

## IMPLEMENTASI GOOGLE TEXT TO SPEECH PADA APLIKASI PENDETEKSI UANG BERBASIS ANDROID

Nafi' Arrizqi<sup>\*</sup>), Imam Santoso dan Yosua Alvin Adi Soetrisno

Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof.Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*</sup>E-mail: [arrizqinafi@gmail.com](mailto:arrizqinafi@gmail.com)

### Abstrak

Text to Speech (TTS) adalah teknologi yang digunakan untuk mengubah teks menjadi ucapan, salah satu speech engine TTS adalah Google TTS. Google TTS diterapkan pada aplikasi pendeteksi uang memudahkan pengguna low vision mengenali nominal uang melalui suara dan tap action. Masalah yang muncul adalah hasil deteksi tidak selalu benar, sehingga Fitur TTS diatur untuk mengucapkan hasil deteksi hanya jika confidence level-nya tinggi untuk memperkecil resiko penyampaian kesalahan informasi deteksi ke pengguna. Confidence level adalah nilai probabilitas adanya objek di dalam bounding box. Oleh karena itu, perlu adanya analisis threshold dari confidence level hasil deteksi untuk memisahkan antara confidence level tinggi dan rendah. Algoritma K-means digunakan untuk menentukan threshold tersebut dan didapatkan nilai sebesar 83,62%. Selain memberikan keluaran suara pada hasil deteksi, fitur TTS juga dipasang untuk memberikan tanda telah terbukanya aplikasi, petunjuk cara penggunaan, tanda proses deteksi dimulai, dan petunjuk untuk mengarahkan uang agar hasil deteksi akurat melalui ucapan kepada pengguna. Tingkat kebenaran informasi yang diucapkan pada proses deteksi sebesar 98,6% jika seluruh bagian uang masuk frame, 98,6% jika setengah sisi kiri bagian uang masuk frame, dan 92,8% jika setengah bagian kanan uang masuk frame.

*Kata kunci: Text to Speech, API Google TTS, confidence level, threshold*

### Abstract

*Text to Speech (TTS) is a technology to convert text into speech, one of the TTS speech engines is Google TTS. Google TTS is applied to money-detection application that makes low vision recognize nominal money through voice and tap actions. The detection results are not always correct, so the TTS feature is enabled to speak the results only when the confidence level is high to minimize the misinformation. The confidence level is the probability value of an object in the bounding box. Therefore, it is necessary to analyze the threshold of the confidence level to separate the high and low confidence levels. The K-means algorithm is used to determine the threshold and gets 83.62% value. TTS feature is also installed to provide an indication of the application opening, how to use instructions, start processing sign, and instructions to direct money so the speech detection results are accurate. The level of correctness of the information spoken in the detection process is 98.6% if all parts of the money in frame, 98.6% if the left half of the money in frame, and 92.8% if the right half of the money in frame.*

*Keywords: Text to Speech, API Google TTS, confidence level, threshold*

### 1. Pendahuluan

Text to Speech (TTS) adalah teknologi yang memungkinkan komputer untuk mengubah teks tertulis menjadi ucapan [1]. Terdapat dua prinsip yang mendasarinya yaitu merubah teks ke fonem dan merubah fonem ke ucapan. Pertama, teks berisi kalimat akan dikonversikan menjadi rangkaian kode suara yang biasanya diwakili oleh kode fonem, durasi, dan nada. Setelah itu, *kode fonem, durasi, dan nada* yang dihasilkan di bagian sebelumnya dikonversi untuk menghasilkan sinyal suara yang cocok dengan *input* [2]. *Text to Speech* ini berbeda dengan *Speech to Text* atau sering pula disebut dengan *Speech Recognition*. *Speech Recognition* adalah

teknologi yang memungkinkan komputer untuk menerima *input* dalam bentuk bahasa lisan. Ucapan tersebut dikonversi menjadi sinyal digital dengan mengubah gelombang suara menjadi serangkaian angka, kemudian menyesuaikan dengan kode tertentu dan mencocokkan pola yang tersimpan di perangkat. Hasil pengenalan bahasa lisan dapat ditampilkan dalam bentuk tertulis atau dapat dibaca oleh peralatan teknis [3]. TTS telah banyak diimplementasikan dalam berbagai bidang utamanya untuk membantu mereka yang memiliki keterbatasan dalam penglihatan. Telah terdapat banyak *speech engine* yang digunakan untuk mengimplementasikan TTS, salah satunya adalah *Google Text to Speech*. Seorang pengembang dapat mengubah teks ke ucapan melalui

tampilan dan kelas yang telah disediakan oleh Google API [4].

Terdapat banyak penelitian yang telah dilakukan sebelumnya terkait dengan penerapan *speech engine* Google TTS pada aplikasi *mobile*. API Google *Text to Speech* diterapkan untuk menghasilkan *output* suara pada aplikasi yang dibangunnya agar tunanetra tetap dapat mengenali objek di sekitarnya [5]. Selain itu terdapat pula penelitian yang memudahkan pengguna mengetahui isi pesan yang diterima tanpa harus membaca isi pesan tersebut [6]. Penelitian serupa diterapkan juga untuk membantu pengguna mengetahui pelafalan bahasa Inggris dengan benar [2], membaca dongeng [7], membantu belajar iqro' [8], dan aplikasi konversi teks menjadi suara berbasis android [9]. Teknologi TTS ini sangat memudahkan pekerjaan serta menjadi jembatan bagi mereka yang memiliki keterbatasan dalam penglihatan.

Berdasarkan penjelasan di atas, maka fitur TTS akan diterapkan pada aplikasi pendeteksi uang. Aplikasi tersebut dibuat dengan bahasa Java pada android studio yang memungkinkan pengguna *low vision* mengenali nominal uang kertas dengan mudah melalui suara dan *tap action*. Masalah yang muncul adalah terkadang hasil deteksi tidak selalu akurat. Fitur TTS disini diatur untuk mengucapkan hasil deteksi hanya ketika *confidence level*-nya memiliki nilai yang tinggi untuk meminimalkan resiko penyampaian kesalahan informasi deteksi kepada pengguna. *Confidence level* adalah nilai probabilitas adanya objek di dalam *bounding box*. Semakin tinggi *confidence level*, maka semakin tinggi pula kemungkinan terdapat objek pada *bounding box*, sehingga sistem lebih andal untuk mengenali objek tersebut dengan benar.

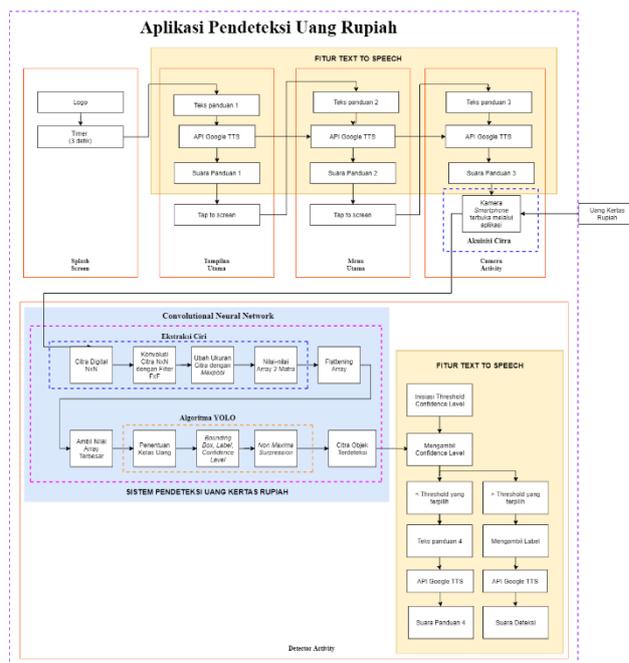
Fitur TTS akan diatur untuk mengucapkan hasil deteksi hanya jika *confidence level*-nya tinggi untuk memperkecil resiko penyampaian kesalahan informasi deteksi ke pengguna. Oleh karena itu, perlu adanya analisis nilai ambang batas atau *threshold* dari *confidence level* hasil deteksi untuk memisahkan antara *confidence level* yang tinggi dan rendah. Proses dalam menentukan *threshold* tersebut dilakukan dengan menggunakan metode algoritma K-Means. Apabila nilai *threshold* yang dipasang sudah sesuai, maka probabilitas hasil keluaran TTS akan akurat sesuai dengan nominal yang ingin dideteksi.

Fitur ini dipasang pula pada beberapa halaman di aplikasi pendeteksi uang untuk memberikan tanda telah terbukanya aplikasi, petunjuk cara penggunaan, tanda proses deteksi dimulai, dan petunjuk untuk mengarahkan uang agar hasil deteksi akurat melalui suara kepada pengguna. Pengguna yang memiliki keterbatasan penglihatan tidak akan kesulitan dalam mengoperasikan aplikasi tersebut. Fitur TTS yang diterapkan pada aplikasi pendeteksi uang akan diamati performanya melalui pengujian kebenaran informasi deteksi melalui suara.

## 2. Metode

### 2.1. Desain Sistem

Secara keseluruhan, sistem yang dirancang adalah sistem yang dapat memberikan panduan penggunaan aplikasi melalui suara dan memberikan hasil deteksi yang benar melalui ucapan pada aplikasi pendeteksi uang berbasis android. Desain sistem dapat dilihat pada Gambar 1.



Gambar 1. Desain Sistem Fitur TTS pada aplikasi pendeteksi Uang Rupiah

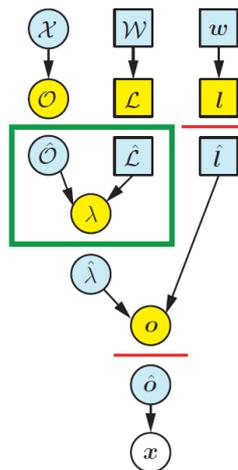
Gambar 1 memperlihatkan bahwa fitur *text to speech* merupakan bagian dari sistem pendeteksi uang. Fitur ini merupakan keluaran sistem berupa suara untuk mempermudah pengguna *low vision*. Fitur ini merupakan keluaran sistem secara keseluruhan berupa suara untuk mempermudah penggunaan aplikasi bagi mereka yang memiliki keterbatasan penglihatan. *Text to speech* dipasang pada 4 *class activity* yaitu tampilan utama, menu utama, *camera activity*, dan *detector activity*. Berdasarkan jenis masukannya, fitur ini dibedakan menjadi dua yaitu fitur TTS dengan *input* teks langsung dan fitur TTS dengan *input* teks yang mengambil dari label hasil deteksi secara *realtime*. TTS yang dipasang pada *detector activity* akan memberikan *output* hasil deteksi yang telah diatur *threshold*-nya. Apabila hasil *confidence level* di atas *threshold* yang diatur, maka TTS hasil deteksi aktif. Namun apabila tidak, maka sistem akan mengaktifkan fitur TTS panduan untuk mengarahkan pengguna agar hasil deteksi memiliki *confidence* yang tinggi sehingga peluang hasil deteksi akurat lebih tinggi.

2.2. Perancangan Fitur TTS

2.2.1. Parametric Text to Speech

Parametric text to speech menghasilkan data audio melalui algoritma pemrosesan sinyal yang dikenal sebagai vocoder [10]. Vocoder adalah salah satu kategori dari voice codec yang menganalisis dan mensintesis sinyal suara manusia untuk kompresi, multipleksing, enkripsi suara, atau transformasi suara. Dalam parametric text to speech, semua informasi yang diperlukan untuk menghasilkan data disimpan dalam parameter model. Salah satu metode dalam teknologi parametrik TTS ini adalah Hidden Markov Model (HMM). Google masih menggunakan metode ini untuk menerapkan fitur TTS dengan tipe suara standar. HMM adalah salah satu model TTS generatif. Model ini mewakili proses di balik produksi ucapan yang dilatih untuk meminimalkan kesalahan terhadap ucapan yang dihasilkan. Model generatif merupakan learned model yang bertujuan untuk memaksimalkan jumlah informasi yang akan diterima [11].

Pemaksimalan langkah dalam parametric TTS dengan metode HMM disajikan pada Gambar 2 [11].



Gambar 2. Maksimalisasi langkah parametric TTS

Proses yang terjadi pada diagram di atas dijelaskan pada Tabel 1 berikut [11].

Tabel 1. Proses maksimalisasi langkah HMM

Variabel	Proses
$\hat{O}$	Analisis vocoder dengan mengekstrak acoustic features maksimal
$\hat{L}$	Analisis teks dengan mengekstrak linguistic features maksimal
$\hat{\lambda}$	Train HMM dari hasil analisis vocoder dan analisis teks maksimal
$\hat{i}$	Analisis teks dari masukan dan memprediksi linguistic features maksimal
$\hat{o}$	Memprediksi acoustic feature dari parameter yang tersedia
$\hat{x}$	Sintesis vocoder berupa suara tersintesis hasil prediksi

Dengan  $x$  adalah suara yang tersintesis,  $w$  adalah teks yang diberikan,  $X$  menunjukkan data gelombang suara,  $W$  menunjukkan data transkripsi,  $O$  dan  $o$  adalah acoustic features,  $L$  dan  $l$  adalah linguistic features, dan  $\lambda$  adalah model training

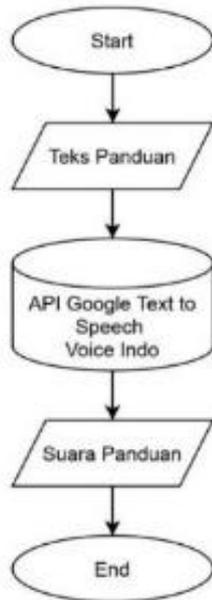
2.2.2. API Google TTS

Text to speech memungkinkan untuk mengubah kata dan kalimat menjadi data audio yang dikodekan base64. Konfigurasi yang dilakukan dalam sintesis ucapan ini adalah dengan memilih jenis masukan, bahasa, negara, dan kecepatan bicara. Jenis input yang digunakan pada fitur TTS ini adalah teks biasa. Kode bahasa dan negara dapat dilihat pada cloud google. Adapun kode bahasa Indonesia adalah "id-ID", yang mana "id" menunjukkan kode bahasa sedangkan "ID" menunjukkan kode negara. Sehingga suara dan logat yang dihasilkan pada fitur TTS ini akan sesuai dengan keinginan. Kode bahasa dan negara tersebut dimasukkan pada method setLanguage yaitu setLanguage(new Locale("id","ID")).

Dalam pengaturan TTS menggunakan API Google TTS, kecepatan durasi dapat diatur dengan menggunakan setSpeechRate(float speechRate) [12]. Kecepatan ucap untuk berbicara dengan jelas pada laki-laki adalah 263,4 spm, sedangkan pada perempuan adalah 227,9 spm [13]. Kecepatan ucap ini akan diatur sedemikian rupa agar suara yang didengar tepat, tidak terlalu cepat dan tidak terlalu lambat. Apabila tidak dilakukan pengaturan kecepatan, fitur TTS tetap dapat berjalan dengan kecepatan default yaitu 212 spm. Kecepatan dapat diatur berkisar dari 0,25 kali default hingga 4 kali default, apabila pengaturan diluar kisaran tersebut maka suara tidak akan terdengar jelas. Ketika parameter speechRate diatur menjadi 0.5, maka kecepatan ucap akan melambat dengan kecepatan setengah kali pengaturan default. Ketika parameter speechRate diatur menjadi 2, maka kecepatan ucap akan dipercepat dengan kecepatan dua kali pengaturan default [14]. Setelah dilakukan beberapa kali percobaan, maka kecepatan suara yang sesuai untuk fitur TTS ini adalah 1.4 kali default

Google TTS juga menyediakan output suara laki-laki. Untuk dapat mengubah pengucap laki-laki dapat dilakukan dengan menambah konfigurasi pada source code TTS di program java yaitu add("male") [15]. Apabila tidak, maka suara yang dihasilkan adalah suara perempuan. Selain itu, Google TTS juga bisa diatur dengan suara sendiri atau biasa disebut dengan custom voice. Untuk bisa mendapatkan fasilitas tersebut, customer harus terlebih dahulu melakukan pembayaran billing Google Cloud dan mengisi formulir. Customer juga perlu melakukan perekaman suara sendiri untuk nantinya dikirimkan ke pihak developer Google [16]. Fitur TTS pada aplikasi ini menggunakan jenis suara default yaitu perempuan dengan dataset dari Google TTS.

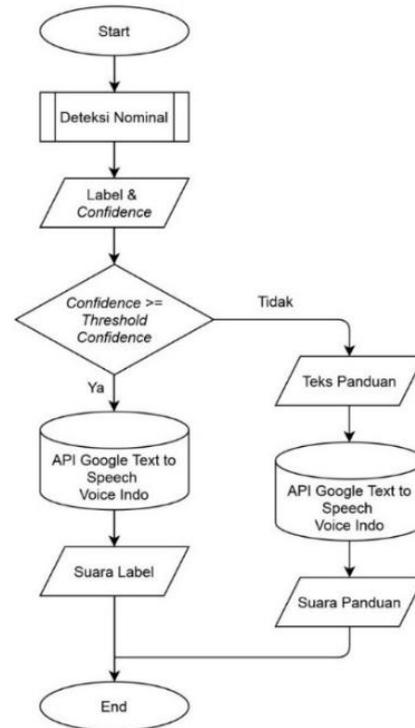
2.2.3. Rancangan Implementasi TTS



Gambar 3. Flowchart fitur TTS teks panduan

Teks sebagai masukan dalam TTS ini berupa panduan untuk memudahkan pengguna *low vision* menjalankan aplikasi. Teks yang pertama berbunyi “Selamat datang di aplikasi pendeteksi uang. Aplikasi ini dapat mendeteksi nominal uang rupiah dari seribu hingga seratus ribu. Ketuk sekali pada layar untuk melanjutkan”. Teks tersebut akan dikonversi menjadi suara pada *activity* “tampilan utama”. Setelah diketuk, aplikasi akan berpindah ke “menu utama” dengan teks yang berbunyi “Anda dapat menggunakan aplikasi ini hanya dengan mengarahkan kamera ke uang yang ingin dideteksi. Ketuk sekali pada layar untuk melanjutkan”. Kemudian setelah diketuk, halaman akan berpindah ke “camera activity”. Disana kamera akan aktif dan proses pendeteksian akan dimulai yang ditandai dengan bunyi “Kamera telah aktif. Jika suara belum terdengar maka uang belum terdeteksi”. Setelah itu, *camera activity* akan tersinkron dengan *detector activity* dan memunculkan hasil deteksi.

Proses pengkonversian nominal uang hasil deteksi menjadi suara dapat dilihat pada Gambar 4. Dapat dilihat bahwa API Google TTS mendapatkan *input* dari label hasil deteksi yang muncul di layar. Apabila label memiliki *confidence* lebih dari *threshold*, maka sistem akan mengkonversikan label ke suara. Namun jika *confidence* kurang dari *threshold* maka sistem akan memberikan isyarat suara yang berbunyi “Arahkan uang tepat didepan kamera”. Ini berguna agar *low vision* tahu bahwa uang sudah tertangkap kamera namun hasil akurasi buruk sehingga tidak dapat memberikan *output* hasil deteksi



Gambar 4. Flowchart fitur TTS dengan *input* label

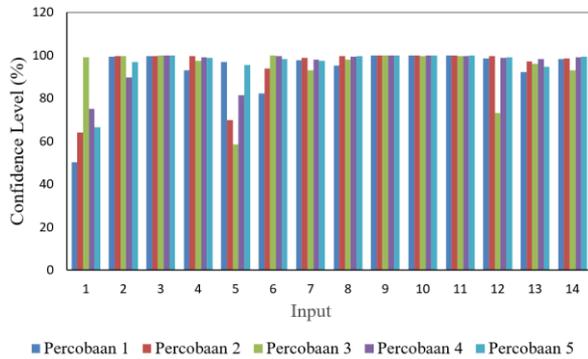
2.2.4. Penyetelan *Confidence Threshold* Deteksi

*Confidence Threshold* deteksi perlu juga diatur dalam implementasi sistem *Text to Speech* karena menjadi parameter keakuratan suara yang disampaikan oleh aplikasi. Apabila tidak diatur, maka aplikasi dapat memberikan keluaran suara saat hasil deteksi memiliki *confidence* rendah, sehingga informasi yang diberikan ke pengguna kurang akurat. Pengaturan *threshold* ini menggunakan metode algoritma perumpunan K-Means

Objek deteksi dalam masukan sistem adalah uang kertas rupiah emisi 2016, sisi depan dan belakangnya. Data dikumpulkan dengan melakukan *testing* sistem pendeteksi uang kertas ke beberapa data set. Dalam pengambilan data, kondisi lingkungan dan skenario perekaman pada saat pengambilan data diatur agar bisa dilakukan perbandingan. Kondisi yang sama tersebut yaitu berupa pencahayaan, kamera, serta objek deteksi yang digunakan. Sementara skenario dilakukan dengan mengarahkan kamera ke seluruh bagian uang.

Tujuan dari pengambilan data ini adalah untuk memisahkan antara *confidence level* yang tinggi dan rendah. Semakin tinggi *confidence level* dari citra hasil deteksi, maka sistem akan semakin percaya bahwa terdapat objek yang dideteksi disana, sehingga bisa dikatakan hasil deteksi semakin akurat. Parameter yang diamati di sini adalah *confidence level* dan label dari citra hasil deteksi. Label yang muncul akan diuji validasi kebenarannya.

Label yang benar maka memiliki validasi 1 sedangkan label yang salah memiliki validasi 0. Sebaran data *confidence level* pada tiap *testing* dapat diamati pada Gambar 5.



Gambar 5. Sebaran data *confidence level*

Untuk bisa memisahkan antara *confidence level* yang benar dan salah menggunakan metode perumpunan K-means, terlebih dahulu harus menentukan jumlah K-nya. Kelas yang akan di kelompokkan adalah kelas benar dan salah, sehingga terdapat dua kelompok, maka  $K = 2$ . Berdasarkan data yang diambil, maka kita dapat menentukan *centroid* secara acak. Data diurutkan terlebih dahulu dari data terkecil hingga besar. Kali ini *centroid* yang dipilih adalah 64,02 dan 98,55.

Dengan melakukan perhitungan pada matlab, maka didapatkan perumpunan untuk kelas 1 atau kelas yang salah adalah *confidence level* yang bernilai.

$$K1 = \{50.21, 58.47, 64.02, 66.58, 69.71, 73.14, 75.14, 81.51, 82.36\}$$

Sementara itu kelas untuk *confidence level* yang benar adalah sebagai berikut.

$$K2 = \{89.56, 92.03, 93.02, 93.04, 93.06, 93.74, 94.54, 95.24, 95.52, 96.13, 96.72, 96.87, 97.15, 97.32, 97.37, 97.70, 97.96, 98.07, 98.25, 98.35, 98.36, 98.44, 98.55, 98.74, 98.86, 98.90, 98.92, 98.95, 99.05, 99.10, 99.21, 99.27, 99.29, 99.48, 99.49, 99.54, 99.59, 99.62, 99.63, 99.65, 99.68, 99.68, 99.69, 99.71, 99.72, 99.81, 99.82, 99.83, 99.84, 99.91, 99.93, 99.93, 99.94, 99.94, 99.94, 99.96, 99.96, 99.96, 99.97, 99.98, 99.98\}$$

Nilai *confidence* tertinggi kelas 1 adalah 82,36 dan nilai *confidence* terendah kelas 2 adalah 89,56. Jarak antara

kedua nilai tersebut menjadi batasan antara kedua rumpun. *Threshold* atau ambang batas dapat dihitung dengan merata-rata kedua kelas.

$$\mu_1 = \frac{\text{Jumlah confidence level K1}}{9} = 69.02$$

$$\mu_2 = \frac{\text{Jumlah confidence level K2}}{61} = 98.22$$

Untuk menghitung *threshold*, kedua rerata tersebut akan dirata-rata kembali.

$$T = \frac{\mu_1 + \mu_2}{2} = \frac{69,02 + 98,22}{2} = 83,62$$

Pada algoritma K-means, *threshold* dapat berubah jika perumpunan belum sempurna akibat dari inisiasi *centroid* yang kurang tepat. Namun pada perhitungan menggunakan Matlab, *threshold* tersebut sudah stabil sehingga *confidence threshold* final yang didapatkan bernilai 83,62%.

Ketika sistem menjalankan proses pendeteksian, maka fitur TTS akan menangkap *confidence level* dan label dari citra hasil deteksi. Jika *confidence level* lebih dari ambang batas yaitu 83,62%, sistem akan mengambil label hasil deteksi dan memasukkannya sebagai *input* ke dalam Google TTS, sehingga aplikasi pendeteksian uang dapat mengeluarkan suara hasil deteksi yang benar.

### 2.2.5. Penyetelan TTS saat *Confidence* kurang dari *Threshold*

Masukan untuk sistem TTS di bawah *threshold* ini berupa teks. Isi perintah tersebut dianalisis dari kondisi uang saat validasi bernilai 0 atau label deteksi yang muncul di layar tidak benar. Hasilnya dapat diamati pada Tabel 2.

Tabel 2. Akurasi pada Tiap Perlakuan Uang

Kondisi Uang yang Masuk Frame	Banyak
1/2 Sisi Kiri	9
1/2 Sisi Kanan	12
1/2 Sisi Tengah	4
3/4 Sisi Kiri	2
3/4 Sisi Kanan	2
Full / Normal	0

Berdasarkan Tabel 2, kondisi uang yang kerap terjadi kesalahan deteksi adalah saat uang yang masuk *frame* hanya separuh sisi kanan saja yaitu sebanyak 12 kasus. Disusul oleh separuh sisi kiri uang yang masuk *frame* yaitu 9 kasus. Berikutnya adalah separuh sisi tengah dengan 4 kasus, dan 3/4 sisi baik kiri maupun kanan masing-masing 2 kasus. Adapun saat uang yang masuk *frame* adalah *full body* atau kondisi normal, tidak terdapat kekeliruan pendeteksi. Ini dikarenakan *dataset* yang dipakai dalam proses *training* sistem deteksi kebanyakan adalah dalam keadaan normal dan *full body*, sehingga *confidence* pada

saat uang masuk semua ke dalam *frame* dan dalam keadaan normal memiliki harga yang tinggi. Oleh karena itu, teks masukan yang sesuai untuk sistem *Text to Speech* saat *confidence* kurang dari *confidence threshold* pendeteksi yaitu 83,62% adalah “Pastikan seluruh bagian uang masuk ke dalam *frame* kamera”. Dengan penyetelan perintah suara seperti itu, maka pengguna dapat mengetahui tindakan yang harus dilakukan agar hasil deteksi nya benar.

### 3. Hasil dan Analisa

#### 3.1. Pengujian Keberhasilan Fitur TTS

Pengujian ini dilakukan dengan mengecek keberhasilan implementasi Google *Text to Speech* dalam sistem deteksi uang kertas rupiah.

Tabel 3. Tingkat keberhasilan implementasi

Input	K1	K2	K3	K4	Keberhasilan
Teks 1	√	√	√	√	100%
Teks 2	√	√	√	√	100%
Teks 3	√	√	√	√	100%
Teks 4	√	√	√	√	100%
Seribu	√	√	√	√	100%
2 ribu	√	√	√	√	100%
5 ribu	√	√	√	√	100%
10 ribu	√	√	√	√	100%
20 ribu	√	√	√	√	100%
50 ribu	√	√	√	√	100%
100 ribu	√	√	√	√	100%
Se ribu	√	√	√	√	100%
Dua ribu	√	√	√	√	100%
Lima ribu	√	√	√	√	100%
Sepuluh ribu	√	√	√	√	100%
Dua puluh ribu	√	√	√	√	100%
Lima puluh ribu	√	√	√	√	100%
Seratus ribu	√	√	√	√	100%

Keterangan:

- K1 : Bahasa Indonesia
- K2 : Negara Indonesia
- K3 : Suara Wanita
- K4 : *Speech rate* yaitu 1,4 kali *default*
- Teks 1 : Selamat datang di aplikasi pendeteksi uang. Aplikasi ini dapat mendeteksi nominal uang rupiah dari seribu hingga seratus ribu. Ketuk sekali pada layar untuk melanjutkan
- Teks 2 : Anda dapat menggunakan aplikasi ini hanya dengan mengarahkan kamera ke uang yang ingin dideteksi. Ketuk sekali pada layar untuk melanjutkan
- Teks 3 : Kamera telah aktif. Jika suara belum terdengar maka uang belum terdeteksi
- Teks 4 : Pastikan seluruh bagian uang masuk ke dalam *frame* kamera

Berdasarkan pengujian yang disajikan pada Tabel 3, masing-masing *input* diproses dalam sistem *Text to Speech* dan menghasilkan tingkat keberhasilan yang sempurna.

Hal ini karena sistem *Text to Speech* menggunakan *framework Google API* dalam implementasinya. *Framework* ini sudah sangat baik untuk mengucapkan kalimat dalam bahasa dan logat Indonesia.

#### 3.2. Pengujian Kebenaran Informasi Suara Deteksi

Pengujian dilakukan dengan mengecek kebenaran informasi deteksi yang diucapkan setelah *confidence threshold* = 83,62% dipasang. Sistem diatur untuk mengaktifkan fitur TTS hasil deteksi jika *confidence level* lebih dari *threshold* yang ditetapkan. Jika tidak maka fitur TTS panduan yang berbunyi “Pastikan seluruh bagian uang masuk ke dalam *frame* kamera” aktif. Suara yang keluar dari aplikasi dicatat untuk dijadikan pembandingan kebenaran keluaran sistem. Dalam melakukan pengujian ini, objek deteksi berupa 14 uang kertas rupiah akan divariasikan menjadi 3 kondisi yaitu semua bagian masuk ke dalam *frame*, sebagian uang masuk ke dalam *frame* (sisi kanan), dan sebagian uang masuk ke dalam *frame* (sisi kiri) dalam 5 kali percobaan.

Tabel 4. Tingkat kebenaran informasi TTS

Bagian Uang yang Masuk Frame	Jumlah Kesalahan	Kebenaran Informasi
Seluruh bagian	1	98,6%
Setengah bagian sisi kanan	5	92,8%
Setengah bagian sisi kiri	1	98,6%

Berdasarkan Tabel 4, ketika seluruh bagian uang masuk dalam *frame*, sistem hanya mengalami satu kali pengucapan kesalahan informasi per 70 kali percobaan atau kebenaran informasi yang diucapkan mencapai 98,6%. Ketika hanya setengah bagian kanan yang masuk *frame*, pengucapan informasi yang salah terjadi 5 kali per 70 kali percobaan atau kebenaran informasi yang diucapkan mencapai 92,8%. Dan ketika hanya setengah bagian kiri uang yang masuk *frame*, sistem hanya mengalami satu kali pengucapan kesalahan informasi per 70 kali percobaan sehingga kebenaran informasi yang diucapkan mencapai 98,6%. Walaupun kesalahan informasi hampir terbilang rendah, namun saat hanya sebagian uang yang masuk ke dalam *frame*, sistem akan lebih sering mengucapkan TTS panduan yang mengarahkan pengguna untuk memastikan seluruh bagian uang masuk ke dalam *frame*. Hal ini dikarenakan saat uang yang masuk *frame* hanya sebagian saja memiliki *confidence level* yang rendah.

### 4. Kesimpulan

Berdasarkan analisis penyetelan *confidence threshold*, batas optimum *confidence* yang didapatkan adalah 83,62% Apabila *confidence* hasil deteksi lebih dari 83,62% maka sistem akan mengaktifkan fitur TTS deteksi. Apabila tidak, maka sistem akan mengaktifkan fitur TTS panduan yang mengarahkan agar hasil deteksi akurat. Fitur TTS dengan *speech engine API Google TTS* Ketika seluruh bagian uang

masuk ke dalam frame, maka rata-rata kebenaran informasi yang diucapkan mencapai 98,6%, ketika hanya bagian kanan yang masuk frame maka kebenaran informasi yang diucapkan mencapai 92,8%, ketika hanya bagian kiri yang masuk frame maka kebenaran informasi yang diucapkan mencapai 98,6%,

## Referensi

- [1]. I. C. Nwakanma, O. Izunna dan I. Oluigbo, "Text – To – Speech Synthesis (TTS)," *IJRIT International Journal of Research in Information Technology*, vol. 2, no. 5, pp. 154-163, 2014.
- [2]. J. A. Pramono, "Rancang Bangun Aplikasi Text to Speech Sebagai Alat Bantu Pembelajaran Bahasa Inggris," Sekolah Tinggi Manajemen Ilmu Komputer Amikom, Yogyakarta, 2013.
- [3]. F. Jelinek, "Statistical Methods for Speech Recognition," Massachusetts Institute of Technology, United States, 1997.
- [4]. A. Irawan, "Implementasi Speech To Text Dan Text To Speech Pada Aplikasi Traveling Conversations Assistance Berbasis Mobile," 2014.
- [5]. M. Fachriyan dan D. Dian, "Pembangunan Aplikasi Pengenalan Objek Terdekat untuk Penyandang Tunanetra Menggunakan Mlkit dan Text to Voice Berbasis Android," Teknik Informatika – Universitas Komputer Indonesia, Bandung.
- [6]. N. Muslimin, "Sistem Pembaca Pesan Text to Speech Berbasis Android Menggunakan Framework Google API," Teknik Informatika Universitas Hasanuddin, Makassar, 2017.
- [7]. U. Hasanah dan Jaroji, "Aplikasi Dongeng Berbasis Text to Speech," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 7, no. 2, pp. 82-88, 2016.
- [8]. M. F. Arifin, F. Marisa dan I. D. Wijaya, "Implementasi Google Speech Untuk Penentuan Level Pembelajaran Iqro' Berbasis Android," Teknik Informatika Universitas Widyagama, Malang.
- [9]. I. Rinaldy, "Aplikasi Speech Recognition dan Text to Speech Berbasis Android," Fakultas Sains dan Teknologi UIN Alauddin, Makassar, 2017.
- [10]. "Standard and WaveNet voices," Google Cloud, [Online]. Available: <https://cloud.google.com/text-to-speech/docs/wavenet>. [Diakses 23 Agustus 2021].
- [11]. H. Zen, "Generative Model-Based Text-to-Speech Synthesis," Google London, London, 2017.
- [12]. "Text to Speech," Android Developer, [Online]. Available: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>. [Diakses 9 Agustus 2021].
- [13]. J. Yoo, H. Oh, S. Jeong dan I.-K. Jin, "Comparison of Speech Rate and Long-Term Average Speech Spectrum between Korean Clear Speech and Conversational Speech," *Journal of audiology and otology*, vol. 23, no. 4, pp. 187-192, 2019.
- [14]. "Method: text.synthesize," Google Cloud, [Online]. Available: <https://cloud.google.com/text-to-speech/docs/reference/rest/v1/text/synthesize>. [Diakses 23 Agustus 2021].
- [15]. S. Gaddelpalliwar, "Android Text To Speech Male Voice," Stackoverflow, 20 Oktober 2018. [Online]. Available: <https://stackoverflow.com/questions/9815245/android-text-to-speech-male-voice>. [Diakses 9 Agustus 2021].
- [16]. "Text to Speech," Google Cloud, [Online]. Available: <https://cloud.google.com/text-to-speech>. [Diakses 9 Agustus 2021].