

## **APLIKASI SPOXTECH UNTUK PENYANDANG TUNA RUNGU- WICARA MENGGUNAKAN ALGORITMA *HIDDEN MARKOV* MODEL DAN METODE *FINITE STATE AUTOMATA* (FSA)**

**Destivanesha Rina<sup>1</sup>, Fauziah<sup>2</sup>, Nur Hayati<sup>3</sup>**

Program Studi Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional  
destivanesharina@gmail.com<sup>1</sup>, fauziah@civitas.unas.ac.id<sup>2</sup>, nurhayati@civitas.unas.ac.id<sup>3</sup>

*Submitted October 16, 2020; Revised November 25, 2020; Accepted March 3, 2021*

### **Abstrak**

Komunikasi adalah hal yang penting untuk menjalin hubungan sosial. Dalam berkomunikasi, penyandang tuna rungu dan tuna wicara mengalami kesulitan untuk menyampaikan informasi apa yang ingin disampaikan. Aplikasi *Spoxtech* berbasis android merupakan aplikasi yang dirancang khusus untuk penyandang tuna rungu dan tuna wicara. Sesuai dengan namanya, *Spoxtech* diambil dari kata *speech to text – text to speech*. Aplikasi ini dapat memudahkan penggunaannya dalam berkomunikasi karena dilengkapi dengan menu suara yang berfungsi untuk mendeteksi suara yang diucapkan kemudian di konversi ke dalam bentuk teks dan menu ketik yang berfungsi untuk mendeteksi tulisan yang kemudian di konversi ke dalam bentuk suara. Aplikasi ini dibuat dengan menggunakan algoritma *Hidden Markov Model* dalam *generate* suara ke dalam bentuk teks dan menggunakan metode *finite state automata* dalam *generate* teks menjadi suara dengan melakukan pemenggalan dua tingkat. Hasil dari penelitian ini dilakukan dengan pengujian *whitebox* dan *blackbox*. Dalam penelitian ini juga dilakukan pengujian normalisasi teks pada menu ketik yaitu setiap kata yang mengandung satuan mata uang, waktu, suhu, dan angka. Nilai akurasi yang di dapat dalam pengujian aplikasi ini menggunakan bahasa Indonesia dengan pengujian pada menu suara dan menu ketik dan pengujian pemenggalan dua tingkat didapatkan nilai sebesar 100% dari 300 kata dalam bahasa Indonesia.

**Kata Kunci :** Aplikasi, Spoxtech, Finite State Automata, Android

### **Abstract**

*Communication is a very important thing to establish social relations. In communicating, deaf and mute persons have difficulty conveying what information they want to convey. The Android-based Spoxtech application is an application specially designed for deaf and speech impaired persons. As the name implies, Spoxtech is taken from the word speech to text - text to speech. This application can make it easier for users to communicate because it is equipped with a voice menu that functions to detect the spoken voice which is then converted into text and a typed menu which functions to detect writing which is then converted into voice form. This application was created using the Hidden Markov Model algorithm in generating sound into text and using the finite state automata method in generating text into sound by performing two-level decapitation. This results of this study were carried out by testing the whitebox and blackbox. In this research, the text normalization test on the typewriter menu was also carried out, namely each word containing currency, time, temperature, and number units. The accuracy value obtained in testing this application uses Indonesian by testing the voice menu and typing menu and testing the two-level decapitation, the value is 100% of the 300 words in Indonesian.*

**Keywords:** Android, Spoxtech, Finite State Automata, Android, Apps.

## **1. PENDAHULUAN**

Perkembangan teknologi dapat memberikan kemudahan dalam berkomunikasi bagi semua orang. *Smartphone* saat ini telah berkembang menjadi asisten pribadi bagi

seluruh masyarakat yang menggunakannya. Terdapat banyak aplikasi yang dapat diakses dengan mudah melalui telefon pintar dengan bentuk yang sederhana sehingga mudah dipahami bagi siapapun yang

menggunakannya. Namun, adanya keterbatasan indera dalam berkomunikasi akan membuat lawan bicara sulit memahami apa yang sedang di bicarakan. Oleh sebab itu, diperlukannya sebuah sistem yang dapat mempermudah para difabel untuk melakukan komunikasi secara mandiri dengan tidak memerlukan bantuan dari orang lain dalam berkomunikasi, sehingga komunikasi mandiri dapat dilakukan kapan saja dan dimana saja. Pada penelitian ini, aplikasi dibuat untuk membantu penyandang tuna rungu dan tuna wicara dalam berkomunikasi secara mudah menggunakan *smartphone* android.

Pemanfaatan *Text to speech* yang telah dilakukan yaitu dengan mengubah tulisan kedalam bentuk suara dan diharapkan dapat membantu penyandang tuna wicara dalam melakukan komunikasi. Pengguna aplikasi dapat menuliskan apa yang ingin disampaikan pada tombol *keyboard smartphone*. Secara otomatis aplikasi akan mengubah tulisan tersebut ke dalam bentuk suara [1].

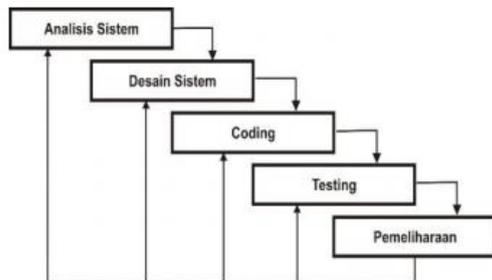
Berdasarkan penjelasan di atas maka dirancanglah sebuah aplikasi Spoxtech yang merupakan singkatan dari *speech to text – text to speech* aplikasi ini digunakan untuk membantu penyandang tuna rungu dan tuna wicara dalam berkomunikasi yang berbasis android. Aplikasi ini dapat memudahkan penggunaanya dalam berkomunikasi karena dilengkapi dengan menu suara yang berfungsi untuk mendeteksi suara yang diucapkan kemudian di konversi ke dalam bentuk teks dan menu ketik yang berfungsi untuk mendeteksi tulisan yang kemudian di konversi ke dalam bentuk suara. API *speech* Android merupakan sebuah *platform* yang cukup sederhana untuk para pengembang dalam melakukan integrasi kemampuan dalam pengenalan suara untuk aplikasi yang telah dibuat [6]. Algoritma Hidden Markov Model (HMM) digunakan dalam melakukan pengenalan suara pada *server* Google. *Smartphone* akan menerima masukan suara

yang kemudian akan dilakukan pengiriman pada *server* Google dan kemudian *server* Google akan melakukan proses pengenalan suara lalu mengubahnya kedalam bentuk teks dengan menggunakan algoritma Hidden Markov Model [8]. Pada menu ketik, dilakukan pengujian pemenggalan dua tingkat dengan menggunakan metode *Finite State Automata* (FSA). Metode *Finite State Automata* (FSA) dilakukan untuk proses pemenggalan kata menjadi suku kata. FSA merupakan suatu metode yang dapat melakukan pengenalan dan pengolahan suatu bahasa dengan pola-pola suku kata. Metode FSA melakukan penentuan huruf-huruf masukan berdasarkan pada fenom kata, dan kemudian akan menjadi dasar sebagai pembentukan pola suku kata dalam Bahasa Indonesia [11].

## 2. METODE PENELITIAN

Dalam metode penelitian ini terdiri dari beberapa tahap yaitu, studi literatur, perumusan masalah, pengembangan sistem dan penarikan simpulan. Studi literatur merupakan beberapa kegiatan yang meliputi metode dalam pengumpulan data pustaka, membaca, mencatat, dan melakukan olah bahan penelitian. Hal yang dilakukan studi adalah dengan mengumpulkan sejumlah buku buku, majalah yang memiliki kaitan dengan masalah dan tujuan dari penelitian, hal ini telah dilakukan oleh beberapa peneliti. Pada tahap pengembangan aplikasi, penulis memulai mengembangkan aplikasi. Pengembangan aplikasi dilakukan didasarkan pada metode pengembangan aplikasi. Selanjutnya dalam memilih metode pengembangan aplikasi, penulis memilih menggunakan metode *waterfall*. Pemilihan penggunaan metode model *waterfall* dikarenakan metode *waterfall* cukup simple dan cocok untuk penelitian skala kecil. Kelebihan dari model ini adalah kualitas dari aplikasi yang dihasilkan akan menjadi baik, karena pelaksanaannya dilakukan secara bertahap [5]. Untuk mencapai target suatu aplikasi dilakukan dengan berbagai cara

dengan mendesain aplikasi untuk memeriksa apakah aplikasi tersebut telah mencapai kebutuhan atau tidak. Pengembangan sistem dengan melakukan Analisa kebutuhan aplikasi, desain, pembuatan program dan pengujian aplikasi.



**Gambar 1. Metode Waterfall**

### **Analisa Kebutuhan Sistem**

Analisis sistem merupakan suatu proses dalam pengumpulan dan menginterpretasikan suatu kenyataan, mendiagnosa persoalan dan menggunakan keduanya dalam memperbaiki sistem. Analisis sistem (analisis informasi) merupakan orang yang menganalisis sistem atau mempelajari masalah yang timbul dan menentukan kebutuhan pemakai sistem untuk mengidentifikasi pemecahan permasalahan tersebut. Untuk itu dalam membuat aplikasi ini kebutuhan yang digunakan adalah :

1. Perangkat Keras (*Hardware*)  
Perangkat keras pada penelitian kali ini terdiri dari :
  - Laptop (Lenovo Thinkpad T510) dengan spesifikasi berikut
    - a. CPU Intel Core i7
    - b. RAM 8GB
    - c. HDD 1TB
    - d. NVIDIA 930MX
2. Perangkat Lunak (*Software*)  
Perangkat lunak pada penelitian kali ini terdiri dari :
  - Windows 10 64bit (Sistem Operasi)
  - Android Studio 4.0

### **Sistem Aplikasi**

Desain sistem yang akan digunakan dalam pembuatan aplikasi tersebut yaitu terdapat pada Android Studio pada bagian GUI. *Tools* yang akan digunakan dalam pembuatan aplikasi tersebut yaitu GUI. GUI merupakan singkatan dari *Graphical User Interface* yang merupakan aplikasi *display* dari Android Studio yang terdapat perintah, tugas, maupun komponen dalam program yang akan mempermudah *user* menjalankan sebuah program Android Studio. Tujuan dalam penggunaan GUI adalah untuk membuat program lebih terlihat praktis dan simpel bagi *end-user*. Oleh sebab itu terdapat hal yang harus diperhatikan dalam membuat GUI agar dapat menampilkan *ui (user interface)* yang *friendly* atau mudah dalam penggunaannya sekalipun *user* merupakan orang awam yang tidak akan mendapatkan kerumitan dalam program yang digunakan.

### **Pengkodean**

Penerapan kode program aplikasi Spoxtech ini menggunakan pemrograman Java Android dan menggunakan Google Speech API.

### **Pengujian**

Melakukan uji coba dengan menggunakan metode *blackbox* dan metode *whitebox*.

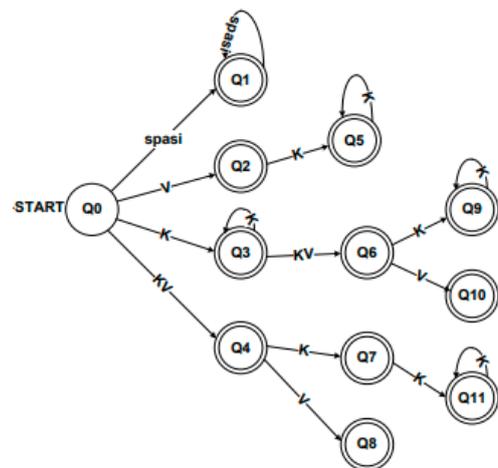
### **Pemeliharaan**

Melakukan perbaikan aplikasi berdasarkan evaluasi dari pengguna dan akan memperbaharukannya.

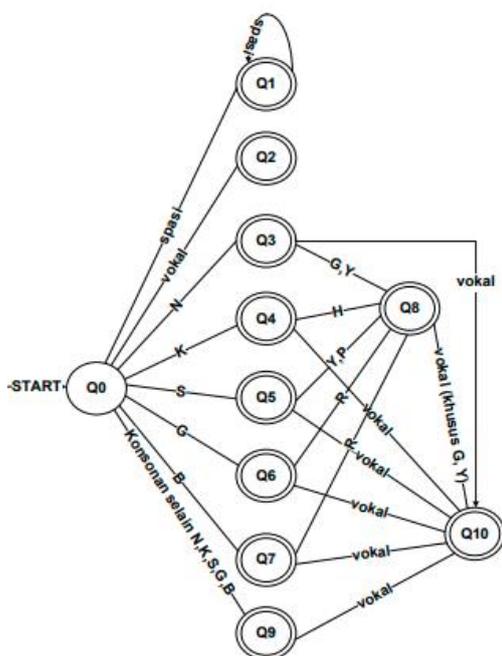
## **3. HASIL DAN PEMBAHASAN**

FSA merupakan model matematika yang dapat melakukan input dan output. FSA memiliki banyak *state* yang dapat berpindah

dari *state* satu ke yang lainnya dengan berdasarkan input dan fungsi transisi. FSA tidak memiliki tempat *memory* atau tempat penyimpanan dan FSA yang hanya dapat mengingat *state* [2]. FSA memiliki dua tingkatan yang digunakan untuk melakukan penelitian ini. Pada tingkatan yang pertama dikenali adalah pola V, K, dan KV. Hasil pengenalan FSA pada suatu tingkatan dapat dijadikan masukan untuk FSA tingkatan selanjutnya. Pada gambar 2 digambarkan diagram transisi FSA tingkatan pertama.



**Gambar 3. Diagram Transisi FSA Tingkatan Kedua**



**Gambar 2. Diagram Transisi FSA Tingkatan Pertama**

Pada tingkatan kedua FSA dapat mengenali 12 suku kata dengan pola V, VK, KV, VKK, KVK, KKV, KVKK, KKVK, KKKV, KKVKK, KKKVK, dan KVKKK. Dalam gambar 3 digambarkan diagram transisi FSA tingkatan kedua.

**Algoritma yang digunakan**

Algoritma Hidden markov Model merupakan algoritma yang sangat akurat dan efisien dalam melakukan pengenalan suara. Hidden Markov Model terdiri dari dua bagian, yaitu hidden model dan proses markov. Proses markov merupakan proses stokastik dengan properti markov. Pengenalan suara berbasis HMM menggunakan konsep bahwa ucapan terdiri dari beberapa urutan suara dasar. Kata apapun yang terdapat di dalam kamus terdiri dari beberapa urutan suara dasar. Masing-masing dari suara dasar itu memiliki model statistik. Oleh karena itu, sebuah kata dapat diekspresikan kedalam suatu urutan model statistik. Suara dijadikan contoh pengenalan suara untuk melakukan pembentukan urutan vektor pada fitur suara atau parameter numerik. Urutan vektor pada fitur suara ini merupakan urutan observasi. Hidden markov model biasa digunakan untuk aplikasi pada bidang pengenalan pola temporal (*temporal pattern recognition*) seperti bioinformatika, kompresi kalimat, ekonomi, finansial, pengenalan not balok, tulisan, *computer vision*, dan pengenalan suara. [6].

**Tampilan Aplikasi**

Dalam membuat tampilan aplikasi ini menggunakan xml. Pada gambar 4 merupakan tampilan dari aplikasi spoxtech.



Gambar 4. Tampilan Aplikasi Spoxtech

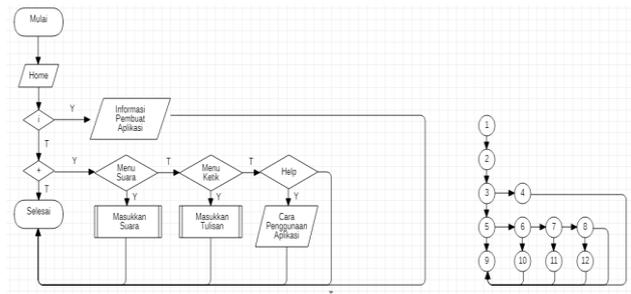
**Pengujian**

Terdapat beberapa pengujian yang telah dilakukan dalam penelitian ini diantaranya adalah melakukan pengujian *whitebox*, *blackbox*, pengujian aplikasi dalam beberapa kata, hasil normalisasi teks pada aplikasi spoxtech dan pengujian pemenggalan dua tingkat.

Pengujian *whitebox* merupakan pengujian dengan struktur aplikasi, dalam pengujian ini membuat *flowchart* dan *flowgraph*. *Flowchart* merupakan tahapan dengan menggambarkan logika program sedangkan *flowgraph* pengujian alur atau alir program. Pengujian ini juga menghitung *region* menghitung berdasarkan *predicate node* seperti *if else flowgraph*, *cyclomatic complexity* menghitung jumlah *path* dari *edge* dan *node*, terakhir *independent path*.

1. *Flowchart* dan *flowgraph* Menu Awal

Pada gambar 4 merupakan *flowchart* dan *flowgraph* menu awal.



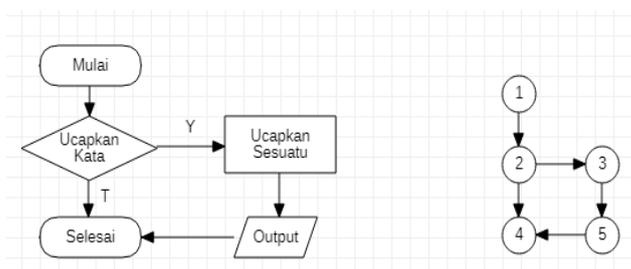
Gambar 5. *Flowchart* dan *Flowgraph* Menu Awal

Tabel 1. *Cyclomatic complexity*, *Region*, dan *Path* pada *Flowgraph* Menu Awal.

| <i>Cyclomatic Complexity</i> nilai <i>Edge</i> dan <i>node</i> | <i>Region</i> dengan <i>predicate Node(P)</i> , <b>P = 5</b> | <i>Path</i> pada <i>flowgraph</i> |
|--|--|-----------------------------------|
| E = 16   | $V(G) = P + 1$<br>= 5 + 1                                    | Path 1 = 1-2-3-4-9                |
| N = 12   | = 6  | Path 2 = 1-2-3-5-6-10-9           |
| $V(G) = E - N + 2$<br>= 16 - 12 + 2<br>= 6                     | <i>R flowgraph</i><br>= 6                                    | Path 3 = 1-2-3-5-6-7-11-9         |
| Jumlah <i>path</i> = 6   |  | Path 4 = 1-2-3-5-6-7-8-12-9       |
|  |  | Path 5 = 1-2-3-5-6-7-8-9          |
|  |  | Path 6 = 1-2-3-5-9                |

2. *Flowchart* dan *flowgraph* Menu Suara

Pada gambar 5 merupakan *flowchart* dan *flowgraph* menu suara.



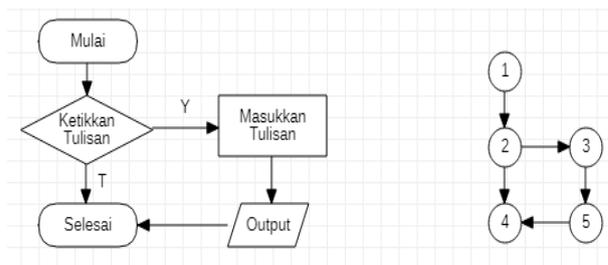
Gambar 6. *Flowchart* dan *Flowgraph* Menu Suara

**Tabel 2. Cyclomatic complexity, Region, dan Path pada Flowgraph Menu Suara**

| Cyclomatic Complexity nilai Edge dan node | Region dengan predicate Node(P), P = 5 | Path pada flowgraph |
|---|--|---------------------|
| E = 5                                     | V(G) = P + 1                           | Path 1 = 1-2-3-5-4  |
| N = 5                                     | = 1 + 1                                | Path 2 = 1-2-4      |
| V(G) = E-N+2                              | = 2                                    |                     |
| = 5-5+2                                   | R flowgraph                            |                     |
| = 2                                       | = 2                                    |                     |
| Jumlah path = 2                           |  |                     |

3. Flowchart dan flowgraph Menu Ketik

Pada gambar 6 merupakan flowchart dan flowgraph menu ketik.



**Gambar 7. Flowchart dan Flowgraph Menu Ketik**

**Tabel 3. Cyclomatic complexity, Region, dan Path pada Flowgraph Menu Ketik.**

| Cyclomatic Complexity nilai Edge dan node | Region dengan predicate Node(P), P = 5 | Path pada flowgraph |
|---|--|---------------------|
| E = 5                                     | V(G) = P + 1                           | Path 1 = 1-2-3-5-4  |
| N = 5                                     | = 1 + 1                                | Path 2 = 1-2-4      |
| V(G) = E-N+2                              | = 2                                    |                     |
| = 5-5+2                                   | R flowgraph                            |                     |
| = 2                                       | = 2                                    |                     |
| Jumlah path = 2                           |  |                     |

**Tabel 4. Data Hasil Pengujian whitebox**

| Flowgraph      | Cyclomatic Complex city (cc) | Region (R) | Independen Path |
|----------------|------------------------------|------------|-----------------|
| Menu Awal      | 6                            | 6          | 6               |
| Menu Home      | 2                            | 2          | 2               |
| Menu Akun Saya | 2                            | 2          | 2               |
| Jumlah         | 10                           | 10         | 10              |

Tabel 4 merupakan hasil pengujian dengan metode *whitebox* secara menyeluruh. Dengan hasil 3 parameter yang sama yaitu *cyclomatic complex city*, *region*, dan *independent path* yaitu 10 maka logika dan alur program sudah sesuai yang diharapkan. Pengujian *blackbox* dilakukan untuk menampilkan fungsi dari aplikasi yang telah dibuat mengenai cara dan kegunaannya, dengan melihat apakah keluaran data sudah sesuai yang diharapkan. Hasil pengujian *blackbox* dapat dilihat pada Tabel 5.

**Tabel 5. Pengujian Blackbox**

| Pengujian      | Bentuk pengujian            | Hasil yang diharapkan   | Hasil pengujian |
|----------------|-----------------------------|---|-----------------|
| Menu Home      | Masuk ke halaman utama      | Terdapat menu ketik, menu suara, informasi, dan help                                | Berhasil        |
| Menu Ketik     | Menekan Navigasi Menu Ketik | Tampilan menu ketik dan membaca teks yang kemudian di konversi kedalam bentuk suara | Berhasil        |
| Menu Suara     | Menekan Navigasi Menu Suara | Tampilan menu suara dan membaca suara yang kemudian di konversi kedalam bentuk teks | Berhasil        |
| Menu Informasi | Menekan Navigasi Informasi  | Tampilan informasi pembuat aplikasi   | Berhasil        |
| Menu Help      | Menekan Navigasi Help       | Tampilan Help   | Berhasil        |

Normalisasi teks dilakukan pada setiap kata yang terdapat angka, suhu, waktu, dan satuan mata uang. Tabel 6 menunjukkan hasil normalisasi teks pada menu ketik.

**Tabel 6. Hasil Normalisasi Teks Pada Menu Ketik**

| No | Teks     | Hasil Normalisasi                    |
|----|----------|--------------------------------------|
| 1. | °C       | Derajat Celcius                      |
| 2. | kW       | Killo Watt                           |
| 3. | 11       | Sebelas                              |
| 4. | 11,78    | Sebelas koma tujuh puluh delapan     |
| 5. | 12.000   | Dua belas ribu                       |
| 6. | Rp 1.000 | Seribu rupiah                        |
| 7. | 200 USD  | Dua ratus dollar Amerika Serikat     |
| 8. | 08:30    | Pukul delapan lewat lima puluh menit |

Pada tabel 7 dan 8 akan dilakukan pengujian aplikasi pada menu ketik dan menu suara, sedangkan tabel 9 merupakan hasil dari pengujian pemenggalan kata dua tingkat

**Tabel 7. Pengujian Aplikasi Pada Menu Ketik**

| No  | Kata     | Hasil Pengujian |
|-----|----------|-----------------|
| 1.  | Anak     | 100%            |
| 2.  | Akar     | 100%            |
| 3.  | Amal     | 100%            |
| 4.  | Asap     | 100%            |
| 5.  | Untuk    | 100%            |
| 6.  | Anda     | 100%            |
| 7.  | Antar    | 100%            |
| 8.  | Ampas    | 100%            |
| 9.  | Punah    | 100%            |
| 10. | Piring   | 100%            |
| 11. | Pusing   | 100%            |
| 12. | Ons      | 100%            |
| 13. | Ekspor   | 100%            |
| 14. | Sumbu    | 100%            |
| 15. | Bantu    | 100%            |
| 16. | Pantai   | 100%            |
| 17. | Gratis   | 100%            |
| 18. | Drama    | 100%            |
| 19. | Kompleks | 100%            |
| 20. | Traktor  | 100%            |
| 21. | Praktik  | 100%            |
| 22. | Praktis  | 100%            |
| 23. | Bangku   | 100%            |
| 24. | Modern   | 100%            |

|     |            |      |
|-----|------------|------|
| 25. | Korps      | 100% |
| 26. | Strategi   | 100% |
| 27. | Strata     | 100% |
| 28. | Struktur   | 100% |
| 29. | Stres      | 100% |
| 30. | Struktural | 100% |

|                             |             |
|-----------------------------|-------------|
| <b>Akurasi (%)</b>          | <b>100%</b> |
| <b>Rata-rata Akurasi(%)</b> | <b>100%</b> |

Berikut ini merupakan rumus untuk menentukan rata-rata akurasi dari pengujian yang telah dilakukan pada menu ketik.

$$\text{Rata-rata akurasi} = \frac{(100)}{1} = 100\%$$

**Tabel 8. Pengujian Aplikasi Pada Menu Suara**

| No  | Kata       | Hasil Pengujian |
|-----|------------|-----------------|
| 1.  | Anak       | 100%            |
| 2.  | Akar       | 100%            |
| 3.  | Amal       | 100%            |
| 4.  | Asap       | 100%            |
| 5.  | Untuk      | 100%            |
| 6.  | Anda       | 100%            |
| 7.  | Antar      | 100%            |
| 8.  | Ampas      | 100%            |
| 9.  | Punah      | 100%            |
| 10. | Piring     | 100%            |
| 11. | Pusing     | 100%            |
| 12. | Ons        | 100%            |
| 13. | Ekspor     | 100%            |
| 14. | Sumbu      | 100%            |
| 15. | Bantu      | 100%            |
| 16. | Pantai     | 100%            |
| 17. | Gratis     | 100%            |
| 18. | Drama      | 100%            |
| 19. | Kompleks   | 100%            |
| 20. | Traktor    | 100%            |
| 21. | Praktik    | 100%            |
| 22. | Praktis    | 100%            |
| 23. | Bangku     | 100%            |
| 24. | Modern     | 100%            |
| 25. | Korps      | 100%            |
| 26. | Strategi   | 100%            |
| 27. | Strata     | 100%            |
| 28. | Struktur   | 100%            |
| 29. | Stres      | 100%            |
| 30. | Struktural | 100%            |

|                             |             |
|-----------------------------|-------------|
| <b>Akurasi (%)</b>          | <b>100%</b> |
| <b>Rata-rata Akurasi(%)</b> | <b>100%</b> |

Berikut ini merupakan rumus untuk menentukan rata-rata akurasi dari pengujian yang telah dilakukan pada menu suara.

$$\text{Rata-rata akurasi} = \frac{(100)}{1} = 100\%$$

**Tabel 9. Hasil Pengujian Pemenggalan Dua Tingkat**

| Pola Suku Kata              | Teks       | Hasil Pemenggalan | Hasil Pengujian |
|-----------------------------|------------|-------------------|-----------------|
| V                           | Anak       | A-nak-            | 100%            |
|                             | Akar       | A-kar-            | 100%            |
|                             | Amal       | A-mal-            | 100%            |
|                             | Asap       | A-sap-            | 100%            |
| VK                          | Untuk      | Un-tuk-           | 100%            |
|                             | Anda       | An-da-            | 100%            |
|                             | Antar      | An-tar-           | 100%            |
|                             | Ampas      | Am-pas-           | 100%            |
| KV                          | Punah      | Pu-nah-           | 100%            |
|                             | Piring     | Pi-ring-          | 100%            |
|                             | Pusing     | Pu-sing-          | 100%            |
| VKK                         | Ons        | Ons-              | 100%            |
|                             | Ekspor     | Eks-por-          | 100%            |
| KVK                         | Sumbu      | Sum-bu-           | 100%            |
|                             | Bantu      | Ban-tu-           | 100%            |
|                             | Pantai     | Pan-tai-          | 100%            |
| KKV                         | Gratis     | Gra-tis-          | 100%            |
|                             | Drama      | Dra-ma-           | 100%            |
| KKVKK                       | Kompleks   | Kom-pleks-        | 100%            |
| KKVK                        | Traktor    | Trak-tor-         | 100%            |
|                             | Praktis    | Prak-tik-         | 100%            |
|                             | Praktik    | Prak-tis-         | 100%            |
| KVKK                        | Bangku     | Bang-ku-          | 100%            |
|                             | Modern     | Mo-dern-          | 100%            |
| KVKKK                       | Korps      | Korps-            | 100%            |
| KKKV                        | Strategi   | Stra-te-gi-       | 100%            |
|                             | Strata     | Stra-ta-          | 100%            |
| KKKVK                       | Struktur   | Struk-tur-        | 100%            |
|                             | Stres      | Stres-            | 100%            |
|                             | Struktural | Struk-tu-ral-     | 100%            |
| <b>Akurasi (%)</b>          |            |                   | <b>100%</b>     |
| <b>Rata-rata Akurasi(%)</b> |            |                   | <b>100%</b>     |

Pengujian pemenggalan dua tingkat ini hanya dilakukan untuk menu ketik. Dari 12 pola suku kata yang telah diujikan pada tabel 9 dapat dilihat bahwa pola suku kata yang semua contoh katanya dapat dipenggal

dengan benar, yaitu V, VK, KV, KVK, KKV, KVKK, KKVK, KVKKK, VKK, KKKV, KKVKK, dan KKKVK.

Berikut ini merupakan rumus untuk menentukan rata-rata akurasi dari pengujian yang telah dilakukan pada pemenggalan dua tingkat

$$\text{Rata-rata akurasi} = \frac{(100)}{1} = 100\%.$$

$$\text{Rata-rata akurasi keseluruhan} = \frac{(100)+(100)+(100)}{3} \times 100\% = 100\%.$$

Jadi, rata-rata akurasi pada aplikasi Spoxtech dengan percobaan menu ketik maupun menu suara dan pemenggalan dua tingkat pada menu ketik yaitu 100% untuk uji coba pada 300 kata dalam bahasa Indonesia.

#### 4. SIMPULAN

Aplikasi Spoxtech merupakan merupakan aplikasi yang dirancang khusus untuk penyandang tuna rungu dan tuna wicara. Sesuai dengan namanya, Spoxtech diambil dari kata *speech to text – text to speech*. Pada aplikasi ini terdapat menu suara yang berfungsi untuk mengonversi suara ke dalam bentuk teks dan menu ketik yang berfungsi untuk mengonversi teks ke dalam bentuk suara. Aplikasi ini dibuat dengan menggunakan algoritma Hidden Markov Model dalam generate suara ke bentuk teks dan menggunakan metode *finite state automata* dalam generate teks menjadi suara dengan melakukan pemenggalan dua tingkat. Hasil pengujian *whitebox* mendapatkan hasil yang sama dengan jumlah 10 pada parameter *cyclomatic complex city*, *region*, dan *independent path*. Pengujian aplikasi dengan menggunakan metode *blackbox* telah dinyatakan berhasil. Dengan menguji menu *home*, menu ketik, menu suara, menu informasi dan menu *help*. Dalam penelitian ini juga dilakukan pengujian normalisasi teks pada menu ketik yaitu setiap kata yang mengandung satuan

mata uang, waktu, suhu, dan angka. Nilai akurasi yang di dapat dalam pengujian aplikasi ini menggunakan bahasa Indonesia dengan pengujian pada menu suara dan menu ketik dan pengujian pemenggalan dua tingkat didapatkan nilai sebesar 100% dari 300 kata dalam Bahasa Indonesia.

#### DAFTAR PUSTAKA

- [1] Lutfi Maulana, Tining Haryanti, "Komunikasi Pintar Untuk Difabel Menggunakan *Smartphone* Android," *Jurnal Ilmiah NERO.*, vol.4 no.3, 2019.
- [2] Rieke A, Herman Tolle, Onny Setyawati "Pengembangan Aplikasi *Text-to-Speech* Bahasa Indonesia Menggunakan *Finite State Automata* Berbasis Android," *JNTETI*, Vol.5, No.1, Februari 2016.
- [3] S. P. Arbie, A. S. M. Lumenta, A. M. Rumagit dan A. P. R. Wowor, "Rancang Bangun *Aplikasi Text to Speech* Bahasa Indonesia," *e-journal Tek.Elektro dan Komput.*, 2013.
- [4] Isroka, Mohamad Yasin, "Aplikasi Konversi Teks Menjadi Suara Dengan Menggunakan Metode Penggal Kata *Finite State Automata* (FSA)," *Cyber-Techn*, Vol.7, No.2, 2013.
- [5] Wendy Liga, Erick Fernando, and Hendri, "Perancangan Aplikasi Komunikasi Penyandang Tunarungu Berbasis *PROCESSOR.*,vol.12, no.1, 1 April 2017.
- [6] Supriyanta, Pudji Widodo, Bekti Maryuni Susanto, "Aplikasi Konversi Suara Ke Teks Berbasis Android Menggunakan Google Speech API," *Bianglala Informatika*, vol. 2, no. 2, September 2014.
- [7] Uswatun Hasanah, Jaroji, "Aplikasi Dongeng Berbasis *Text to Speech* Untuk Platform Android," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 7, no. 2, November 2016:82-88.
- [8] Supriyanta, Pudji Widodo, "Aplikasi Konversi Suara Ke Teks Berbasis Android Menggunakan Google Speech API," *Indonesian Journal on Networking and Security*, Vol. 5, No. 2, Mei 2016.
- [9] A. Akbar, Ariyo Y.H, dan Ariyan Z. "Implementasi Google Speech API Pada Aplikasi Koreksi Hafalan AL-QUR'AN Berbasis Android," *JTIKA*, vol. 1, no. 1, Maret 2019, ISSN:2657-0327.
- [10] A. Widayanto, Suleman, and I. Anita S, "Rancang Bangun Aplikasi Kategori Bahasa Untuk Tuna Wicara Berbasis Android 1)," *J. Evolusi*, 2017.
- [11] E. Sari, "Perancangan Aplikasi Pengucapan Jam Berbahasa Inggris Dengan Metode *Finite State Automata*", Medan, *Jurnal Pelita Informatika Budi Darma*, Volume V, Nomer 2, 2013.