

**SISTEM PEMBACA PESAN *TEXT TO SPEECH*
BERBASIS ANDROID MENGGUNAKAN
*FRAMEWORK GOOGLE API***



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1
Departemen Teknik Informatika Fakultas Teknik
Universitas Hasanuddin
Makassar*

Disusun Oleh

**NURPADILA MUSLIMIN
D42113031**

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR**

2017



LEMBAR PENGESAHAN SKRIPSI

SISTEM PEMBACA PESAN *TEXT-TO-SPEECH* BERBASIS ANDROID MENGGUNAKAN FRAMWORK GOOGLE API

Oleh:

NURPADILA MUSLIMIN D42113031

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 22 November 2017.

Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik
pada Program Studi S1 Teknik Informatika Departemen Teknik Informatika Fakultas
Teknik Universitas Hasanuddin

Makassar, 22 November 2017

Disetujui Oleh:

Pembimbing I

Dr. Ir. Ingrid Nurtanio, M.T
Nip. 19610813 198811 2 001

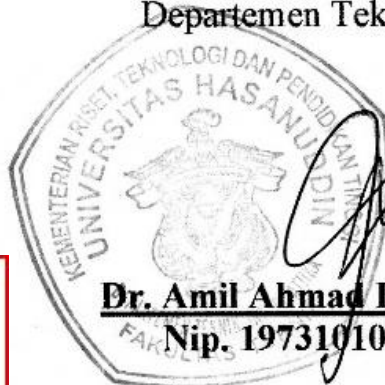
Pembimbing II

Dr. Eng. Intan Sari Areni, ST., MT.
Nip. 19750203 200012 2 002

Diterima dan disahkan oleh:

Ketua Program Studi S1 Teknik Informatika

Departemen Teknik Informatika



Dr. Amil Ahmad Ilham, S.T., M.IT.
Nip. 19731010 199802 1 001



ABSTRAK

Telegram, Line dan WhatsApp merupakan beberapa pesan instan yang telah banyak digunakan sebagai pengiriman pesan singkat yang mudah digunakan dan terpopuler hingga saat ini. Namun dalam berkirim pesan, pengguna biasanya mengalami keterbatasan saat akan membaca pesan yang masuk. Beberapa keterbatasan yang terjadi diantaranya sedang berkendara, penyandang tunanetra, dan jarak ponsel dengan pengguna. Aplikasi pembaca pesan yang berjalan pada sistem operasi *Android* dan terintegrasi dengan sistem *text to speech* dapat menjadi solusi alternatif bagi pengguna yang ingin melihat atau mendengarkan pesan yang disampaikan kepadanya tanpa harus mengganggu aktifitas. Sistem pembaca pesan *text to speech* akan menerima pesan instan dari *Telegram, Line dan WhatsApp* menjadi ucapan dan bekerja di belakang layar serta mengkonversi teks pesan yang diterima jika pesan mengandung singkatan. Dengan menggunakan *Android Studio*, bahasa pemrograman *Java* dan sistem manajemen *database SQLite* serta *NotificationListenerService* membuat sistem mampu menerima *pack, tittle dan text* dari pemberitahuan pesan *Telegram, Line dan WhatsApp* yang ditampilkan pada layar *smartphone* sehingga sistem berjalan dibelakang layar tanpa harus membuka aplikasi utama. *API Google Text to Speech Voice Indonesia* membuat sistem menghasilkan keluaran suara sesuai fonem dengan lafal bahasa Indonesia. Tingkat akurasi membaca pesan sesuai keluaran yang diharapkan mencapai 78,43%. Dengan memasukkan kata singkatan dan kata yang dimaksud (kepanjangan) ke dalam *database* sistem mampu mengkonversi kata dengan benar sehingga di peroleh peningkatan akurasi hingga 92,85%.

Kata Kunci: *android, text to speech, NotificationListenerService, Line, Telegram, WhatsApp, SQLite*



KATA PENGANTAR

Assalamu Alaikum Warahmatullahi Wabarakatuh

Puji dan rasa syukur mendalam penulis panjatkan kehadiran Allah SWT, karena berkat limpahan rahmat, hidayah dan inayah-Nya maka skripsi ini dapat diselesaikan dengan baik. Salam dan salawat semoga selalu tercurah pada baginda Rasulullah Muhammad SAW.

Skripsi yang berjudul "**Sistem Pembaca Pesan *Text to Speech* Berbasis Android Menggunakan Framework Google Api**" di susun untuk memenuhi persyaratan kurikulum sarjana strata-1 (S-1) Teknik Informatika Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis mengucapkan rasa terima kasih yang sebesar-besarnya atas semua bantuan yang telah diberikan, baik secara langsung maupun tidak langsung selama penyusunan tugas akhir ini hingga selesai. Secara khusus rasa terimakasih tersebut kami sampaikan kepada:

1. Allah SWT, Tuhan Yang Maha Esa atas segala berkah rahmat dan hidayah-Nya penulis akhirnya dapat menyelesaikan sistem dan penulisan laporan tugas akhir ini
2. Kedua Orang Tua dan Keluarga, Muslimin Sirumpa dan Nurhayati yang tidak pernah henti-hentinya memberikan doa, kasih sayang, dan motivasi selama masa perkuliahan hingga penyelesaian tugas akhir ini.
3. Ibu Dr.Ir.Ingrid Nurtanio, M.T selaku pembimbing I dan Ibu Dr.Eng.Intan

ari Areni, ST., MT selaku pembimbing II atas kesabaran dalam membimbing serta arahan dalam penyusunan tugas akhir ini.



4. Bapak Dr. Amil Ahmad Ilham ST.,M.IT., Ph.D selaku Ketua Prodi Teknik Informatika fakultas Teknik Universitas Hasanuddin.
5. Bapak Robert dan Bapak Zainuddin selaku staff Kantor Prodi Informatika Universitas Hasanuddin atas segala bantuannya.
6. Arisal Ahmad Saila, M.Takdir, Haryati Amaliyah, Andi Januar Khaerul, Muhammad Idmaril Amri, Andi Mabror Abulkhaer, dan Siti Khumaerah Mufti selaku teman yang telah banyak memberikan kenyamanan, dan segala bantuannya selama masa perkuliahan hingga penyusunan tugas akhir ini.
7. Wardiman Hasbi, Zandy Rizaldi, Surya Adi Tama, Ahmad Rifaldi, Zaldy Eka Putra, Rahmat Hardian dan Arya Surya Prayoga yang telah memberikan waktu berharga hingga penyusunan tugas akhir ini.
8. Suharyuni, Andi Febriana, Sri Ayu Putri Ningsi, Hasriana, Sisca Rosari Putriani, Nurul Fajrin Alifah, Angreyni Devianti, Isnaeni,Ayu Dhiya Mufidah, Himawan Sandhi, Arfan Almukaddas dan kak Sri Wahyuni selaku teman seperjuangan menyelesaikan tugas akhir.
9. Teman-teman Prodi Teknik Informatika terhusus angkatan 2013 atas segala dukungan dan bantuannya.
10. Serta seluruh pihak yang tidak sempat saya sebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran selama penyusunan Laporan Tugas Akhir ini.

Akhirnya dengan segala kerendahan hati, penyusun menyadari masih terdapat an dalam penyusunan Laporan Skripsi ini baik isi maupun cara penyajian. ena itu penyusun mengharapkan adanya saran dan kritik yang bersifat



membangun demi kesempurnaan laporan ini. Penyusun berharap semoga Laporan Skripsi ini dapat memberikan manfaat bagi pembaca pada umumnya dan manfaat bagi penyusun pada khususnya.

Wassalamu Alaikum Warahmatullahi Wabarakatuh

Makassar, November 2017

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN SKRIPSI.....	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan.....	5
TINJAUAN PUSTAKA.....	7
2.1 Landasan Teori Dasar	7
2.1.1 Teknologi Pemrosesan Bahasa.....	7
2.1.2 Text To Speech (TTS).....	8
2.1.3 Google API	10
2.1.5 Bahasa Pemograman	16
2.1.5.1 Bahasa Java	16
2.1.6 Sistem Operasi Android	17
2.1.6.1 Versi Android.....	17
2.1.6.2 Komponen Aplikasi Android	22
2.1.6.3 Database SQLite.....	24
Software Yang Digunakan	25
2.1 Android Studio.....	25
METODOLOGI PENELITIAN.....	28



3.1	Tahapan Penelitian	28
3.2	Waktu dan Lokasi Penelitian.....	29
3.3	Studi Literatur	29
3.4	Analisis Kebutuhan Perangkat.....	29
3.4.1	<i>Software</i>	29
3.4.2	<i>Hardware</i>	30
3.5	Rancangan Sistem.....	31
3.5.1	Rancangan Proses	32
3.5.2	Rancangan Fitur	39
3.5.3	Rancangan Basis Data	42
3.6	Validasi Penelitian	43
BAB IV		45
IMPLEMENTASI DAN PEMBAHASAN.....		45
4.1	Gambaran Umum Aplikasi.....	45
4.2	Implementasi Sistem.....	45
4.2.1	Implementasi Antarmuka	46
4.3	Pengujian Sistem.....	51
4.3.1	Rencana Pengujian	51
4.1.1	Kasus dan Hasil Pengujian	52
BAB V		60
PENUTUP		60
5.1	Kesimpulan.....	60
5.2	Saran	61
DAFTAR PUSTAKA.....		62
LAMPIRAN		63



DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian	28
Gambar 3.2 Flowchart Sistem Secara Umum	32
Gambar 3.3 Inisialisasi Notification Listener Service.....	33
Gambar 3.4 Inisialisasi TTS	34
Gambar 3.5 Objek ke string	35
Gambar 3.6 Output suara setelah normalisasi	39
Gambar 3.7 Fragment dalam activity	40
Gambar 3.8 Flowchart Input Singkatan.....	41
Gambar 3.9 Option Menu.....	42
Gambar 4.1 layout fragment Telegram	46
Gambar 4.2 layout fragment Line dan WhatsApp.....	47
Gambar 4.3 Options menu	48
Gambar 4 4 Alert Dialog tambah singkatan	49
Gambar 4.5 Menghapus berdasarkan package	50
Gambar 4 6 Penerimaan Pesan.....	50



DAFTAR TABEL

Tabel 2.1 Daftar Android	17
Tabel 3.1 Tabel Chat	43
Tabel 3.2 Tabel Singkatan.....	43
Tabel 4.1 Rencana Pengujian fitur Aplikasi.....	52
Tabel 4.2 Hasil Pengujian Fitur.....	53
Lanjutan Tabel 4.2 Hasil Pengujian Fitur.....	54
Tabel 4.3 Hasil Membaca Pesan	55
Lanjutan Tabel 4.3 Hasil Membaca Pesan	56
Lanjutan Tabel 4.3 Hasil Membaca Pesan.....	57



BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut lembaga riset pasar *e-Marketer*, populasi *netter* Tanah Air mencapai 83,7 juta orang pada 2014. Angka yang berlaku untuk setiap orang yang mengakses internet setidaknya satu kali setiap bulan itu mendudukkan Indonesia di peringkat ke-6 terbesar di dunia dalam hal jumlah penggunaan internet. Pada 2018, *e-marketer* memperkirakan *netter* Indonesia bakal mencapai 123 juta orang, mengalahkan Jepang di peringkat ke-5 yang pertumbuhan jumlah pengguna internetnya lebih lambat (KOMINFO, 2014).

Marius dan Sapto menyatakan bahwa ada tiga alasan orang Indonesia menggunakan internet. Tiga alasan itu adalah untuk mengakses sarana sosial/komunikasi (72%), sumber informasi harian (65%) dan mengikuti perkembangan jaman (51%). Tiga alasan utama mengakses internet itu dipraktikan melalui empat kegiatan utama, yaitu menggunakan jejaring sosial (87%), mencari informasi (69%), *instant messaging* (60%) dan mencari berita terbaru (60%) (Marius and Sapto, 2014).

Dalam artikel yang dibahas oleh Scott Jeffrey ada tiga *instant messaging* yang memiliki kelebihan yang berbeda menjadi daya tarik bagi pengguna ponsel diantaranya *Telegram* dengan kelebihan lebih ringan dan berbasis *cloud* yang

yai proses sinkronisasi yang mulus, *Line* dengan kelebihan memiliki fitur



yang menarik serta menyajikan berita-berita terbaru dan *WhatsApp* dengan kelebihan tampilan yang mudah dengan sistem 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 yang diterima 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 merupakan suatu sistem yang melakukan dua konversi yaitu konversi teks ke fonem dan konversi fonem ke ucapan yang dilakukan secara berurutan dengan *input* teks dan menghasilkan *output* 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.

Penelitian yang telah dilakukan Dodi Sireger, terkait dengan sistem *text to speech* bahasa Indonesia pada pesan SMS dengan konverter PDU. Dari hasil penelitian yang telah dilakukan memiliki beberapa batasan masalah dimana menggunakan *Personal Computer* (PC) sebagai pengambil pesan SMS yang ada



pada ponsel serta menggunakan MBROLA *Speech Engine* sebagai pembangkit ucapan dengan *database diphone* bahasa Indonesia (Siregar, 2015).

Maka dari pembahasan yang telah dilakukan, diusulkan sistem pembaca pesan yang akan menerima teks pesan dari *Telegram, Line* dan *WhatsApp* kemudian mengubahnya menjadi ucapan dengan menggunakan *google API text-to-speech voice* bahasa Indonesia. Sehingga dalam penelitian ini penulis menyusun tugas akhir yang berjudul “Sistem Pembaca Pesan *Text to Speech* Berbasis *Android* Menggunakan *Framework Google Api*”

1.2 Rumusan Masalah

Berdasarkan pada latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana membuat sistem *text to speech* dari pesan instan *Telegram, Line* dan *WhatsApp* dan bekerja di belakang layar pada *android*?
2. Bagaimana mengkonversi teks pesan yang diterima jika *output* ucapan yang diterima tidak sesuai (mengandung singkatan)?

1.3 Tujuan Penelitian

Tujuan akhir dari penelitian ini adalah :

1. Membuat sistem *text to speech* dari pesan instan *Telegram, Line* dan *WhatsApp* dan bekerja di belakang layar pada *android*.
2. Membuat konversi teks pesan yang diterima jika *output* ucapan yang diterima tidak sesuai (mengandung singkatan)



1.4 Manfaat Penelitian

Manfaat akhir dari penelitian ini adalah :

1. Mempermudah dan membantu user ketika mendapat hambatan dalam menerima pesan dari *Telegram*, *Line* dan *WhatsApp* tanpa harus melihat dan membaca pesan yang diterima.
2. Memberikan esensi baru dalam hal penggunaan sistem yang berbasis *android* serta, membantu penelitian lebih lanjut dalam pengembangan sistem *text to speech*.

1.5 Batasan Masalah

Yang menjadi batasan masalah dalam tugas akhir ini adalah :

1. Sistem ini akan berjalan di *Operating System Android*.
2. Mengaktifkan aplikasi pembaca pesan pada akses notifikasi yang berada pada pengaturan di *smartphone*.
3. Mengaktifkan dan mengatur Bahasa keluaran *google TTS* pada pengaturan di *smartphone* untuk *voice* Indonesia.
4. Aplikasi ini akan membaca pesan *Line*, *Telegram* dan *WhatsApp*.
5. Pesan yang dibaca adalah pesan baru yang diterima dan muncul sebagai pemberitahuan pada layar *smartphone*.
6. Pesan yang diterima sebagai pemberitahuan akan di simpan pada aplikasi pembaca pesan sebagai tanda pesan diterima dan dapat mendengarkan kembali pesan yang ada.



7. Pesan singkatan akan dibaca ketika singkatan telah di *input* dan sudah terdapat dalam *database* singkatan.
8. Menyelesaikan pesan yang dibaca pertama kemudian mengkonversi pesan selanjutnya sampai pesan dalam antrian selesai.
9. Database yang digunakan adalah *SQLite*

1.6 Metodologi Penelitian

Adapun metodologi penelitian pada pengerjaan proyek tugas akhir ini adalah :

1. Studi literatur

Pada tahap ini dilakukan kajian terhadap berbagai literatur yang dapat menunjang perancangan proyek tugas akhir ini.

2. Diskusi dan konsultasi

Pada tahap ini dilakukan proses diskusi dan konsultasi dengan dosen pembimbing serta berbagai pihak yang kompeten dalam bidang ini.

3. Perancangan system

Pada tahap ini dilakukan analisis kerja dan perancangan aplikasi.

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terbagi menjadi 5 (lima) bab dengan beberapa sub bab. Berikut ini adalah sistematika penulisannya:



BAB I. PENDAHULUAN

Bab ini terdiri dari tujuh sub bab, yaitu Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang digunakan dalam Aplikasi Pembaca Pesan *Text to Speech* yang berjalan pada android.

BAB III. METODOLOGI PENELITIAN

Bab ini menguraikan secara rinci tentang Tahapan Penelitian, Waktu Dan Lokasi Penelitian, Studi Literatur, Analisis Kebutuhan Perangkat, Rancangan Sistem, Rancangan Basis Data dan Validasi Penelitian.

BAB IV. IMPLEMENTASI DAN PEMBAHASAN

Bab ini terdiri dari empat sub bab yang membahas mengenai Gambaran Umum Aplikasi, Implementasi Sistem dan Pengujian Sistem.

BAB V. PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan tentang hasil pemecahan masalah yang diperoleh selama penyusunan tugas akhir ini dan beberapa saran untuk pengembangan lebih lanjut.



BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori Dasar

2.1.1 Teknologi Pemrosesan Bahasa

Bahasa dapat dibedakan menjadi dua, yaitu bahasa alami dan bahasa buatan. Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Inggris, Jepang dan sebagainya. Bahasa Buatan adalah bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu, misalnya bahasa pemrograman komputer.

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

- a. Sub-Sistem *Natural Language Processing* (NLP), berfungsi untuk melakukan pemrosesan secara simbolik terhadap bahasa tulisan. Beberapa bentuk aplikasi sub-sistem ini adalah translator bahasa alami, sistem pemeriksaan sintak bahasa, sistem yang dapat menyimpulkan suatu narasi dan sebagainya.
- b. Sub-Sistem *Text-to-Speech* (TTS), berfungsi untuk mengubah teks (tulisan) menjadi ucapan (bahasa lisan).
- c. Sub-Sistem *Speech Recognition* (SR), yaitu sistem yang berfungsi untuk mengubah atau mengenali suatu ucapan (bahasa lisan) menjadi teks (bahasa

san).



2.1.2 Text To Speech (TTS)

TTS dapat didefinisikan sebagai sistem yang dapat mengubah suatu teks menjadi ucapan secara otomatis dengan cara *fonetisasi* (penyusunan fonem-fonem untuk membentuk ucapan). Tujuan dari *text-to-speech* (TTS) adalah untuk mengkonversi *input* berupa teks menjadi ucapan yang dimengerti oleh pendengar. Jadi, *text-to-speech* mengirimkan informasi dari mesin kepada manusia.

Membaca adalah proses men-*decode*-kan *signal* tulisan menjadi *message*, lalu berusaha memahami *message* tersebut untuk mendapatkan suatu pengertian (pemahaman). Membaca dengan bersuara (*reading aloud*) berarti proses mengkonversi *signal* tulisan menjadi *signal* dalam bentuk yang lain, yaitu ucapan (*speech*).

Dalam tulisan ada satu hal yang paling penting yang diabaikan, yaitu prosodi. Prosodi berfungsi untuk membentuk emosi dari pembicara. Dalam bahasa tulisan, prosodi ini memiliki komponen yang terbatas karena tidak adanya cara untuk mengekspresikan beberapa fitur di dalam prosodi tersebut. Kita memang dapat memberikan tanda baca seperti tanda seru, tanda tanya dan sebagainya untuk memberi penekanan, namun hal-hal seperti kemarahan, kesakitan, atau sarkasme, tidak ada acara yang khusus untuk mengutarakannya.

Dalam tulisan terkandung bahasa yang bukan merupakan bagian dari linguistik, misalnya bahasa matematika. Ketika menemui suatu bahasa matematika, mesin harus dapat membedakan dengan bahasa natural (*natural*

e). Kita ambil sebagai contoh, $x^2 + y^2$. Kita tahu bahwa angka 2 diatas



huruf x dan y menyatakan pemangkatan, sehingga kalimat ini dibaca sebagai, x kuadrat ditambah y kuadrat.

Permasalahan tersebut termasuk di dalam masalah semiotik. Bahasa natural dikenal sebagai sistem semiotik. Mesin harus menginterpretasikan kalit yang mengandung lambang semiotik ketika menemui kalimat yang mengandung lambang tersebut. Agar mesin dapat memproses tulisan seperti itu, mesin harus dapat mengidentifikasi bagian mana dari teks yang berhubungan dengan masalah semiotik.

Masalah-masalah yang dibahas di atas merupakan beberapa masalah yang sering ditemui di dalam proses membaca. Sehingga dapat diidentifikasi model dari TTS yang mampu melakukan proses yang diinginkan (Wundari, 2009).

Pada dasarnya *Text-to-Speech* adalah suatu sistem yang dapat mengubah teks menjadi ucapan. Suatu pensintesa ucapan atau *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu:

- a. Bagian konverter teks ke fonem (*Text to Phoneme*)
- b. Bagian konverter fonem ke ucapan (*Phoneme to Speech*)

Phonem adalah istilah linguistik dan merupakan satuan terkecil dalam sebuah bahasa yang masih bisa menunjukkan perbedaan makna. Bagian konverter teks ke fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode *phonem*, durasi serta *pitch*-nya.



Bagian konverter *phonem* ke ucapan akan menerima masukan berupa kode-kode *phonem* serta *pitch* dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian konverter *phonem* ke ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat-kalimat yang ingin diucapkan.

Konversi dari teks ke *phonem* sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tersebut menjadi simbol-simbol *phonetik* yang mempresentasikan unit bunyi terkecil dalam suatu bahasa (Pramono, 2013).

2.1.3 Google API

Google API merupakan bagian dari *Framework Google*. *Google* menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang aplikasi baik *web*, *mobile*, maupun *desktop* untuk memanfaatkan berbagai fitur yang disediakan oleh *Google* seperti: *AdSense*, *Search Engine*, *Translation*, *YouTube* dan lain sebagainya. API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang dibuat dengan fungsi-fungsi yang dikerjakan. Misalnya dalam hal ini *Google API* berarti kode program yang dapat ditambahkan pada aplikasi untuk memanfaatkan fungsi atau fitur yang disediakan *Google*. Contohnya developer dapat menambahkan fitur *Google Map* pada aplikasi yang

dibangun. *Google API* dapat dipelajari langsung melalui *Google Code*.



Melalui *Google Code*, developer dapat belajar tentang *Google API* dan dapat mengimplementasikan pada aplikasi yang sedang dikembangkan. Ada banyak API yang disediakan oleh *Google*, beberapa diantaranya adalah (Irawan, 2014).

1. *Language API*: memanfaatkan fitur translation yang dimiliki *Google*
2. *Earth API*: memanfaatkan fitur yang ada pada *Google Earth*
3. *Maps API*: memanfaatkan fitur yang ada pada *Google Maps*
4. *Search API*: memanfaatkan fitur pencarian pada *Google Search*
5. *Visualization API*: membuat grafik maupun chart dengan *Google API*
6. *YouTube API*: memanfaatkan fitur yang ada pada *YouTube* misalnya untuk pencarian video.

2.1.4 *Google Text To Speech API*

Penggunaan *Google Text To Speech API* memungkinkan developer untuk mengkonversi teks ke dalam ucapan (suara). Untuk dapat menggunakan fitur tersebut, developer *android* dapat menggunakan *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*, diantaranya:

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



2. *Classes*

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

- 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 terpasangnya mesin *text-to-speech*,
- e. *TextToSpeechService*, kelas abstrak dasar untuk implementasi mesin TTS.

Untuk dapat memanfaatkan fitur *Google Text To Speech* API pada program diperlukan beberapa langkah, diantaranya mengimpor *class* yang berhubungan dengan *text to speech*, yaitu `android.speech.tts.TextToSpeech`;

Untuk menggunakan kelas *TextToSpeech*, perlu menginisialisasi sebuah objek dari *class TextToSpeech* dan menentukan *initListenere*.

```
Private EditText write;
```

```
tts = new TextToSpeech(getApplicationContext(),
```

```
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.

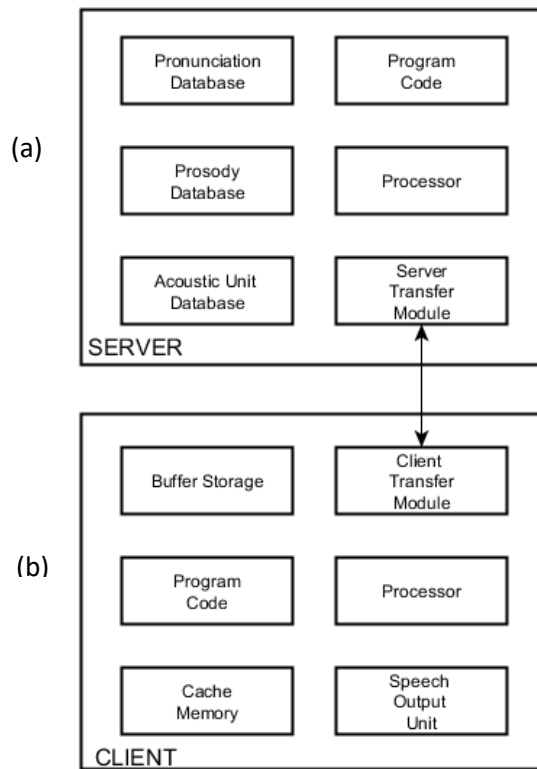
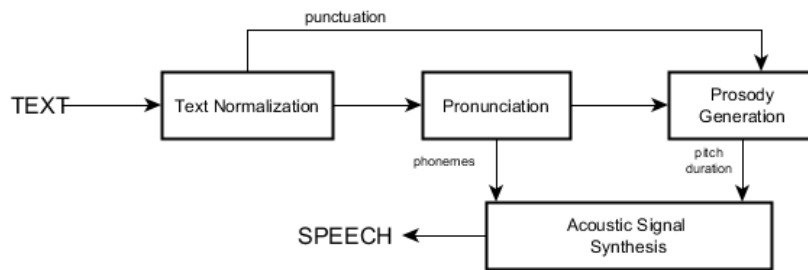
Setelah mengatur bahasa, dapat memanggil *method speak*

```
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()*.



Berikut Gambar 2.1 merupakan alur diagram proses *Google Text to Speech*



Gambar 2.1 Diagram blok sistem sintesis *text-to-speech* dan diagram blok sistem *text-to-speech* klien/server



Pada Gambar 2.1 dapat dilihat teks di konversi 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 ke **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 analisa setiap kata untuk menentukan representasi morfologisnya. Kata-kata yang dihasilkan kemudian diubah menjadi urutan fonem. Terdapat 3 aturan yang digunakan oleh **unit pronunciation** untuk melakukan konversi diantaranya *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

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 telpon 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, dimana 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 and Sunnyvale, 2004).

2.1.5 Bahasa Pemrograman

2.1.5.1 Bahasa Java

Program Android umumnya dibuat menggunakan bahasa Java. Setiap bahasa pemrograman memiliki aturan-aturan tersebut saat menulis suatu program, maka komputer tidak akan mampu memahaminya. Spesifikasi bahasa Java dan Java API mendefinisikan standar baku pemrograman Java.

API (*application programming interface*) memuat kelas dan antarmuka yang telah didefinisikan untuk mengembangkan program Java. Spesifikasi bahasa Java

etapi API masih terus berkembang.



Java merupakan bahasa pemrograman yang tangguh dan terbukti handal pada banyak aplikasi (Sianipar, 2013).

2.1.6 Sistem Operasi Android

2.1.6.1 Versi Android

Android telah mengalami sejumlah pembaruan sejak pertama kali dirilis. Rata-rata, versi terbaru dari Android dirilis setiap enam bulan. Table 2.1 menunjukkan beberapa jenis Android dan nama kodenya. Penamaan kode menggunakan nama makanan dan huruf depannyaurut sesuai abjad.

Tabel 2.1 Daftar Android

Versi	Tanggal Rilis	Kode
1.1	9 Februari 2009	
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Éclair
2.2	20 Mei 2010	Frozen Yoghurt(Froyo)
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb
4.0	19 Oktober 2011	Ice Cream Sandwich
4.1	27 Juni 2012	Jelly Bean
4.2	29 Oktober 2012	Jelly Bean
4.3	24 Juli 2013	Jelly Bean
4.4	3 September 2013	Kitkat
5.0	25 Juli 2014	Lolipop
6.0	17 Agustus 2015	Marshmallow
7.0	1 Juli 2016	Nougat



Berikut beberapa jenis Android dan penjelasannya:

a. *Cupcake*

Android 1.5 (API Level 3) yang dikenal dengan nama kode *Cupcake* ini dirilis pada April 2009. Api level 3 membawa perbaikan dari versi sebelumnya dan tentu saja fitur baru. Misalnya, dukungan *On-screen keyboard*, *video recording*, *home screen widget*, *fitur auto pairing* pada *Bluetooth*, *Kernel Linux* versi 2.6.7 dan kemampuan memperbaiki *filesystem SD card* yang rusak.

b. *Donut*

Android 1.6 (Api level 4) yang dirilis pada Oktober 2009 ini merupakan rilis minor dengan fitur utama dukungan teknologi CDMA, *gesture*, *text-to-speech engine* dan fitur pencarian cepat yang memungkinkan anda mencari semua hal seperti kontak, riwayat jelajah internet, *bookmark* dan lainnya.

c. *Éclair*

Android 2.1 (Api level 7) yang dirilis pada Januari 2010 ini mengungkap fitur baru seperti *live wallpaper* dan beberapa fitur baru seperti Android 2.0, misalnya menggunakan banyak akun kontak cepat, *Bluetooth 2.1* dan *profil Bluetooth* baru yaitu *Object Push Profile (OPP)* dan *Phone Book Access Profile (OBAP)*.

d. *Froyo*

Android 2.2 (API level 8) dirilis pada Mei 2010. Fitur baru yang dibawa adalah

ngan *OpenGL ES 2.0*, instalasi aplikasi penyimpanan eksternal (*SD card*)



dan *Android Cloud to Device Messanging* yang memungkinkan aplikasi melakukan *push messanging* dan *portable hotspot* yang memungkinkan piranti Android menjadi *hotspot* Wi-Fi untuk berbagai koneksi internet.

e. *Gingerbread*

Android 2.3 (API Level 9 dan 10) mrngudung beberapa fitur baru, antara lain dukungan banyak kamera, *Near Field Communicarion* (NFC), *download manager service*, dukungan terhadap sensor lain seperti *giroskop* dan *barometer*.

f. *Honeycomb*

Andoid 2.3 (API Level 11) membawa perubahan besar terutama pada tampilan UI yang berubah drastis supaya optimal untuk piranti layar besar seperti tablet. Pada versi ini diperkenalkan *fragment*, *action bar*, *sistem clipboard* dan *cursor loader*.

g. *Ice Crem Sandwich*

Andoid 4.0 (API Level 14) membawa perubahan besar terutama pada tampilan UI Android 3.0 supaya cocok pada layar kecil sehingga memungkinkan aplikasi anda tampak konsisten di tablet maupun ponsel. Fitur baru seperti *Android Beam* dan *Wi-Fi Direct* juga ditambahkan.

h. *JellyBean*

Android 4.1 (API Level 16) adalah rilis minor yang membawa perbaikan fusionalitas dan performa rendering *User-Interface* (UI). Versi 4.2(API Level

mengusung fitur baru seperti *gesture typing* dan dukungan *multiuser* di



piranti tablet. Versi 4.3 (API Level 18) membawa perbaikan berupa dukungan *Bluetooth low energy* dan *OpenGL ES 3.0*. Beberapa fitur baru antara lain dukungan terhadap Bahasa internasional dan penulisan teks dua arah (kiri kekanan atau kanan ke kiri untuk bahasa-bahasa tertentu seperti bahasa Arab).

i. *Kitkat*

Android 4.4 (API Level 19) ini mengusung sejumlah perbaikan dan fitur baru, terutama dukungan teknologi *Near Field Communication* (NFC) melalui *host card emulation*, pencetakan ke *printer nirkabel*, *WebView* dengan *rendering engine Chromium*, dan dukungan yang lebih baik bagi piranti yang menggunakan RAM rendah.

j. *Lollipop*

Di Android 5.0 (API Level 21) versi L yang merupakan '*Lollipop*', Google lebih memfokuskan pada sisi desain dan performa. Dari segi performa, Android 5.0 sudah mensupport *prosesor* 64 bit dan juga Android 5.0 memiliki *runtime*, *grafis* dan daya tahan baterai yang lebih baik dibanding Android versi sebelumnya. Sebagai catatan bahwa dalam jangka waktu 3 tahun ini *Google* hanya menerbitkan versi Android baru dengan perubahan kecil saja, yaitu pada Android Ice Cream Sandwich hingga Android Kitkat.

k. *Marshmallow*

Android 6.0 (API Level 23) *Marshmallow* memiliki penguatan sistem keamanan lebih baik dari versi sebelumnya, sehingga pengguna bisa memilih

untuk menolak atau menerima permintaan akses keamanan dari aplikasi yang



dipasang ke perangkatnya. Versi ini membuat settingan aplikasi yang sudah dipasang di ponsel tidak akan bisa sembarangan mengambil informasi lokasi atau kamera, tanpa mendapatkan izin terlebih dahulu dari pengguna. Kelebihan Android 6.0 *Marshmallow* juga memiliki fitur bernama *Doze*, yaitu sebuah pengelola daya yang dipakai ponsel. Misalnya, jika ponsel sedang tidak digunakan sama sekali, *Doze* akan otomatis mematikan aplikasi yang aktif. Selain itu, memiliki fitur Android Pay. Dengan adanya fitur ini, para pengguna akan dengan mudah melakukan pembayaran baik di aplikasi dengan menggunakan ponsel saja.

l. *Nougat*

Versi Android 7.0 (API Level 24) *Nougat* umumkan pada tanggal 1 Juli 2016, dan diluncurkan pada 23 Agustus 2016. *Android Nougat* menyediakan banyak fitur baru yang menarik bagi pengembang Android. Fitur-fitur tersebut meliputi:

- Dukungan *multi-window* yang memungkinkan pengguna membuka dua aplikasi di layar sekaligus,
- *Notification* yang telah diperbaharui. Sistem notifikasi yang didesain ulang di *Android Nougat* mencakup fitur *Direct Reply* (Balasan langsung) dari UI notifikasi. Selain itu juga terdapat fitur *bundle* notifikasi yang dapat mengelompokkan notifikasi-notifikasi tertentu,



- Data saver. Fitur ini adalah layanan sistem baru yang membantu mengurangi pengguna data seluler oleh aplikasi. Aplikasi ini memungkinkan pengguna memberi kontrol bagaimana aplikasi menggunakan data seluler.

Selain itu, *Android Nougat* juga membawa banyak perangkat tambahan yang menarik bagi pengembang aplikasi seperti fitur konfigurasi keamanan jaringan baru, *Doze on the Go*, pengesahan kunci, pengaturan API cepat yang baru, dukungan multi-lokal, API ICU4J, penambahan *WebView*, akses ke Java 8 fitur bahasa, akses direktori *scoped*, API penunjuk kustom, dukungan platform VR, file *virtual* dan pengoptimalan prosesan latar belakang.

2.1.6.2 Komponen Aplikasi Android

Aplikasi Android ditulis dengan bahasa pemrograman java. Semua file kode *intermediate* dan *asset* disatukan dalam satu paket berupa file berekstensi. APK, sebuah file yang dapat didistribusi. Tiap file APK adalah sebuah aplikasi tunggal. Komponen aplikasi Android terdiri dari beberapa jenis, antara lain (Juhara, 2016):

a) *Activity*

Activity adalah istilah yang digunakan dalam pemrograman Android untuk mengacu pada satuan interaksi dengan pengguna melalui antarmuka grafis (*graphiacal user- interface*, GUI). Tiap aplikasi dapat terdiri dari nol atau lebih *activity*.



b) *Intent*

Intent adalah istilah yang digunakan dalam pemrograman Android untuk mengacu pada mekanisme berbagai pesan pemberitahuan atau bertukar data *Activity* atau untuk menjalankan aplikasi lain.

c) *Service*

Service adalah komponen aplikasi yang berjalan di belakang layar tanpa *user-interface* untuk menyediakan layanan seperti mengecek RSS *feed* secara kontinu atau memainkan musik. *Service* tetap berjalan meski *Activity* yang mengendalikannya telah berhenti. *Media player* adalah sebuah contoh aplikasi yang menggunakan *service*.

d) *Content Provider*

Content provider membuat suatu aplikasi dapat berbagi sejumlah data tertentu kepada aplikasi lain. Jika membutuhkan nama-nama kontak, aplikasi anda tinggal meminta data tersebut.

e) *Broadcast Receiver*

Broadcast Receiver adalah komponen yang memantau, menerima, dan beraksi terhadap pesan yang disebarkan, baik sistem maupun aplikasi lain. Misalnya, ketika baterai lemah, Android akan mengirim pesan “baterai lemah” kepada semua *broadcast receiver* yang ingin diberitahu pesan ini. Untuk menggunakan *broadcast receiver*, pada dasarnya perlu membuat turunan tipe *broadcast receiver*, melengkapi *method onReceive()*, dan mendaftarkan di

AndroidManifest.xml atau dengan *method Content.registerReceiver()*. *Intance*



broadcast receiver hanya valid selama pemanggilan *method onRecive()* sehingga tidak boleh menyimpan referensi ke *instance* ini.

2.1.6.3 Database SQLite

Android menyediakan sistem manajemen database *SQLite*. Berikut beberapa alasan menggunakan *SQLite* (Juhara, 2016):

a) Sistem *database* yang ringan

SQLite adalah sistem manajemen *database* yang ringan yang cocok diterapkan pada perangkat dengan sumber daya terbatas seperti ponsel. *SQLite* dirancang sebagai pustaka yang berdiri sendiri dengan ketergantungan pada pustaka lain seminimal mungkin.

b) Tidak menggunakan server

SQLite tidak mengadopsi model *client-server*. *SQLite* disediakan sebagai pustaka yang menyediakan fungsionalitas pengelolaan *database*. Melalui pustaka *SQLite*, aplikasi dapat langsung mengelola database di file tanpa melalui *server*. Keuntungannya, langkah *setup* akan lebih mudah. Tidak akan menemukan kerumitan konfigurasi *server* seperti pada sistem manajemen *database* lain. Kekurangannya, karena dikelola sendiri oleh aplikasi, integritas *database* sangat terpengaruh pada praktik pemograman yang baik.

c) Tanpa aturan tipe data ketat

SQLite adalah sistem manajemen *database* yang mengadopsi aturan tipe data yang longgar (*typeless*). Meskipun mendeklarasikan kolom sebuah table

tipe *integer*, dapat diizinkan memasukkan data bertipe lain.



d) Integritas referensi tidak dikelola

SQLite tidak mengelola integritas referensi. Tidak ada batasan *FOREIGN KEY*. Anda dapat mengacu sebuah record pada kolom tabel lain meskipun tabel tersebut sudah tidak lagi valid. Lalu bagaimana mengelola integritas referensi? Anda dapat mengelolanya sendiri menggunakan *TRIGGER*.

2.1.7 Software Yang Digunakan

2.1.7.1 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*). Android studio memiliki fitur (Juansyah, 2015):

- a. Projek berbasis pada *Gradle Build*
- b. *Refactory* dan pembenahan *bug* yang cepat
- c. *Tool* baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaanm serta kompetibelitas apkiasi dengan cepat
- d. Menfukung Proguard And App-sinning untuk keamanan
- e. Memiliki GUI apliasi android lebih mudah

Dukungan C++ dan NDK



- g. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan

Struktur folder Project Android Studio dapat dilihat pada gambar dibawah:

1. *Manifest*

Pada folder pertama yaitu *manifest*, yang berisi file *AndroidManifest.xml*. File ini berisi informasi tentang *package* (paket) pada proyek, yang didalamnya terdapat komponen Aplikasi seperti, *Activity*, *Service*, *User Permission*, *Content Provider* dan lain-lain.

2. *Java*

Pada folder kedua yaitu *Java*, yang terdapat nama *package* berisi *class Activity java*, biasanya berisi:

- *package* sebagai pengelompokkan beberapa *file class Activity java*
- *import* yang digunakan untuk mengimpor *library* untuk komponen Aplikasi
- *public class MainActivity extends AppCompatActivity* dimana, *public class* adalah *modifier* yang digunakan dalam pemrograman java untuk menyatakan bahwa *class/method/attribute* tersebut dapat diakses oleh kelas lain. *Main Activity* adalah nama dari kelas tersebut dan *extends AppCompatActivity* menyatakan bahwa *Class Main Activity* akan mewariskan sifat dari *library* yang bernama *AppCompatActivity*, yang merupakan sifat dari program java yaitu *inheritance* (pewarisan).



- *Protected void, protected* adalah *modifiler* yang digunakan dalam pemograman java untuk menyatakan bahwa kelas tersebut dapat diakses dari kelas turunannya. Sedangkan *void* adalah *method* yang digunakan dalam pemograman java untuk menyatakan bahwa, tipe data tersebut tidak mempunyai nilai balik (*return value*).
- *onCreate*, fungsi ini akan dipanggil ketika sebuah class baru dibuat, untuk menampilkan Activity yang di olah di (*Bundle savedInstanceState*) dan akan menampilkan sebuah halaman *user interface* pada *layout* yang bernama *activity_main* yang berlokasi di *R.layout.activity_main*.

3. *Res*

Res berisi semua sumber daya untuk menyimpan file-file gambar, menu, *file layout* sebagai tempat membuat *User Interface* (antarmuka), *mipmap* untuk menyimpan *file icon launcher* dan *values* yang digunakan untuk menyimpan file-file seperti *string.xml*, *styles.xml* dan *color.xml*

4. *Gradle*

Gradle adalah *build tools* yang digunakan pada Android Studio, untuk men *compile-run project* Aplikasi. Pada *build.gradle* digunakan untuk mengidentifikasi versi OS SDK, *library* yang akan digunakan.

