral su-dir-man]	Berhasil

## 7. Pengujian berdasarkan singkatan huruf kapital

**Tabel Lampiran 7 Pengujian berdasarkan singkatan huruf kapital**

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	NKRI	[ne-ga-ra ke-sa-tu-an re-pub-lik in-do-ne-sia]	[ne-ga-ra ke-sa-tu-an re-pub-lik in-do-ne-sia]	[ne-ga-ra ke-sa-tu-an re-pub-lik in-do-ne-sia]	Berhasil
2	ABRI	[ang-ka-tan ber-sen-ja-ta re-pub-lik in-do-ne-sia]	[ang-ka-tan ber-sen-ja-ta re-pub-lik in-do-ne-sia]	[ang-ka-tan ber-sen-ja-ta re-pub-lik in-do-ne-sia]	Berhasil
3	BMKG	[ba-dan me-teo-ro-lo-gi kli-ma-to-lo-gi geo-fi-si-ka]	[ba-dan me-teo-ro-lo-gi kli-ma-to-lo-gi geo-fi-si-ka]	[ba-dan me-teo-ro-lo-gi kli-ma-to-lo-gi geo-fi-si-ka]	Berhasil
4	SMA	[se-ko-lah me-ne-ngah a-tas]	[se-ko-lah me-ne-ngah a-tas]	[se-ko-lah me-ne-ngah a-tas]	Berhasil
5	KTP	[kar-tu tan-da pen-du-duk]	[kar-tu tan-da pen-du-duk]	[kar-tu tan-da pen-du-duk]	Berhasil
6	UGM	[uni-ver-si-tas ga-djah ma-da]	[uni-ver-si-tas ga-djah ma-da]	[uni-ver-si-tas ga-djah ma-da]	Berhasil

## 8. Pengujian berdasarkan akronim

Tabel Lampiran 8 Pengujian berdasarkan akronim

No.	Input	Output Telegram	Output WhatsApp	Output yang diinginkan	Hasil Uji Coba
1	iptek	[il-mu pe-nge-ta-hu-an tek-no-lo-gi]	[il-mu pe-nge-ta-hu-an tek-no-lo-gi]	[il-mu pe-nge-ta-hu-an tek-no-lo-gi]	Berhasil
2	PERSIJA	[per-sa-tu-an se-pak bo-la ja-kar-ta]	[per-sa-tu-an se-pak bo-la ja-kar-ta]	[per-sa-tu-an se-pak bo-la ja-kar-ta]	Berhasil
3	Bawaslu	[ba-dan pe-nga-was pe-mi-lu]	[ba-dan pe-nga-was pe-mi-lu]	[ba-dan pe-nga-was pe-mi-lu]	Berhasil
4	Sulsel	[su-la-we-si se-la-tan]	[su-la-we-si se-la-tan]	[su-la-we-si se-la-tan]	Berhasil
5	tilang	[buk-ti pe-lang-ga-ran]	[buk-ti pe-lang-ga-ran]	[buk-ti pe-lang-ga-ran]	Berhasil

# **PANDUAN PENGGUNAAN APLIKASI PEMBACA PESAN**

## PANDUAN PENGUNAAN



### APLIKASI PEMBAKA PESAN

#### 1. Pengantar

Aplikasi Pembaca Pesan adalah sistem pembaca pesan yang akan menerima teks pesan dari Telegram dan WhatsApp kemudian mengubahnya menjadi ucapan dengan menggunakan *google API text-to-speech* bahasa Indonesia. Sistem *Text-To-Speech* Bahasa Indonesia ini berbasis android yang mampu mengucapkan kata singkatan dengan jelas. Sistem juga melakukan interaksi pada user dalam proses penyempurnaan database kata singkatan sehingga sistem dapat mengenali kata singkatan lebih banyak. Selain itu sistem ini dilengkapi dengan fitur manajemen pesan dan kontrol ucapan, seperti menghapus dan mendengarkan kembali pesan yang telah diterima, memasukkan singkatan untuk menghasilkan keluaran yang baik, serta dapat mengaktifkan ucapan dan menonaktifkan ucapan saat tidak ingin mendengarkan pesan.

Berikut spesifikasi untuk menjalankan aplikasi Pembaca Pesan:

- Android versi minimum versi 8.0
- Bahasa *smartphone* atau bahasa keluaran *text-to-speech*, bahasa Indonesia
- Telah mempunyai aplikasi Telegram dan/atau Whatsapp

#### 2. Membuka Aplikasi Pembaca Pesan

Jika aplikasi sudah dipasang, silahkan tap icon Aplikasi Pembaca Pesan pada *smartphone* anda.



Gambar 2. 1. Icon Aplikasi Pembaca Pesan

Panduan Penggunaan Aplikasi Pembaca Pesan | 1

#### DAFTAR ISI

COVER .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iii
1. Pengantar .....	1
2. Membuka Aplikasi Pembaca Pesan .....	1
3. Tampilan Halaman Welcome Activity .....	2
4. Tampilan Halaman Home Screen .....	4
5. Fitur Aplikasi Pembaca Pesan .....	5
5.1. Tambah Singkatan .....	5
5.2. Hapus Pesan .....	7
5.3. Kontrol Ucapan .....	10
5.4. Mendengarkan Kembali Pesan Masuk .....	11
5.5. Shortcut Akses Notifikasi .....	12
5.6. Keluar dan Menonaktifkan Aplikasi .....	13
6. Penggunaan Aplikasi .....	14

Panduan Penggunaan Aplikasi Pembaca Pesan | ii

#### DAFTAR GAMBAR

Gambar 2. 1. Icon Aplikasi Pembaca Pesan .....	1
Gambar 3. 1. Tampilan Welcome Activity Slide 1 dan Slide 2 .....	2
Gambar 3. 2. Tampilan Welcome Activity Slide 3 dan Akses Notifikasi .....	3
Gambar 4. 1. Tampilan Home Screen Aplikasi Pembaca Pesan .....	4
Gambar 5. 1. 1. Langkah-Langkah Menambah Kata Singkatan .....	5
Gambar 5. 1. 2. Kotak Dialog Masukkan Kata Singkatan .....	6
Gambar 5. 2. 1. Langkah-Langkah Menghapus Pesan Whatsapp atau Telegram .....	7
Gambar 5. 2. 2. Kotak Dialog Menghapus Pesan .....	8
Gambar 5. 2. 3. Langkah-Langkah dan Kotak Dialog Menghapus Pesan Tertentu .....	9
Gambar 5. 3. 1. Langkah-Langkah Kontrol Ucapan .....	10
Gambar 5. 4. 1. Mendengarkan Kembali Pesan .....	11
Gambar 5. 5. 1. Fitur Shortcut Akses Notifikasi .....	12
Gambar 5. 6. 1. Keluar dan Menonaktifkan Aplikasi .....	13
Gambar 6. 1. Tampilan Penerimaan Pesan .....	14

Panduan Penggunaan Aplikasi Pembaca Pesan | iii



Gambar 3. 2. Tampilan Welcome Activity Slide 3 dan Akses Notifikasi

Pada slide ke-3 halaman Welcome Activity, tap IZINKAN di pojok kiri bawah, kemudian anda akan diarahkan ke halaman Akses Notifikasi, anda disarankan mengizinkan Akses Notifikasi untuk aplikasi Pembaca Pesan. Setelah berhasil, kembali ke tampilan Welcome Activity, kemudian tap MULAI di pojok kanan bawah.

### 3. Tampilan Halaman Welcome Activity

Jika sudah berhasil membuka aplikasi Pembaca Pesan, akan muncul tampilan Welcome Activity berupa panduan singkat penggunaan aplikasi seperti gambar di bawah ini.



Gambar 3. 1. Tampilan Welcome Activity Slide 1 dan Slide 2

Panduan Penggunaan Aplikasi Pembaca Pesan | 3

Panduan Penggunaan Aplikasi Pembaca Pesan | 2

### 5.6. Keluar dan Menonaktifkan Aplikasi

Fitur ini bertujuan untuk keluar dari aplikasi dan menghentikan izin aplikasi untuk menerima dan mendengarkan pesan.



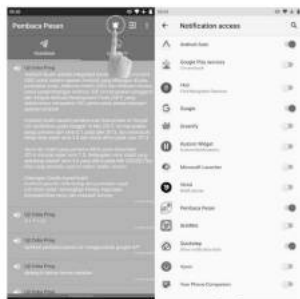
Gambar 5. 6. 1. Keluar dan Menonaktifkan Aplikasi

Tap di icon keluar di bagian atas kanan aplikasi untuk keluar.

Panduan Penggunaan Aplikasi Pembaca Pesan | 13

### 5.5. Shortcut Akses Notifikasi

Shortcut ini akan mengarahkan anda ke halaman akses notifikasi



Gambar 5. 5. 1. Fitur Shortcut Akses Notifikasi

Tap di icon lonceng di bagian atas kanan untuk mengakses halaman akses notifikasi.

Panduan Penggunaan Aplikasi Pembaca Pesan | 12

## 5. Fitur Aplikasi Pembaca Pesan

### 5.1. Tambah Singkatan

Fitur ini terdapat di menu options. Untuk pilihan Tambah Singkatan, tap menu dengan icon titik tiga di pojok kanan atas kemudian tap Tambah Singkatan.



Gambar 5. 1. 1. Langkah-Langkah Menambah Kata Singkatan

### 4. Tampilan Halaman Home Screen

Di aplikasi Pembaca Pesan, anda dapat menerima dan mendengarkan pesan yang masuk berupa pesan Telegram dan pesan Whatsapp. Pada halaman ini terdapat dua tab untuk penerimaan pesan, tab telegram untuk menampilkan pesan telegram dan tab whatsapp untuk menampilkan pesan whatsapp.



Gambar 4. 1. Tampilan Home Screen Aplikasi Pembaca Pesan

Panduan Penggunaan Aplikasi Pembaca Pesan | 5

Panduan Penggunaan Aplikasi Pembaca Pesan | 4

### 5.2. Hapus Pesan

Fitur ini terdapat di menu options. Untuk pilihan Hapus Pesan, tap menu dengan icon titik tiga di pojok kanan atas kemudian tap Hapus Pesan, kemudian akan muncul sub-menu Hapus Pesan Whatsapp untuk menghapus semua pesan Whatsapp dan Hapus Pesan Telegram untuk menghapus semua pesan Telegram.



Gambar 5. 2. 1. Langkah-Langkah Menghapus Pesan Whatsapp atau Telegram

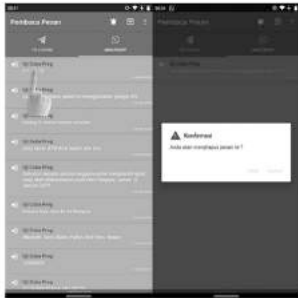


Gambar 5. 1. 2. Kotak Dialog Masukkan Kata Singkatan

Setelah muncul kotak dialog Masukkan Kata singkatan, anda di perkenankan mengisi form Kata Singkatan dan Kata yang Dimaksud, kemudian tap SIMPAN untuk menyimpan kata singkatan dan TIDAK untuk membatalkan.

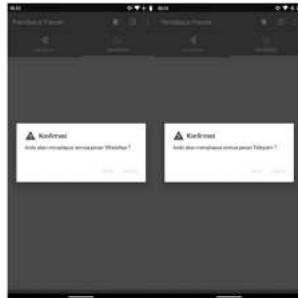
Panduan Penggunaan Aplikasi Pembaca Pesan | 7

Panduan Penggunaan Aplikasi Pembaca Pesan | 6



Gambar 5. 2. 3. Langkah-Langkah dan Kotak Dialog Menghapus Pesan Tertentu

Fitur untuk menghapus pesan tertentu juga bisa dilakukan dengan cara tahan tap dan tahan di pesan yang ingin dihapus, kemudian akan muncul kotak dialog Konfirmasi untuk menghapus pesan, tap HAPUS untuk menghapus dan TIDAK untuk membatalkan.



Gambar 5. 2. 2. Kotak Dialog Menghapus Pesan

Setelah tap Hapus Pesan Whatsapp atau Hapus Pesan Telegram, akan muncul kotak dialog Konfirmasi untuk menghapus pesan, tap HAPUS untuk menghapus dan TIDAK untuk membatalkan.

#### 5.4. Mendengarkan Kembali Pesan Masuk

Fitur ini berfungsi untuk mendengarkan pesan masuk yang tidak sempat terdengar atau hanya ingin mendengarkan kembali pesan.



Gambar 5. 4. 1. Mendengarkan Kembali Pesan

Untuk mendengarkan pesan tap di icon speaker.

#### 5.3. Kontrol Ucapan

Fitur ini terdapat di menu options. Untuk pilihan Kontrol Ucapan, tap menu dengan icon titik tiga di pojok kanan atas kemudian tap Kontrol Ucapan, kemudian akan muncul sub-menu Aktifkan Ucapan dan Matikan Ucapan.



Gambar 5. 3. 1. Langkah-Langkah Kontrol Ucapan

Sub-menu Aktifkan Ucapan untuk mengaktifkan ucapan pada saat pesan masuk dan Sub-menu Matikan Ucapan untuk menonaktifkan ucapan pada saat pesan masuk.



### 5.6. Keluar dan Menonaktifkan Aplikasi

Fitur ini bertujuan untuk keluar dari aplikasi dan menghentikan izin aplikasi untuk menerima dan mendergarkan pesan.



Gambar 5. 6. 1. Keluar dan Menonaktifkan Aplikasi

Tap di ikon keluar dibagian atas kanan aplikasi untuk keluar.

### 5.5. Shortcut Akses Notifikasi

Shortcut ini akan mengarahkan anda ke halaman akses notifikasi



Gambar 5. 5. 1. Fitur Shortcut Akses Notifikasi

Tap di ikon lonceng di bagian atas kanan untuk mengakses halaman akses notifikasi.

## Source Code

## ChatAdapter.java

```

1  package com.example.fiqhimaulana.pembacapan;
2
3  import android.annotation.SuppressLint;
4  import android.content.Context;
5  import android.content.DialogInterface;
6  import android.speech.tts.TextToSpeech;
7  import android.support.v7.app.AlertDialog;
8  import android.support.v7.widget.RecyclerView;
9  import android.view.LayoutInflater;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.ImageButton;
13 import android.widget.TextView;
14 import java.util.List;
15 import static com.example.fiqhimaulana.pembacapan.NotificationService.tts;
16
17 /**
18  * Created by FIQHI MAULANA.
19  */
20
21 public class ChatAdapter extends android.support.v7.widget.RecyclerView.Adapter<ChatAdapter.ViewHolder> {
22
23     private Context context;
24     private List <ChatModel> daftarchat;
25     TextToSpeech textToSpeech;
26     DataBase db;
27     List<KataSingkatan> kataSingkatan;
28     ParseSingkatan parser;
29
30     public ChatAdapter(Context context, List<ChatModel>daftarchat){
31         this.context=context;
32         this.daftarchat=daftarchat;
33
34         textToSpeech = NotificationService.getTts();
35
36         db = new DataBase(context);
37         kataSingkatan = db.getKata();
38         parser = new ParseSingkatan(context);
39     }
40
41     @Override
42     public ChatAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
43         Context context = parent.getContext();
44         LayoutInflater inflater = LayoutInflater.from(context);
45
46         View list = inflater.inflate(R.layout.barischat,parent,false);
47
48         ViewHolder viewHolder = new ViewHolder(list);
49
50         return viewHolder;
51     }
52
53     @Override
54     public void onBindViewHolder(ChatAdapter.ViewHolder holder, @SuppressLint("RecyclerView") final int position) {
55         final ChatModel model = daftarchat.get(position);
56         holder.pengirim.setText(model.getPengirim());
57         holder.pesan.setText(model.getPesan());
58         holder.datetime.setText(model.getDatetime());
59
60         String parsed = parser.parseSingkatan(model.getPesan());
61         parsed = parser.parseDatabase(parsed);
62         final String finalParsed = parsed;
63         holder.btnSuara.setOnClickListener(new View.OnClickListener() {
64             @Override
65             public void onClick(View v) {
66
67                 tts.speak(finalParsed,textToSpeech.QUEUE_ADD,null) ;
68             }
69         });
70
71         holder.itemView.setOnLongClickListener(new View.OnLongClickListener() {
72
73             @Override
74             public boolean onLongClick(View v) {
75
76                 AlertDialog.Builder alert = new AlertDialog.Builder(v.getContext());
77                 alert.setTitle("Konfirmasi");
78                 alert.setMessage("Anda akan menghapus pesan ini ?");
79                 alert.setIcon(R.drawable.ic_warning);
80
81

```

```

82         final AlertDialog.Builder yes = alert.setPositiveButton("HAPUS", new DialogInterface.OnClickListener() {
83             @Override
84             public void onClick(DialogInterface dialog, int which) {
85                 db.deleteChatById(model.getId());
86                 daftarchat.remove(position);
87                 notifyDataSetChanged();
88             }
89         });
90         alert.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {
91             @Override
92             public void onClick(DialogInterface dialog, int which) {
93                 dialog.cancel();
94             }
95         });
96         alert.show();
97         return false;
98     }
99 }
100 }
101 }
102 }
103
104 @Override
105 public int getItemCount() {
106     return daftarchat.size(); //jumlah data yang ditampilkan di RecyclerView
107 }
108 public class ViewHolder extends RecyclerView.ViewHolder{
109     TextView pengirim;
110     TextView pesan;
111     ImageButton btnSuara;
112     TextView datetime;
113     public ViewHolder(View itemView) {
114         super(itemView);
115         pengirim = (TextView) itemView.findViewById(R.id.text_pengirim);
116         pesan = (TextView) itemView.findViewById(R.id.text_pesan);
117         datetime = (TextView) itemView.findViewById(R.id.txt_datetime);
118         btnSuara = (ImageButton) itemView.findViewById(R.id.btnBunyi);
119     }
120 }
121 }

```

## ChatModel.java

```
1 package com.example.fiqhimaulana.pembacapan;
2
3 /**
4  * Created by FIQHI MAULANA.
5  */
6
7 public class ChatModel {
8     String id;
9     String pengirim;
10    String pesan;
11    String datetime;
12
13    public ChatModel({});
14    public ChatModel(String id, String pengirim, String pesan, String datetime) {
15        this.id = id;
16        this.pengirim = pengirim;
17        this.pesan = pesan;
18        this.datetime = datetime;
19    }
20
21    public String getId() {
22        return id;
23    }
24
25    public void setId(String id) {
26        this.id = id;
27    }
28
29    public String getPengirim() {
30        return pengirim;
31    }
32
33    public void setPengirim(String pengirim) {
34        this.pengirim = pengirim;
35    }
36
37    public String getPesan() {
38        return pesan;
39    }
40
41    public void setPesan(String pesan) {
42        this.pesan = pesan;
43    }
44
45    public String getDatetime() {
46        return datetime;
47    }
48    public void setDatetime(String datetime) {
49        this.datetime = datetime;
50    }
51 }
```

## DataBase.java

```

1 package com.example.fiqhmulana.pembacapan;
2
3 import android.content.ContentValues;
4 import android.content.Context;
5 import android.database.Cursor;
6 import android.database.sqlite.SQLiteDatabase;
7 import android.util.Log;
8 import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;
9 import java.util.ArrayList;
10 import java.util.HashMap;
11 import java.util.List;
12
13 /**
14  * Created by FIQIH MAULANA.
15  */
16
17 public class DataBase extends SQLiteAssetHelper {
18     private static final String TAG = DataBase.class.getSimpleName();
19     public static final String TELEGRAM_PACK = "org.telegram.messenger";
20     public static final String WHATSAPP_PACK = "com.whatsapp";
21
22     // Database Version
23     private static final int DATABASE_VERSION = 1;
24
25     // Database Name
26     private static final String DATABASE_NAME = "autospeech.db";
27
28     // Table chat
29     private static final String TABLE_CHAT = "chat";
30
31     //table singkatan
32     private static final String TABLE_SINGKATAN = "singkatan";
33
34     // kolom chat
35     private static final String KEY_ID = "id";
36     private static final String KEY_PENIRIM = "pengirim";
37     private static final String KEY_PESAN = "pesan";
38     private static final String KEY_PACKAGE = "pack";
39     private static final String KEY_DATETIME = "created_at";
40
41     //kolom singkatan
42     private static final String KEY_IDS = "id";
43     private static final String KEY_SINGKATAN = "word";
44     private static final String KEY KEPANJANGAN = "meaning";
45
46     public DataBase(Context context) {
47         super(context, DATABASE_NAME, null, DATABASE_VERSION);
48     }
49
50     public void addChat( String pengirim,String pesan,String pack,String datetime){
51         SQLiteDatabase db = this.getWritableDatabase();
52         ContentValues chat = new ContentValues();
53         chat.put(KEY_PENIRIM,pengirim);
54         chat.put(KEY_PESAN,pesan);
55         chat.put(KEY_PACKAGE,pack);
56         chat.put(KEY_DATETIME,datetime);
57
58         long id = db.insert(TABLE_CHAT,null,chat);
59         db.close();
60
61         Log.d(TAG, "Chat baru masuk ke database " + id);
62     }
63     /*
64     * tambahkan singkatan
65     */
66     public void addkata (String singkatan, String kepanjangan){
67         SQLiteDatabase db = this.getWritableDatabase();
68         ContentValues kata = new ContentValues();
69         kata.put(KEY_SINGKATAN, singkatan);
70         kata.put(KEY_KEPANJANGAN, kepanjangan);
71
72         long kta = db.insert(TABLE_SINGKATAN,null,kata);
73         db.close();
74         Log.d(TAG, "kata yang masuk " + kta);
75     }
76
77     public List<ChatModel> getChatDetails(String pack) {
78         HashMap<String, String> chat = new HashMap<String, String>();
79         String selectQuery = "SELECT * FROM " + TABLE_CHAT + " WHERE " + KEY_PACKAGE + " = '" + pack + "' ORDER BY " + KEY_ID + " DESC";
80
81         List<ChatModel> chatList = new ArrayList<>();
82         ChatModel model;
83         SQLiteDatabase db = this.getReadableDatabase();
84         Cursor cursor = db.rawQuery(selectQuery, null);
85         // Move to first row
86         cursor.moveToFirst();
87         for (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext()) {
88             // do what you need with the cursor here
89             model = new ChatModel(String.valueOf(cursor.getLong(0)), cursor.getString(1), cursor.getString(2), cursor.getString(4));
90
91             chatList.add(model);

```

```

92     }
93
94     cursor.close();
95     db.close();
96     // return chat
97     Log.d(TAG, "Mengambil chat dari sqlite: " + chatList.toString());
98
99     return chatList;
100 }
101
102 public List<KataSingkatan> getKata(){
103     HashMap<String, String> singkatan = new HashMap<String, String>();
104     String selectQuery = "SELECT * FROM " + TABLE_SINGKATAN ;
105     List<KataSingkatan> kataSingkatanList = new ArrayList<>();
106     KataSingkatan model;
107     SQLiteDatabase db = this.getReadableDatabase();
108     Cursor cursor = db.rawQuery(selectQuery,null);
109     cursor.moveToFirst();
110     for (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext()) {
111         model = new KataSingkatan(String.valueOf(cursor.getLong(0)),cursor.getString(1),cursor.getString(2));
112
113         kataSingkatanList.add(model);
114     }
115     cursor.close();
116     db.close();
117     Log.d(TAG, "Mengambil kata dari sqlite: " + kataSingkatanList.toString());
118     return kataSingkatanList;
119 }
120
121 public void deleteChat(String id) {
122     SQLiteDatabase db = this.getWritableDatabase();
123     // Delete All Rows
124     db.delete(TABLE_CHAT, null, null);
125     db.close();
126     Log.d(TAG, "Deleted all chat info from sqlite");
127 }
128
129 public void deleteChatById(String id) {
130     SQLiteDatabase db = this.getWritableDatabase();
131     // Delete Id
132     db.delete(TABLE_CHAT, KEY_ID + " = ?",new String[]{id});
133     db.close();
134     Log.d(TAG, "Delete Chat Id"+id);
135 }
136
137 public void deleteChatByPack(String pack) {
138     SQLiteDatabase db = this.getWritableDatabase();
139     // Delete Pack
140     db.delete(TABLE_CHAT, KEY_PACKAGE + " = ?",new String[]{pack});
141     db.close();
142     Log.d(TAG, "Delete Chat Pack"+pack);
143 }
144 }
145

```

## KataSingkatan.java

```
1 package com.example.fiqhimaulana.pembacapesan;
2
3 /**
4  * Created by FIQHI MAULANA.
5  */
6
7 public class KataSingkatan {
8     String ids ;
9     String singkatan;
10    String kepanjangan;
11
12    public KataSingkatan(){
13    }
14
15    public KataSingkatan (String id, String singkatan, String kepanjangan){
16        this.ids = ids;
17        this.singkatan = singkatan ;
18        this.kepanjangan = kepanjangan;
19    }
20
21    public String getIds(){
22        return ids;
23    }
24    public void setIds(String ids){
25        this.ids = ids;
26    }
27
28    public String getSingkatan(){
29        return singkatan;
30    }
31    public void setSingkatan(String singkatan){
32        this.singkatan = singkatan;
33    }
34
35    public String getKepanjangan(){
36        return kepanjangan;
37    }
38    public void setKepanjangan(String kepanjangan){
39        this.kepanjangan =kepanjangan;
40    }
41 }
42
```



## MainActivity.java

```

1  package com.example.fiqhinaulana.pembacapesan;
2
3  import android.content.DialogInterface;
4  import android.content.Intent;
5  import android.os.Bundle;
6  import android.support.design.widget.TabLayout;
7  import android.support.v4.app.Fragment;
8  import android.support.v4.app.FragmentManager;
9  import android.support.v4.app.FragmentPagerAdapter;
10 import android.support.v4.view.ViewPager;
11 import android.support.v7.app.AlertDialog;
12 import android.support.v7.app.AppCompatActivity;
13 import android.view.LayoutInflater;
14 import android.view.Menu;
15 import android.view.MenuItem;
16 import android.view.View;
17 import android.widget.EditText;
18 import android.widget.TableLayout;
19 import android.widget.Toast;
20 import java.util.ArrayList;
21 import java.util.List;
22
23 public class MainActivity extends AppCompatActivity {
24
25     private TabLayout tabLayout;
26     private ViewPager viewPager;
27     private PreferenceManager prefManager;
28
29     List<ChatModel> listChat = new ArrayList<>();
30     ChatAdapter mAdapter;
31
32     DataBase db;
33     TableLayout tab;
34
35     private int[] tabIcons = {
36         R.mipmap.icon_telegram,
37         R.mipmap.icon_whatsapp
38     };
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) {
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_mainmain);
44         prefManager = new PreferenceManager(this);
45
46         db = new DataBase(getApplicationContext());
47         mAdapter = new ChatAdapter(getApplicationContext(),listChat);
48         viewPager = (ViewPager) findViewById(R.id.viewpager);
49         setupViewPager(viewPager);
50         tabLayout = (TabLayout) findViewById(R.id.tabs);
51         tabLayout.setupWithViewPager(viewPager);
52         setTabIcons();
53         tab = (TableLayout) findViewById(R.id.tab);
54     }
55
56     @Override
57     public boolean onCreateOptionsMenu(Menu menu) {
58         getMenuInflater().inflate(R.menu.context_menu,menu);
59         return true;
60     }
61
62     @Override
63     public boolean onOptionsItemSelected(MenuItem item) {
64
65         switch (item.getItemId()){
66             case R.id.exitit:
67                 Toast.makeText(this, "Keluar dari Pembaca Pesan", Toast.LENGTH_SHORT).show();
68                 try {
69                     // clearing app data
70                     String packageName = getApplicationContext().getPackageName();
71                     Runtime runtime = Runtime.getRuntime();
72                     runtime.exec("pm clear "+packageName);
73                 }
74                 catch (Exception e) {
75                     e.printStackTrace();

```

```

76     }
77     break;
78     case R.id.notifsetting:
79         startActivity(new Intent("android.settings.ACTION_NOTIFICATION_LISTENER_SETTINGS"));
80         break;
81     case R.id.tts:
82         prefsManager.setIsTtsOn(true);
83         Toast.makeText(this, "Text-to-Speech Aktif", Toast.LENGTH_SHORT).show();
84         break;
85     case R.id.tts:
86         prefsManager.setIsTtsOn(false);
87         Toast.makeText(this, "Text-to-Speech Tidak Aktif", Toast.LENGTH_SHORT).show();
88         break;
89     case R.id.hapus_pesanW:
90         AlertDialog.Builder alert1 = new AlertDialog.Builder(this);
91         alert1.setTitle("Konfirmasi");
92         alert1.setMessage("Anda akan menghapus semua pesan WhatsApp ?");
93         alert1.setIcon(R.drawable.ic_warning);
94         alert1.setPositiveButton("HAPUS", new DialogInterface.OnClickListener() {
95             @Override
96             public void onClick(DialogInterface dialog, int which) {
97                 db.deleteChatByPack(db.WHATSAPP_PACK);
98                 listChat = new ArrayList<ChatModel>();
99                 mAdapter.notifyDataSetChanged();
100                setupViewPager(viewPager);
101                setTabIcons();
102            }
103        });
104        alert1.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {
105            @Override
106            public void onClick(DialogInterface dialog, int which) {
107                dialog.cancel();
108            }
109        });
110        alert1.create();
111        alert1.show();
112        break;
113     case R.id.hapus_pesanT:
114         AlertDialog.Builder alerttt = new AlertDialog.Builder(this);
115         alerttt.setTitle("Konfirmasi");
116         alerttt.setMessage("Anda akan menghapus semua pesan Telegram ?");
117         alerttt.setIcon(R.drawable.ic_warning);
118         alerttt.setPositiveButton("HAPUS", new DialogInterface.OnClickListener() {
119             @Override
120             public void onClick(DialogInterface dialog, int which) {
121                 db.deleteChatByPack(db.TELGRAM_PACK);
122                 listChat = new ArrayList<ChatModel>();
123                 mAdapter.notifyDataSetChanged();
124                 setupViewPager(viewPager);
125                 setTabIcons();
126             }
127        });
128        alerttt.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {
129            @Override
130            public void onClick(DialogInterface dialog, int which) {
131                dialog.cancel();
132            }
133        });
134        alerttt.create();
135        alerttt.show();
136        break;
137     case R.id.add_database_singkatan:
138         AlertDialog.Builder alertdbs = new AlertDialog.Builder(this);
139         LayoutInflater inflater = getLayoutInflater(); // mengkonvert view pada xml menjadi java objek
140         final View dialogView = inflater.inflate(R.layout.tambah_data_singkatan, null);
141         alertdbs.setView(dialogView);
142         alertdbs.setCancelable(true);
143         alertdbs.setTitle("Masukkan Kata Singkatan");
144         alertdbs.setPositiveButton("SIMPAN", new DialogInterface.OnClickListener() {
145             @Override
146             public void onClick(DialogInterface dialog, int which) {
147                 final EditText singkatan = (EditText) dialogView.findViewById(R.id.in_singkatan);
148                 final String textsingkatan = singkatan.getText().toString();
149                 final EditText kata = (EditText) dialogView.findViewById(R.id.in_kata);
150                 final String textkata = kata.getText().toString();
151                 db.addKata(textsingkatan, textkata);
152             }
153        });

```

```

154
155         alertdbs.setNegativeButton("TIDAK", new DialogInterface.OnClickListener() {
156             @Override
157                 public void onClick(DialogInterface dialog, int which) {
158                     dialog.cancel();
159                 }
160             });
161         alertdbs.create();
162         alertdbs.show();
163         break;
164     }
165     return super.onOptionsItemSelected(item);
166 }
167
168 private void setTabIcons(){
169     tablayout.getTabAt(0).setIcon(tabIcons[0]);
170     tablayout.getTabAt(1).setIcon(tabIcons[1]);
171 }
172 private void setupViewPager(ViewPager viewPager) {
173     ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());
174     adapter.addFragment(new Telegram(), "Telegram");
175     adapter.addFragment(new Whatsapp(), "Whatsapp");
176     viewPager.setAdapter(adapter);
177 }
178
179 class ViewPagerAdapter extends FragmentPagerAdapter {
180     private final List<Fragment> mFragmentManager = new ArrayList<>();
181     private final List<String> mFragmentManagerTitleList = new ArrayList<>();
182     public ViewPagerAdapter(FragmentManager manager) {
183         super(manager);
184     }
185
186     @Override
187     public Fragment getItem(int position) {
188         return mFragmentManager.get(position);
189     }
190
191     @Override
192     public int getCount() {
193         return mFragmentManager.size();
194     }
195     public void addFragment(Fragment fragment, String title) {
196         mFragmentManager.add(fragment);
197         mFragmentManagerTitleList.add(title);
198     }
199
200     @Override
201     public CharSequence getPageTitle(int position) {
202         return mFragmentManagerTitleList.get(position);
203     }
204 }
205 }

```

## NotificationService.java

```

1 package com.example.fiqhinaulana.pembacapesan;
2
3 /**
4  * Created by FIQHI MAULANA.
5  */
6
7 import android.app.Notification;
8 import android.content.Context;
9 import android.os.Bundle;
10 import android.service.notification.NotificationListenerService;
11 import android.service.notification.StatusBarNotification;
12 import android.speech.tts.TextToSpeech;
13 import android.support.v4.content.LocalBroadcastManager;
14 import android.util.Log;
15 import android.widget.Toast;
16 import java.text.SimpleDateFormat;
17 import java.util.Calendar;
18 import java.util.Locale;
19
20
21 public class NotificationService extends NotificationListenerService implements TextToSpeech.OnInitListener {
22
23     public static TextToSpeech getTts() {
24         return tts;
25     }
26     public void setTts(TextToSpeech tts) {
27         this.tts = tts;
28     }
29
30     DataBase db;
31     public static TextToSpeech tts;
32     private String mPreviousNotificationKey = "";
33     private Notification mPreviousNotification = null;
34     private PreferenceManager prefManager;
35
36     Context context;
37
38     boolean bicara = false;
39
40     @Override
41     public void onCreate() {
42         super.onCreate();
43         //inisialisasi context, penghubung service dengan context
44         context = getApplicationContext();
45         //inisialisasi text to speech
46         tts = new TextToSpeech(context, this);
47
48         Log.d("NOTIFICASERV", "On Created " + bicara);
49
50         db = new DataBase(context);
51         prefManager = new PreferenceManager(context);
52     }
53
54     public void onNotificationPosted(StatusBarNotification sbn) {
55         if(bicara){
56             bicara = false;
57         } else {
58             Log.i("NotifyService", "got notification");
59             String pack = sbn.getPackageName();
60             Bundle extras = sbn.getNotification().extras;
61             String title = extras.getString("android.title");
62             String text = extras.getString("android.text");
63             mPreviousNotificationKey = sbn.getKey();
64             mPreviousNotification = sbn.getNotification();
65
66             if (pack.equals("org.telegram.messenger")|| pack.equals("com.whatsapp")){
67                 Intent msgrcv = new Intent("Msg");
68                 msgrcv.putExtra("package", pack);
69                 msgrcv.putExtra("title", title);
70                 msgrcv.putExtra("text", text);
71                 Calendar calendar = Calendar.getInstance();
72                 SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
73                 String created_at = dateFormat.format(calendar.getTime());
74                 db.addChat(title, text, pack, created_at);

```

```
75         if(tts != null && preffManager.isTtsOn()){
76             ParseSingkatan parser = new ParseSingkatan(context);
77             text = parser.parseSingkatan(text);
78             text = parser.parseDatabase(text);
79             tts.speak("Pengirim,"+title+" "+text,TextToSpeech.QUEUE_ADD,null);
80         }
81         LocalBroadcastManager.getInstance(context).sendBroadcast(msgrcv);
82         bicara = true;
83     }
84 }
85 }
86
87 @Override
88 public void onNotificationRemoved(StatusBarNotification sbn) {
89     tts.stop();
90     Log.i("Msg","Notification Removed");
91 }
92
93 @Override
94 public void onInit(int status) {
95     if (status == TextToSpeech.SUCCESS){
96         tts.setLanguage(new Locale("id","ID"));
97     } else {
98         Toast.makeText(context,"Inisialisasi TTS tidak berhasil",Toast.LENGTH_LONG).show();
99     }
100 }
101 }
```

## ParseSingkatan.java

```
1 package com.example.fiqhimaulana.pembacapesan;
2
3 import android.content.Context;
4 import android.text.TextUtils;
5 import java.util.List;
6
7 /**
8  * Created by FIQHI MAULANA.
9  */
10
11 public class ParseSingkatan {
12     DataBase db;
13     List<KataSingkatan> kataSingkatan;
14     public ParseSingkatan(Context context){
15         db = new DataBase(context);
16         kataSingkatan = db.getKata();
17     }
18     public String parseSingkatan(String words){
19         String[] s = words.split("\\s+");
20         for ( int i = 0; i < s.length; i++) {
21             s[i].replaceAll("[^a-zA-Z0-9]", "");
22         }
23         String parsed = TextUtils.join(" ",s);
24         return parsed;
25     }
26
27     public String parseDatabase(String words){
28         kataSingkatan = db.getKata();
29         String[] s = words.split("\\s+");
30         for (int i =0; i< s.length;i++){
31             for (KataSingkatan kata: kataSingkatan) {
32                 if (s[i].equals(kata.getSingkatan())){
33                     s[i] = kata.getKepanjangan();
34                 }
35             }
36         }
37
38         String parsed = TextUtils.join(" ",s);
39         return parsed;
40     }
41 }
```

## PreferenceManager.java

```
1 package com.example.fiqhimaulana.pembacapesan;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5
6 /**
7  * Created by FIQHI MAULANA.
8  */
9 public class PreferenceManager {
10     SharedPreferences pref;
11     SharedPreferences.Editor editor;
12     Context _context;
13
14     int PRIVATE_MODE = 0;
15
16     private static final String PREF_NAME = "intro_slider-welcome";
17
18     private static final String IS_FIRST_TIME_LAUNCH = "IsFirstTimeLaunch";
19
20     private static final String IS_TTS_ON = "IsTTSOn";
21
22     public PreferenceManager(Context context) {
23         this._context = context;
24         pref = _context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
25         editor = pref.edit();
26     }
27
28     public void setFirstTimeLaunch(boolean isFirstTime) {
29         editor.putBoolean(IS_FIRST_TIME_LAUNCH, isFirstTime);
30         editor.commit();
31     }
32
33     public boolean isFirstTimeLaunch() {
34         return pref.getBoolean(IS_FIRST_TIME_LAUNCH, true);
35     }
36
37     public void setIsTtsOn(boolean isTtsOn){
38         editor.putBoolean(IS_TTS_ON, isTtsOn);
39         editor.commit();
40     }
41
42     public boolean isTtsOn(){
43         return pref.getBoolean(IS_TTS_ON, true);
44     }
45 }
46
47
```

## Telegram.java

```

1 package com.example.fiqhismaulana.pembacapan;
2
3 /**
4  * Created by FIQHI MAULANA.
5  */
6
7 import android.content.BroadcastReceiver;
8 import android.content.Context;
9 import android.content.Intent;
10 import android.content.IntentFilter;
11 import android.os.Bundle;
12 import android.support.annotation.Nullable;
13 import android.support.v4.app.Fragment;
14 import android.support.v4.content.LocalBroadcastManager;
15 import android.support.v7.widget.DefaultItemAnimator;
16 import android.support.v7.widget.DividerItemDecoration;
17 import android.support.v7.widget.LinearLayoutManager;
18 import android.support.v7.widget.RecyclerView;
19 import android.view.LayoutInflater;
20 import android.view.View;
21 import android.view.ViewGroup;
22 import java.text.SimpleDateFormat;
23 import java.util.ArrayList;
24 import java.util.Calendar;
25 import java.util.List;
26
27 public class Telegram extends Fragment{
28     RecyclerView pesan;
29     List<ChatModel> listChat = new ArrayList<>();
30     ChatAdapter mAdapter;
31     DataBase db;
32
33     public Telegram() {
34     }
35
36     @Override
37     public void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39     }
40
41     @Nullable
42     @Override
43     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
44         View view = inflater.inflate(R.layout.telegram, container, false);
45         pesan = (RecyclerView) view.findViewById(R.id.rcv_telegram);
46         db = new DataBase(getContext());
47         listChat = db.getChatDetails(db.TELEGRAM_PACK);
48         mAdapter = new ChatAdapter(getActivity().getApplicationContext(), listChat);
49         pesan.setLayoutManager(new LinearLayoutManager(getActivity().getApplicationContext()));
50         pesan.setItemAnimator(new DefaultItemAnimator());
51         pesan.setHasFixedSize(true);
52         pesan.addItemDecoration(new DividerItemDecoration(getActivity().getApplicationContext(), LinearLayoutManager.VERTICAL));
53         pesan.setAdapter(mAdapter);
54         LocalBroadcastManager.getInstance(getActivity().getApplicationContext()).registerReceiver(onNotice, new IntentFilter("Msg"));
55         return view;
56     }
57
58     private BroadcastReceiver onNotice = new BroadcastReceiver() {
59         @Override
60         public void onReceive(Context context, Intent intent) {
61             String pack = intent.getStringExtra("package");
62             String title = intent.getStringExtra("title");
63             String text = intent.getStringExtra("text");
64             Calendar calendar = Calendar.getInstance();
65             SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
66             String created_at = dateFormat.format(calendar.getTime());
67
68             if(pack.equals("org.telegram.messenger")){
69                 listChat.add(0, new ChatModel("", title, text, created_at));
70                 mAdapter.notifyDataSetChanged();
71             }
72         }
73     };
74 }
75

```



## WelcomeActivity.java

```

1 package com.example.fiqhimaulana.pembekapesan;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.graphics.Color;
7 import android.os.Build;
8 import android.os.Bundle;
9 import android.support.v4.view.PagerAdapter;
10 import android.support.v4.view.ViewPager;
11 import android.text.Html;
12 import android.view.LayoutInflater;
13 import android.view.View;
14 import android.view.ViewGroup;
15 import android.view.Window;
16 import android.view.WindowManager;
17 import android.widget.Button;
18 import android.widget.LinearLayout;
19 import android.widget.TextView;
20
21 /**
22  * Created by FIQHI MAULANA.
23  */
24
25 public class WelcomeActivity extends Activity {
26
27     private ViewPager viewPager;
28     private MyViewPagerAdapter myViewPagerAdapter;
29     private LinearLayout dotsLayout;
30     private TextView[] dots;
31     private int[] layouts;
32     private Button btnSkip, btnNext, btnAccess;
33     private PreferenceManager prefManager;
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         prefManager = new PreferenceManager(this);
39         if (!prefManager.isFirstTimeLaunch()) {
40             launchHomeScreen();
41             finish();
42         }
43         if (Build.VERSION.SDK_INT >= 21) {
44             getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_STABLE | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
45         }
46         setContentView(R.layout.activity_welcome);
47         viewPager = (ViewPager) findViewById(R.id.view_pager);
48         dotsLayout = (LinearLayout) findViewById(R.id.layoutDots);
49         btnSkip = (Button) findViewById(R.id.btn_skip);
50         btnNext = (Button) findViewById(R.id.btn_next);
51         btnAccess = (Button) findViewById(R.id.accessbtn);
52         layouts = new int[]{
53             R.layout.slide1,
54             R.layout.slide2,
55             R.layout.slide3};
56
57         addBottomDots(0);
58
59         changeStatusBarColor();
60
61         myViewPagerAdapter = new MyViewPagerAdapter();
62         viewPager.setAdapter(myViewPagerAdapter);
63         viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
64
65             btnSkip.setOnClickListener(new View.OnClickListener() {
66                 @Override
67                 public void onClick(View v) {
68                     launchHomeScreen();
69                 }
70             });
71             btnAccess.setOnClickListener(new View.OnClickListener() {
72                 @Override
73                 public void onClick(View v) {
74                     startActivity(new Intent("android.settings.ACTION_NOTIFICATION_LISTENER_SETTINGS"));
75                 }
76             });
77             btnNext.setOnClickListener(new View.OnClickListener() {
78                 @Override
79                 public void onClick(View v) {
80                     int current = getCurrent();
81                     if (current < layouts.length) {
82                         viewPager.setCurrentItem(current);
83                     } else {
84                         launchHomeScreen();
85                     }
86                 }
87             });
88         });
89     }
90
91     private void addBottomDots(int currentPage) {
92         dots = new TextView[layouts.length];

```

```

93
94 int[] colorsActive = getResources().getIntArray(R.array.array_dot_active);
95 int[] colorsInactive = getResources().getIntArray(R.array.array_dot_inactive);
96
97 dotsLayout.removeAllViews();
98 for (int i = 0; i < dots.length; i++) {
99     dots[i] = new TextView(this);
100     dots[i].setText(Html.fromHtml("&#8226;"));
101     dots[i].setTextSize(42);
102     dots[i].setTextColor(colorsInactive[currentPage]);
103     dotsLayout.addView(dots[i]);
104 }
105
106 if (dots.length > 0)
107     dots[currentPage].setTextColor(colorsActive[currentPage]);
108 }
109
110 private int getItem(int i) {
111     return viewPager.getCurrentItem() + i;
112 }
113
114 private void launchHomeScreen() {
115     prefsManager.setFirstTimeLaunch(false);
116     startActivity(new Intent(this, MainActivity.class));
117     finish();
118 }
119
120 ViewPager.OnPageChangeListener viewPagerPageChangeListener = new ViewPager.OnPageChangeListener() {
121
122     @Override
123     public void onPageSelected(int position) {
124         addBottomDots(position);
125
126         if (position == layouts.length - 1) {
127             btnNext.setText(getString(R.string.start));
128             btnSkip.setVisibility(View.GONE);
129             btnAccess.setVisibility(View.VISIBLE);
130         } else {
131             btnNext.setText(getString(R.string.next));
132             btnSkip.setVisibility(View.VISIBLE);
133             btnAccess.setVisibility(View.GONE);
134         }
135     }
136
137     @Override
138     public void onPageScrolled(int arg0, float arg1, int arg2) {
139     }
140
141     @Override
142     public void onPageScrollStateChanged(int arg0) {
143     }
144 };
145
146 private void changeStatusBarColor() {
147     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
148         Window window = getWindow();
149         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
150         window.setStatusBarColor(Color.TRANSPARENT);
151     }
152 }
153
154 public class MyViewPagerAdapter extends PagerAdapter {
155     private LayoutInflater inflater;
156     public MyViewPagerAdapter() {
157     }
158
159     @Override
160     public Object instantiateItem(ViewGroup container, int position) {
161         inflater = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
162         View view = inflater.inflate(layouts[position], container, false);
163         container.addView(view);
164         return view;
165     }
166
167     @Override
168     public int getCount() {
169         return layouts.length;
170     }
171
172     @Override
173     public boolean isViewFromObject(View view, Object obj) {
174         return view == obj;
175     }
176
177     @Override
178     public void destroyItem(ViewGroup container, int position, Object object) {
179         View view = (View) object;
180         container.removeView(view);
181     }
182 }
183 }

```

## Whatsapp.java

```

1 package com.example.fiqhismaulana.pembacapan;
2
3 import android.support.v4.app.Fragment;
4 import android.content.BroadcastReceiver;
5 import android.content.Context;
6 import android.content.Intent;
7 import android.content.IntentFilter;
8 import android.os.Bundle;
9 import android.support.annotation.Nullable;
10 import android.support.v4.content.LocalBroadcastManager;
11 import android.support.v7.widget.DefaultItemAnimator;
12 import android.support.v7.widget.DividerItemDecoration;
13 import android.support.v7.widget.LinearLayoutManager;
14 import android.support.v7.widget.RecyclerView;
15 import android.view.LayoutInflater;
16 import android.view.View;
17 import android.view.ViewGroup;
18 import java.text.SimpleDateFormat;
19 import java.util.ArrayList;
20 import java.util.Calendar;
21 import java.util.List;
22
23 /**
24  * Created by FIQHI MAULANA.
25  */
26
27 public class Whatsapp extends Fragment {
28     RecyclerView pesan;
29     List<ChatModel> listChat = new ArrayList<>();
30     ChatAdapter mAdapter;
31     DataBase db;
32
33     public Whatsapp(){
34     }
35
36     @Override
37     public void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39     }
40
41     @Nullable
42     @Override
43     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
44         View view = inflater.inflate(R.layout.whatsapp, container, false);
45         pesan = (RecyclerView) view.findViewById(R.id.rcv_whatsapp);
46         db = new DataBase(getContext());
47         listChat = db.getChatDetails(db.WHATSAPP_PACK);
48         mAdapter = new ChatAdapter(getContext().getApplicationContext().listChat);
49         pesan.setLayoutManager(new LinearLayoutManager(getContext().getApplicationContext()));
50         pesan.setItemAnimator(new DefaultItemAnimator());
51         pesan.setHasFixedSize(true);
52         pesan.addItemDecoration(new DividerItemDecoration(getContext(),LinearLayoutManager.VERTICAL));
53         pesan.setAdapter(mAdapter);
54         LocalBroadcastManager.getInstance(getContext()).getApplicationContext().registerReceiver(onNotice,new IntentFilter("Msg"));
55         return view;
56     }
57
58     private BroadcastReceiver onNotice = new BroadcastReceiver() {
59         @Override
60         public void onReceive(Context context, Intent intent) {
61             String pack = intent.getStringExtra("package");
62             String title = intent.getStringExtra("title");
63             String text = intent.getStringExtra("text");
64             Calendar calendar = Calendar.getInstance();
65             SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
66             String created_at = dateFormat.format(calendar.getTime());
67
68             if(pack.equals("com.whatsapp")){
69                 listChat.add(0,new ChatModel("",title,text,created_at));
70                 mAdapter.notifyDataSetChanged();
71             }
72         }
73     };
74
75 }

```

## BIODATA PENULIS



**Fiqhi Maulana** meraih gelar Sarjana Komputer di STMIK Handayani pada Program Studi Teknik Informatika; sekarang bekerja pada beberapa proyek terutama di Desain Grafik, Layanan TI, dan Konsultan TI; menguasai Bahasa pemrograman Java, Corel Draw, Adobe Photoshop, Adobe Illustrator, SketchUp, Microsoft Office Advance.



**Najirah Umar** adalah Dosen Kopertis Wilayah IX yang dipekerjakan pada STMIK Handayani; sebagai dosen pengampuh mata kuliah Teknik Riset Operasi, Rekayasa Perangkat Lunak, dan aktif melakukan penelitian antara lain Aplikasi E\_Trash Menggunakan Teknologi Location Based Service (LBS), Personal popular Name Identification Through Twitter Data, Penelitian yang dilaksanakan didanai internal PT, maupun DP2M Kemenristek DIKTI dan menjadi Riviewer Jurnal Nasional.



**M. Adnan Nur** adalah Dosen Kopertis Wilayah IX yang dikerjakan pada STMIK Handayani; sebagai dosen pengampuh mata kuliah Pemrogram Web, Sistem Basis Data dan aktif melakukan penelitian antarlain Mobile Application Bimbingan Tugas Akhir Mahasiswa Pada STMIK Handayani Makassar sebagai Media Pendukung Pembelajaran Daring, Perbandingan Levenshtein Distance dan Jaro-Winkler Distance untuk Koreksi Kata dalam Preprocessing Analisis Sentimen Pengguna Twitter, Penelitian yang dilakukan didanai internal PT, maupun DP2M Kemenristek DIKTI.