

## APLIKASI PERINTAH SUARA UNTUK OPERASI POWERPOINT

Sandy Kurniawan Ruhiat<sup>1</sup>, Ucuk Darusalam<sup>2</sup> dan Benrahman<sup>3</sup>

<sup>1,2,3</sup> Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional  
Jalan Sawo Manila, Pasar Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12520

<sup>1</sup>Email: [themonears@gmail.com](mailto:themonears@gmail.com)

<sup>2</sup>Email: [ucuk.darusalam@gmail.com](mailto:ucuk.darusalam@gmail.com)

<sup>3</sup>Email: [benrahman@civitas.unas.ac.id](mailto:benrahman@civitas.unas.ac.id)

### ABSTRAK

Microsoft PowerPoint merupakan aplikasi yang sering dipakai dalam presentasi dikarenakan kemudahan pengoperasiannya. Meski begitu, pengguna masih diharuskan untuk mengatur jalannya presentasi secara manual seperti memulai *slide*, meneruskan ke *slide* selanjutnya, kembali ke *slide* sebelumnya, dan sebagainya dengan menekan tombol di *keyboard*. Sering kali, sulit bagi pengguna meraih komputer saat sedang melakukan presentasi di depan orang banyak dan tentu akan memakan waktu. Tujuan penelitian ini adalah untuk mengurangi kontak fisik antara pengguna dengan komputer dengan menggunakan suara sebagai pengganti perintah keyboard untuk mengatur jalannya presentasi sehingga lebih efektif. Penelitian ini mengimplementasikan *Hidden Markov Models* dasar Microsoft Speech API dengan bahasa pemrograman C#. Tingkat toleransi perubahan pola kata yang mendekati kata asli adalah 84,29% serta akurasi jalannya program dalam latar suara jernih adalah 100%. Pengguna bisa menggunakan aplikasi dengan mikrofon atau semacamnya sebagai *input* yang sesuai dengan komputer mereka.

Kata kunci: perintah suara, *speech recognition*, HMM, power point, C#

### ABSTRACT

Microsoft PowerPoint is an application that is often used in presentations because of the ease of operation. Even so, users are still required to adjust the presentation manually such as starting a slide, continuing to the next slide, returning to the previous slide, and so on by pressing the keyboard button. Often, it is difficult for users to reach the computer while doing a presentation in front of many people and it will certainly take time. The purpose of this research is to reduce physical contact between users and computers by using voice instead of keyboard commands to operate the presentation so that it is more effective. This research implements the Hidden Markov Models base of Microsoft Speech API in C# programming language. The tolerance level of word pattern changes approaching the original word is 84.29% and the accuracy of the program running in the clear background is 100%. Users can use the application with a microphone or the like as input that is compatible with their computer.

Keywords: voice command, speech recognition, HMM, power point, C#

## 1. PENDAHULUAN

Penggunaan Microsoft PowerPoint sudah tak asing lagi dalam dunia pendidikan hingga bisnis. PowerPoint digunakan untuk mempresentasikan ide serta pendapat pengguna dalam bentuk digital, sehingga lebih mudah dicerna dan dinikmati karena dapat disisipkan berbagai macam gambar, suara, hingga video. Meski dibidang mudah untuk digunakan, tetap saja masih tak praktis untuk mengatur jalannya presentasi dengan perintah *keyboard* setiap ingin memulai *slide*, meneruskan *slide*, mengembalikan *slide*, dan menutup *slide*. Oleh karena itu, perintah suara akan lebih efektif dalam menggantikan perintah *keyboard*.

Dalam standar pengoperasian PowerPoint terdapat beberapa pintasan yang dapat diakses melalui *keyboard*. Memang terdapat perangkat *remote control* yang dapat mengakses pintasan perintah aplikasi sebagai pengganti standar *keyboard*, namun terkadang dapat membuat pengguna kesal karena harus menghafal banyak pintasan serta kesal karena dapat kehabisan baterai di tengah presentasi. Namun jika dengan menggunakan perintah suara, pengguna dapat menggunakan *built-in* mikrofon dari komputer atau laptop mereka. Pengguna juga bisa menyediakan perangkat mikrofon eksternal untuk memastikan suara yang tertangkap terdengar jelas.

Metode untuk mengubah suara menjadi teks disebut dengan *speech-to-text* (STT), atau biasa dikenal dengan *speech recognition* di mana suara dikonversi menjadi input dalam sistem komputer. Terdapat

beberapa model dalam *speech recognition*, misalnya seperti *Hidden Markov Models* (HMM), *Dynamic Time Warping* (DTW), dan *Neural Networks*. HMM merupakan yang paling banyak digunakan.

Pada penelitian sebelumnya dengan judul “Powerpoint Controller using Speech Recognition” oleh Christina, C., Rosalina, R., Wahyu, R. B., dan Roestam, R untuk aplikasi perintah suara PowerPoint, bahasa pemrograman yang dipakai adalah Java dengan algoritma HMM dan *CMUSphinx Library* sebagai dasar pembentuk arsitektur aplikasi [1]. Namun dalam penelitian tersebut, pengguna masih diharuskan menyesuaikan suara mereka sebelum menggunakan aplikasi. Dan juga, tak terdapat uji coba sejauh mana toleransi pola serta fonem kata dalam menginput perintah suara.

Aplikasi ini akan menyesuaikan input suara dari pengguna dengan daftar kata yang telah didefinisikan dalam *Grammar class* dari *.NET Framework* dengan referensi *Speech Recognition Grammar Specification Grammar* (SRGS Grammar) lalu mengonversinya ke dalam perintah *keyboard* yang telah ditentukan juga. Tak ada indikasi khusus terhadap jenis suara yang bisa digunakan, sehingga tak ada penyesuaian suara oleh pengguna. Selama pengucapan kata dalam bahasa Inggris benar, maka perintah suara akan berjalan dengan semestinya.

## 2. MATERI DAN METODE

### *Speech Recognition*

*Speech recognition* adalah proses mengubah kata yang diucapkan menjadi teks. *Speech recognition* melibatkan menangkap ucapan dari pengguna, mendigitalkan ucapan tersebut menjadi sinyal digital kemudian mengubahnya menjadi unit dasar ucapan dan menganalisis secara kontekstual kata-kata untuk memastikan ejaan yang benar terhadap kata-kata yang terdengar sama [2]. Ketika pengguna berbicara ke mikrofon, sistem menangkap suara pengguna dalam bentuk sinyal akustik analog dan menghasilkan impuls listrik. Kemudian *sound card* akan mengubah sinyal akustik menjadi sinyal digital dan mesin *speech recognition* akan memproses sinyal digital menjadi fonem, di mana fonem akan menciptakan kata-kata [1].

Dalam pembentukan algoritma *speech recognition* terdapat tiga model yang dapat digunakan, seperti DTW untuk menghitung kesamaan antara dua deret waktu yang mungkin berbeda dalam waktu dan kecepatan, dan juga terdapat Neural Network yang mempunyai varian yaitu Long Short-Term Memory yang mampu mempelajari urutan ketergantungan dalam memprediksi urutan masalah [3]. Selanjutnya ada HMM yaitu model statistik yang menghasilkan keluaran berupa susunan simbol atau jumlah.

### *Speech Recognition Grammar Specification (SRGS)*

SRGS merupakan standar *grammar* dari *World Wide Web Consortium* (W3C) untuk spesifikasi *grammar* melalui seperangkat pola kata dan memberitahu *speech recognition* kata apa yang akan dikatakan manusia. *Syntax* dalam format *grammar* terbagi dalam dua form, yaitu modifikasi *Backus-Naur form* (BNF) dan *Extensible Markup Language* (XML) form [4]. SRGS Grammar digunakan sebagai dasar referensi pembentuk kata dalam *System.Speech namespace*.

### *Hidden Markov Models (HMM)*

HMM digunakan pada berbagai banyak sistem disebabkan karena kalimat dapat dilihat sebagai *piecewise stationary signal*, sehingga setiap perkataan dapat dilihat sebagai pendekatan sebuah proses yang tidak bergerak/tetap yang terbentuk dalam susunan simbol [5].

Saat memasukkan kalimat yang kompleks bisa terjadi kesalahan dalam pengenalan huruf yang membentuk kata dalam kalimat tersebut. HMM berfungsi agar kesalahan tersebut bisa diatasi dan perbedaan dalam pengejaan kata dapat dikenali.

Elemen dasar dalam algoritma HMM dengan notasi pembentuk  $\lambda = (A, B, \pi)$  sebagai berikut [6]:

1. *Hidden state*:  $Q = \{q_1, q_2, q_3, \dots, q_n\}$ ;
2. *Observed state*:  $V = \{v_1, v_2, \dots, v_m\}$ ;
3. Probabilitas transisi *state*:  
 $A = \{a_{ij}\}, a_{ij} = P_t(q_{t+1} = S_j | q_t = S_i), 1 \leq (i,j) \leq N$ ;
4. Probabilitas emisi simbol:  
 $B = \{b_j(k)\}, b_j(k) = P_t(v_k = o_t | q_t = S_j), 1 \leq k \leq M, 1 \leq j \leq N$ ;
5. Distribusi peluang *initial state*:  
 $\pi = \{\pi_i\}, \pi_i = P[q_1 = S_i], 1 \leq j \leq N$ ;

Contoh sederhana misalkan terdapat kata “first”, maka akan dibentuk susunan sesuai fonetiknya: / f / ó / r / s / t / dengan transisi huruf yang mendekati kata dan bunyi asli.

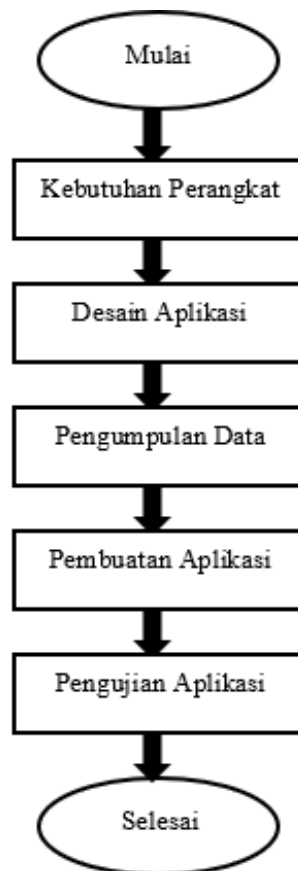
Pada dasarnya, algoritma HMM membentuk susunan huruf yang diambil dari *string* kumpulan kata yang sebelumnya telah didefinisikan (biasanya melalui standar *grammar*) lalu menyesuaikan transisi pembentukan kata sesuai suara yang ditangkap oleh *input* mikrofon.

### Microsoft Speech API

Speech Application Programming Interface (SAPI) merupakan Application Programming Interface (API) yang dikembangkan oleh Microsoft untuk digunakan *pada speech recognition* dan *speech synthesis* dengan sistem operasi berbasis Windows. Microsoft telah meneliti dan mengembangkan teknologi *speech* selama lebih dari satu decade [7]. Pada tahun 1993, perusahaan tersebut merekrut Xuedong (XD) Huang, Fil Allewa, dan Mei-Yuh Hwang — tiga dari empat orang yang bertanggung jawab atas sistem Sphinx-II *speech recognition* berbasis HMM di Universitas Carnegie Mellon, yang mencapai ketenaran di bidang *speech* pada tahun 1992 karena akurasi yang belum pernah ada sebelumnya. Sejak awal, dengan pembentukan tim Speech API (SAPI) 1.0 pada tahun 1994, Microsoft terdorong untuk menciptakan teknologi *speech* yang akurat dan dapat diakses oleh pengembang melalui API yang kuat[8][9].

### Kerangka Kerja Penelitian

Gambar 1 adalah kerangka kerja yang menunjukkan langkah-langkah penelitian yang telah dilakukan.



Gambar 1. Kerangka kerja penelitian

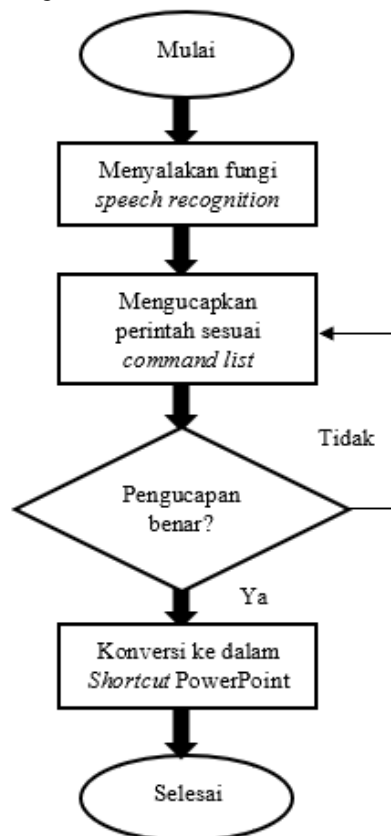
Detail langkah-langkah tersebut adalah sebagai berikut:

1. Kebutuhan Perangkat  
Untuk memulai penelitian ini, diperlukan perangkat yang memadai agar penelitian tersebut dapat berjalan dengan baik dan benar. Spesifikasi perangkat juga perlu disesuaikan dengan kebutuhan penelitian. Dalam penelitian ini, diperlukan satu buah laptop atau komputer dengan OS Windows yang terinstalasi Visual Studio dan *.NET Framework* minimal 3.0 serta mikrofon baik built-in maupun eksternal sebagai alat input suara [10].
2. Desain Aplikasi  
Setelah memastikan perangkat tersedia, dilanjutkan dengan membuat sketsa struktur aplikasi, dari mulai arsitektur program hingga tampilan antarmuka
3. Pengumpulan Data  
Proses pencarian serta pengumpulan keseluruhan data yang diperlukan dalam proyek. Dalam pembuatan aplikasi ini terbatas dalam struktur penulisan program, semacam *namespaces* serta *references* apa saja yang dibutuhkan, atau bahkan pencarian *external library* atau *third party software* jika diperlukan untuk melengkapi koding program.

4. Pembuatan Aplikasi  
Dalam tahap ini setelah semua data lengkap sesuai dengan desain awal, dilanjutkan dengan membuat aplikasi melalui bahasa pemrograman C# dalam aplikasi Visual Studio.
5. Pengujian Aplikasi  
Diperlukan pengujian di saat aplikasi telah selesai dari tahap pembuatan dengan menjalankan aplikasi tersebut dan memastikan seluruh fungsi berjalan sesuai dengan desain awal, apakah ada kesalahan atau tidak baik dalam coding maupun tampilan aplikasi. Terdapat dua tahap pengujian, yaitu *alpha* di mana diuji oleh pembuat, dan selanjutnya *beta* yang diuji oleh pengguna akhir.

#### Flowchart Sistem Aplikasi

Jalan kerja sistem aplikasi ini sederhana, pengguna hanya perlu mengaktifkan fungsi *speech recognition* lalu mengucapkan perintah ke mikofon dan sistem akan dengan sendirinya mengonversi suara pengguna ke dalam pintasan yang biasa digunakan dalam operasi PowerPoint jika pengucapan perintahnya jelas. Selengkapnya bisa dilihat dalam gambar 2.



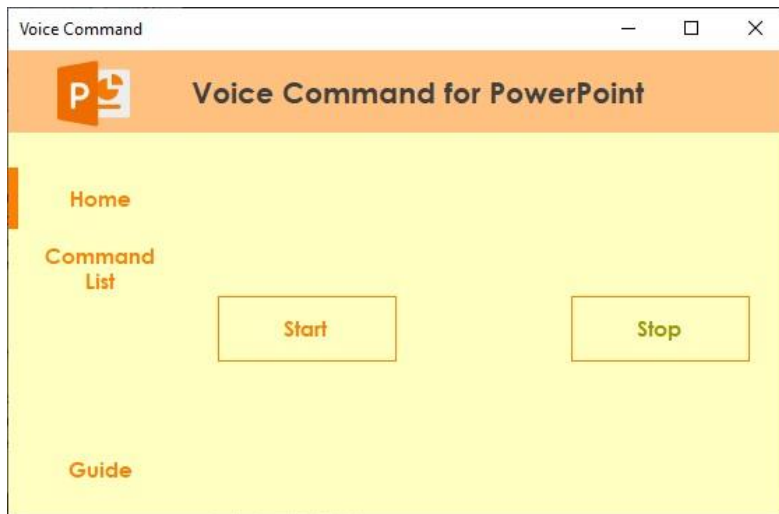
Gambar 2. Flowchart aplikasi

### 3. HASIL DAN PEMBAHASAN

#### Pembahasan Aplikasi

Setelah mengimplementasi kerangka dalam metode penelitian, hasil aplikasi yang didapat adalah sebagai berikut. Gambar 3 merupakan tampilan Home dari aplikasi yang terdapat tombol Start dan Stop. Saat pengguna menekan tombol Start, fungsi *speech recognition* dapat digunakan dalam mengoperasikan presentasi PowerPoint. Untuk memberhentikan fungsi *speech recognition* pengguna cukup menekan tombol Stop.

Dengan memanfaatkan fungsi *Choices* seperti pada gambar 4 pengguna dapat menaruh beberapa kata atau kalimat dalam satu daftar dan memasukkannya ke dalam *Grammar* sehingga tak perlu mengulang *GrammarBuilder* berkali-kali. Dengan *Grammar* tersebut pengguna dapat menggunakan kalimat yang sudah didaftarkan untuk dikonversi ke dalam pintasan *keyboard* dalam aplikasi PowerPoint dengan menghubungkan *event* tersebut pada *switch case* lalu melanjutkan ke fungsi *SendKeys.Send()*. Misalkan saat pengguna mengucapkan “start first” maka program akan memberikan input “F5” sebagai ganti *keyboard* lalu memulai *fullscreen* di PowerPoint.



Gambar 3. Tampilan menu Home

```
Choices list = new Choices();  
list.Add(new string[] {"start first",  
    "start here",  
    "end slide",  
    "next slide",  
    "previous slide",  
    "first slide",  
    "last slide" });  
Grammar gr = new Grammar(new GrammarBuilder(list));
```

Gambar 4. Grammar building

Dalam beberapa laptop, tombol fungsi Fn secara *default* aktif yang membuat pintasan F1 – F12 tak bisa dimasukkan langsung, melainkan harus berbarengan dengan tombol Fn sehingga tak bisa digunakan sebagai pengganti pintasan. Masalah tersebut bisa diatasi dengan cara menonaktifkan fungsi Fn di menu BIOS.

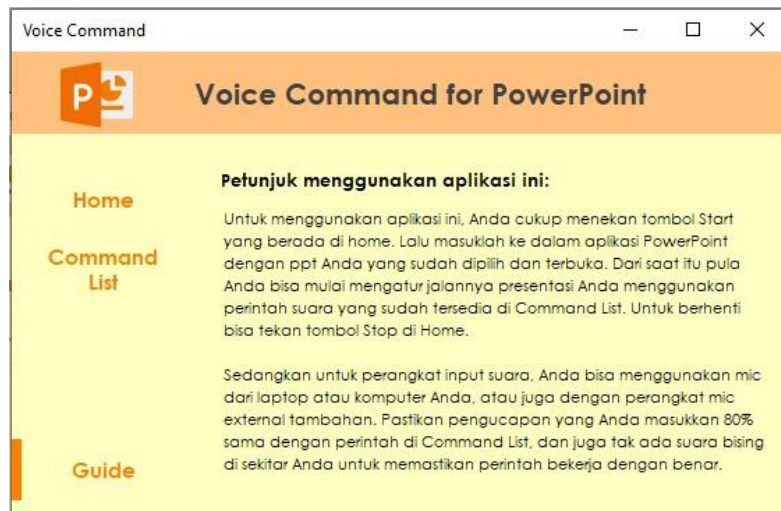


Gambar 5. Tampilan menu Command List

Menu pada gambar 5 merupakan daftar perintah yang bisa pengguna ucapkan di mikrofon. Dengan menekan tombol Hear di masing-masing perintah, pengguna juga bisa mendengarkan bagaimana pengucapan perintah yang benar sehingga lancar dalam mengoperasikan presentasi PowerPoint.

Fungsi *speechSynthesizer.Speak()* yang membaca daftar perintah dari *Grammar* dan mengeluarkan suaranya melalui perangkat *output* yang ada, contohnya *speaker* atau *headset* [11]. Misalkan saat pengguna memilih tombol Hear yang berada di samping perintah mulai presentasi dari *slide* pertama, maka akan terdengar *output* pengucapan “start first” yang benar. Gambar 6 merupakan petunjuk penggunaan aplikasi.





Gambar 6. Tampilan menu Guide

### Pengujian Aplikasi

Setelah melalui pembahasan aplikasi, tahap berikutnya merupakan hasil pengujian aplikasi yang terdiri dari tiga tabel.

Tabel 1. Evaluasi perintah suara

Skenario	Hasil yang diharapkan	Evaluasi
Perintah “Start First”	Memulai tampilan <i>fullscreen</i> dari <i>slide</i> awal atau pertama	Berhasil 100%
Perintah “Start Here”	Memulai tampilan <i>fullscreen</i> dari <i>slide</i> yang sedang dipilih	Berhasil 100%
Perintah “End Slide”	Mengakhiri tampilan <i>fullscreen</i> PowerPoint	Berhasil 100%
Perintah “First Slide”	Menampilkan <i>slide</i> pertama	Berhasil 100%
Perintah “Last Slide”	Menampilkan <i>slide</i> terakhir	Berhasil 100%
Perintah “Next Slide”	Menuju <i>slide</i> selanjutnya	Berhasil 100%
Perintah “Previous Slide”	Menuju <i>slide</i> sebelumnya	Berhasil 100%

Pada tabel 1 fungsi perintah suara yang telah didaftarkan dapat mengenali ucapan pengguna melalui mikrofon dan menjalankan program sesuai desain awal. Pengguna juga harus mengucapkan keseluruhan kalimat yang telah didaftarkan.

Tabel 2. Uji coba persamaan bunyi dalam pengucapan

Perintah (fonetik)	Pola pengucapan kata		
	Pola 1	Pola 2	Pola 3
Start (stárt)	Stort, Stert, Sturt	Stat, Tart, Stas	Ars. Art. Sat
First (fórst)	Frits, Firs, Furs	Fus, Fits	Ist
End (énd)	En, Un, And	N	Den
Here (hír)	Ir, Hir	Her, Hire	Ere
Next (nékst)	Nek, Ex	Nes	Es
Previous (prívíøs)	Pervious, Prepious	Vious, Pious	Ious
Last (læst)	Lats, Las	Lus, Le	Lat
Slide (slájd)	Side, Slid, Slat	Said, Laid	Sed
<b>Berat Tes</b>	70	70	70
<b>Hasil Tes</b>	58	43	17
<b>Hasil Efektif</b>	84,29%	61,43%	24,29%

Tabel 2 menunjukkan pola kata asli diatur ulang menjadi pola-pola kata yang mendekati bunyi kata asli. Terdapat 3 pola, pola 1 adalah pola yang paling mendekati kata asli dan hampir tak ada perbedaan dari pola kata asli, hingga pola 3 yang paling jauh dari pola kata asli namun setidaknya masih terdapat sedikit bunyi yang sama.

Berat tes pada tabel 2 adalah 10 dari setiap perintah maka berjumlah 70. Perintah diucapkan melalui kombinasi kata masing-masing pola tersendiri, sehingga diketahui seberapa jauh toleransi pengucapan kata yang digambarkan oleh hasil efektif.

Tabel 3. Uji coba lingkungan latar suara

Skenario Latar Suara	Berat Tes	Hasil Tes	Hasil Efektif
Jernih	70	70	100%
Bising (30%)	70	65	92,86%
Bising (50%)	70	59	84,28%
Musik (Volume 50%)	70	60	85,71%
Musik (Volume 70%)	70	56	80%
Musik (Volume 100%)	70	51	72,86%
Ukuran ruangan (4m <sup>2</sup> )	70	69	98,57%
Ukuran ruangan (6m <sup>2</sup> )	70	68	97,14%

Sama seperti pada tabel 2, total berat tes pada tabel 3 adalah 70. Kriteria seperti jernih, bising, musik, serta ukuran ruangan digunakan sebagai penentu berapa kali perintah dapat berjalan dalam lingkungan tersebut.

Keadaan latar suara lingkungan masing-masing dapat mempengaruhi bagaimana suara pengguna dapat diproses oleh program. Tak hanya keadaan lingkungan, kualitas mikrofon, sensitivitas penerima suara, jarak mulut dengan mikrofon juga dapat mengubah hasil efektif dalam tabel 3.

#### 4. KESIMPULAN DAN SARAN

PowerPoint merupakan media yang sangat memudahkan orang untuk dapat melakukan presentasi. Aplikasi ini dapat membantu pengguna lebih efisien dan praktis karena pengguna tak perlu lagi menggunakan *keyboard* untuk mengatur jalannya tayangan presentasi mereka. Pengguna hanya perlu mengucapkan perintah yang telah terdaftar, dan bisa lebih leluasa mempresentasikan materi mereka di depan orang banyak.

Beberapa inti yang dapat disimpulkan dari hasil penelitian dalam menggunakan aplikasi ini adalah sebagai berikut:

1. Pengucapan perintah secara penuh sesuai dengan kalimat yang telah didaftarkan dalam *grammar* diperlukan agar program berjalan dengan benar. Misalkan perintah yang didaftarkan “start first” maka pengucapannya harus keseluruhan kalimat tersebut.
2. Tingkat toleransi perubahan pola kata yang diucapkan adalah 84,29% untuk pola 1, 61,43% untuk Pola 2, dan 24,29% untuk pola 3 dari berat tes 70. Kriterianya merupakan pola 1 yang mendekati kata asli hingga pola 3 yang paling jauh dari kata asli namun masih terdapat bunyi yang sama.
3. Latar suara mempengaruhi jalannya program dengan akurasi tertinggi di latar jernih = 100% dan akurasi terendah berada di latar musik (volume 100%) = 72,86%.

#### DAFTAR PUSTAKA

- [1] C. Christina, Rosalina, R. B. Wahyu, and R. Roestam, “Powerpoint Controller using Speech Recognition,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 3, no. 2, Aug. 2017, doi: [10.28932/jutisi.v3i2.609](https://doi.org/10.28932/jutisi.v3i2.609).
- [2] M. Rusdi and A. Yani, “Sistem Kendali Peralatan Elektronik Melalui Media Bluetooth Menggunakan Voice Recognition,” *Journal of Electrical Technology*, vol. 3, no. 1, Feb. 2018.
- [3] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [4] “Speech Recognition Grammar Specification Version 1.0.” <https://www.w3.org/TR/speech-grammar/> (accessed Feb. 10, 2021).
- [5] S. Marigold and S. Magesh, “Voice Command System Using C# .Net Application with Artificial Intelligence,” *International Journal of Recent Technology and Engineering*, vol. 7, no. 6S4, Apr. 2019.
- [6] R. S. Chavan and D. G. S. Sable, “An Overview of Speech Recognition Using HMM,” *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 6, pp. 233–238, Jun. 2013.
- [7] Shofwanin, A. Rakhmadi, and F. Yasin Irsyadi, “Program Pemanggilan Aplikasi Perkantoran Dengan Microsoft Speech Application Programming Interface Yang Terintegrasi Melalui Visual Basic.Net,” Universitas Muhammadiyah Surakarta, Surakarta, 2014.
- [8] “Exploring Speech Recognition And Synthesis APIs In Windows Vista | Microsoft Docs.” <https://docs.microsoft.com/en-us/archive/msdn-magazine/2006/january/exploring-speech-recognition-and-synthesis-apis-in-windows-vista> (accessed Feb. 11, 2021).

- [9] “Learning to use the CMU SPHINX Automatic Speech Recognition system.” <http://www.speech.cs.cmu.edu/sphinx/tutorial.html> (accessed Feb. 07, 2021).
- [10] N. M. Kavya, S. P. D’Souza, Gagan, and G. R. Poojary, “Voxification – Voice based manipulation of slides in powerpoint,” *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 3, May 2018.
- [11] A. Rianda, “Aplikasi Perintah Suara untuk Mengendalikan Tayangan Presentasi pada Microsoft Power Point,” *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 22, no. 1, May 2017.