

Implementasi Sistem Kendali Lampu dengan *Voice Command* Menggunakan Google API

Implementation of Lamp Control System by Google API

Priscilya Inri Sasia^{1*}, Muhammad Ary Murti², Casi Setianingsih³
Telkom University^{1,2,3}

priscilyainri@gmail.com^{1*}, arymurti@telkomuniversity.ac.id², setiacasie@telkomuniversity.ac.id³

Abstrak - Lampu merupakan sebuah perangkat yang saat ini digunakan dimana-mana dan kini menjadi suatu kebutuhan manusia yang tidak dapat diabaikan. Smart lamp merupakan suatu sistem yang dirancang untuk mengendalikan lampu-lampu yang sudah terintegrasi dengan sistem yang ada pada suatu ruangan. Sistem ini menggunakan Raspberry Pi sebagai microcontroller dan algoritma RNN dan LSTM yang sudah tersedia pada Google API dalam mengolah suara perintah dari user, serta Relay sebagai aktuator yang akan menggerakkan lampu untuk on atau off. Dari penelitian yang dibangun, dilakukan pengujian pada tiga device. Diperoleh hasil pengujian bahwa rata-rata waktu yang diperlukan sistem untuk menyalakan dan mematikan Lampu I secara berturut-turut adalah 3,3 second dan 3,28 second dengan intensitas suara minimal 60,6 dB. Untuk menyalakan dan mematikan Lampu II secara berturut-turut adalah 3,43 second dan 3,61 second dengan intensitas suara minimal 60,8 dB. Sedangkan untuk menyalakan dan mematikan Lampu III secara berturut-turut adalah 3,32 second dan 3,39 second dengan intensitas suara minimal 61,32 dB. Ketiga lampu dapat dikendalikan dengan jarak terjauh adalah 1,2 meter.

Kata kunci : Lampu, voice command, smart lamp.

Abstract - Lamp is a device that currently used everywhere and now lamp be a need that cannot be ignored. Smart lamp is a system that designed to control lamp that have been integrated with existing systems in a room. This system uses the Raspberry Pi as a controller and the RNN and LSTM algorithms that are already available on the Google API in processing voice commands from the user, and the Relay as an actuator that will move the lights on or off. From this research, the results of testing show that the average time required for the system to turn on and turn off Lamp I are 3.3 seconds and 3.28 seconds respectively with a minimum sound intensity of 60.6 dB. To turn on and turn off Lamp II in succession is 3.43 second and 3.61 second with a minimum sound intensity of 60.8 dB. Meanwhile, to turn on and turn off Lamp III in a row is 3.32 seconds and 3.39 seconds with a sound intensity of at least 61.32 dB. The three lamps can be controlled with the furthest distance is 1.2 meters.

Keywords: Lamp, voice command, smart lamp.

I. Pendahuluan

Perkembangan teknologi yang semakin pesat membuat perangkat-perangkat listrik yang sebelumnya pengendaliannya masih dilakukan secara manual, kini dapat dikendalikan secara otomatis. Salah satunya adalah lampu. Lampu LED penerang ruangan biasanya dikendalikan oleh sebuah sakelar yang mengharuskan *user* untuk bergerak menuju ke tempat saklar itu berada jika ingin menyalakan/mematikan lampu. Untuk mengurangi pergerakan manusia dan menghemat waktu yang digunakan, lampu dapat dikendalikan melalui Android[1] dan perintah suara[2].

Pada penelitian sebelumnya, pengoperasian sistem yang menggunakan Android[1] membuat *user* harus selalu membawa dan membuka Android terlebih dahulu, sehingga kurang efisien dalam waktu. Sistem seperti ini juga tidak mendukung *multiple platform*[3]. Akibatnya, ketika Android rusak, hilang, atau tidak dapat digunakan, sistem yang dibangun juga tidak dapat digunakan. Dalam pengolahan suara, penelitian sebelumnya menggunakan *Mel-Frequency Cepstrum Coefficients* [4], *Hidden Markov Model* (HMM)[4], *Principal Component Analysis* (PCA)[5], *Analog to Digital Converter*[6], dan lain sebagainya. Salah satu *tool* yang dapat digunakan juga untuk mengolah suara adalah Google Cloud *speech API*. Google

Cloud *speech* API adalah salah satu layanan *machine learning* yang dikembangkan oleh Google dimana Google Cloud *speech* API juga termasuk dalam *Artificial Intellegent* [7].

Tujuan dari penelitian ini adalah untuk mengatasi keterbatasan-keterbatasan dari penelitian sebelumnya. Jika sebelumnya menggunakan Android dalam pengendaliannya, pada sistem yang dibangun kali ini menggunakan perintah suara sebagai pengendali sistem sehingga dapat mengatasi keterbatasan dalam *multiple platform* pada Android. Selain itu, *tools* yang digunakan pada sistem ini adalah Google API yang lebih mudah dalam implementasinya untuk pengolahan suara. Sistem ini menggunakan *voice command* sebagai *input*, *Raspberry Pi* sebagai *controller*, dan menggunakan algoritma *Reccurent Neural Network* (RNN) dan *Long-Short Term Memory* (LSTM) yang sudah tersedia pada *tool Google Cloud API* yang berperan dalam proses pengolahan *voice command*. *Google Cloud API* ini sudah terintegrasikan di *Raspberry Pi*. Selain itu, sistem ini juga menggunakan *Relay* sebagai aktuator yang akan menggerakkan lampu untuk *on/off*.

II. Metodologi Penelitian

2.1 Studi Literatur

Studi literatur merupakan tahap awal dalam mempelajari landasan-landasan teori yang berkaitan dengan penelitian, baik perangkat keras maupun perangkat lunak yang digunakan.

2.2 Sistem Kontrol

Sistem kontrol adalah interkoneksi komponen yang membentuk konfigurasi sistem yang akan memberikan respon sistem yang diinginkan. Sistem kontrol ini digunakan dalam menganalisis suatu sistem yang mengasumsikan adanya hubungan sebab-akibat untuk setiap komponen-komponen suatu sistem [8].

2.3 Relay

Relay adalah komponen elektronika berupa saklar atau *switch* yang menggunakan prinsip elektromagnetik dalam pengoperasiannya. Relay biasanya digunakan untuk menggerakkan saklar dari posisi *off* ke posisi *on* atau sebaliknya, sehingga komponen dapat menghantarkan listrik yang bertegangan tinggi, namun dengan arus listrik yang kecil. Daya yang dibutuhkan relatif kecil untuk mengaktifkan Relay tetapi Relay dapat mengendalikan sesuatu yang membutuhkan daya lebih besar[9].

Dalam prinsip kerjanya berdasarkan referensi[10], Relay memiliki 2 kontak *point* berdasarkan kondisi awal Relay sebelum diaktifkan yaitu.

- a. *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada pada posisi *close* (tertutup).
- b. *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada pada posisi *open* (terbuka).

2.4 Google Cloud *Speech to text* API

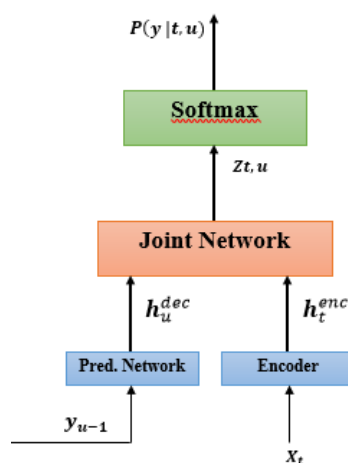
Google Cloud *Speech to Text* API merupakan suatu produk AI dan *Machine Larning* yang diproduksi oleh Google yang dapat mengkonversi audio ke teks [11]. Google Cloud *Speech-to-Text* memungkinkan peneliti untuk mengkonversi audio ke teks dengan menerapkan model *Neural Network* yang kuat dalam API yang mudah digunakan. Kelebihan dalam menggunakan *Google API* berdasarkan referensi [11] adalah sebagai berikut:

- mengenali 120 bahasa dan varian untuk mendukung basis pengguna global.
- dapat mengaktifkan perintah dan kontrol suara, menyalin audio dari pusat panggilan dan memproses *streaming real-time* atau audio yang direkam sebelumnya.
- secara otomatis mengidentifikasi kata yang diucapkan dan diubah ke teks dengan menggunakan format tertentu, seperti tanggal, jam, dan lain-lain.

Google API menggunakan algoritma *Neural Network*. Algoritma ini terdiri dari beberapa jenis seperti *Deep Neural Networks* (DNNs), *Recurrent Neural Networks* (RNNs), *Long Short-Term Memory Networks* (LSTMs), *Convolutional Networks* (CNNs), dan lain sebagainya. Google

API sendiri mengembangkan sebuah algoritma sendiri yang bernama *Recurrent Neural Network Transducers* (RNN-T). Algoritma ini dikembangkan dari jenis-jenis algoritma *Neural Network* yang sudah ada [12].

Dalam implementasinya, keluaran dari algoritma ini adalah karakter alfabet. Sistem pengenalan yang dimiliki oleh RNN-T akan menampilkan karakter satu-per-satu saat *user* berbicara. Sistem ini bekerja dengan menggunakan sistem *loop* umpan balik. Sampel *input* yang akan diprediksi ini kemudian dibandingkan satu sama lain sehingga hasil dari pengenalan suara/perintah akan semakin akurat. Sistem ini dapat dideskripsikan dalam Gambar 1 berdasarkan [12].



Gambar 1 Deskripsi sistem RNN-T.

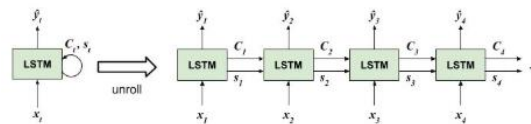
Gambar di atas merupakan sebuah representasi dari algoritma RNN-T dengan x sebagai sampel *input* audio, dan y sebagai simbol yang diprediksi (*output* dari *Softmax*). Sampel x akan diproses ke dalam sistem melalui *Joint Network*. Sampel x yang sudah diproses akan keluar sebagai *output* dari *Softmax* sebagai sampel y . Sampel y akan diteruskan kembali untuk masuk kembali ke dalam sistem. Kemudian sampel y akan diproses oleh *Prediction Network* yang akan membandingkan kesamaan karakter dari sampel y (yang sudah diproses) dengan sampel x yang dibaca pertama kali sebagai sampel untuk memastikan keakuratan sistem. *Prediction* dan *Encoder Networks* merupakan LSTM dan RNNs.

2.5 Recurrent Neural Network (RNN)

RNN merupakan jenis arsitektur *Neural Network* yang biasanya memiliki *input* berupa data sekuensial dimana pemrosesannya dipanggil secara berulang. RNN telah mengalami kemajuan yang pesat dan telah merevolusi bidang-bidang seperti pemrosesan bahasa alami (NLP), pengenalan suara, sintesa musik, pemrosesan data finansial seri waktu, analisa deret DNA, analisa video, dan sebagainya [13]. RNN juga dapat dikategorikan sebagai algoritma *deep learning* karena data diproses melalui banyak *layer*. Karena memiliki banyak *layer*, perhitungan komputasi pada RNN lebih banyak sehingga *output* yang dihasilkan cukup akurat. Terbukti pada percobaan penelitian yang berjudul “*Multiple Feature Extraction for RNN-based Assamese Speech Recognition for Speech to Text Conversion Application*” memaparkan bahwa tingkat keakuratan sistem dengan menggunakan RNN meningkat 10% [14].

2.6 Long-Short Term Memory (LSTM)

LSTM adalah jenis RNN yang didesain ulang, dan memiliki cara kerja yang hampir sama yaitu melakukan proses data secara berulang. Bedanya pada LSTM terdapat sinyal tambahan yang diberikan dalam pemrosesan dari satu waktu ke waktu berikutnya. Sinyal tersebut disebut *cell state* dan *memory cell* yang direpresentasikan dengan simbol C_t [15].

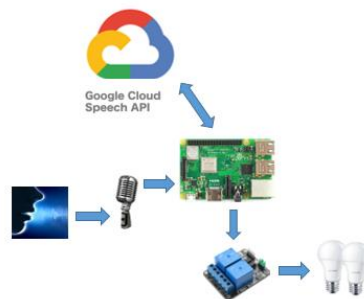
Gambar 2 Proses kerja LSTM^[15].

RNN sederhana memiliki kendala pada *vanishing/exploding gradient* [16] ketika sinyal yang *error* adalah *back-propagated through time* (BTT). Namun, kendala ini dapat diatasi dengan baik dengan menggunakan LSTM melalui penggunaan empat komponen berikut [16]:

- a. *memory units* : untuk menyimpan keadaan jaringan dalam setiap waktu.
- b. *input gates* : untuk memodulasi aktivasi sampel *input* ke dalam sel jaringan.
- c. *output gates* : untuk memodulasi aktivasi *output* ke dalam sel jaringan
- d. *forget gates* : untuk mengatur ulang secara otomatis memori sel.

LSTM menggunakan komponen-komponen di atas untuk mencapai jaringan yang dapat mempertahankan status data dan menyebarkan *gradient* secara stabil dalam rentang waktu yang lama. Sehingga, selain dapat melengkapi RNN, LSTM juga telah terbukti mengungguli *Deep Neural Network* (DNN) pada berbagai perintah dalam *Automatic Speech Recognizer* (ASR) [16].

2.7 Perancangan Sistem Perangkat Keras

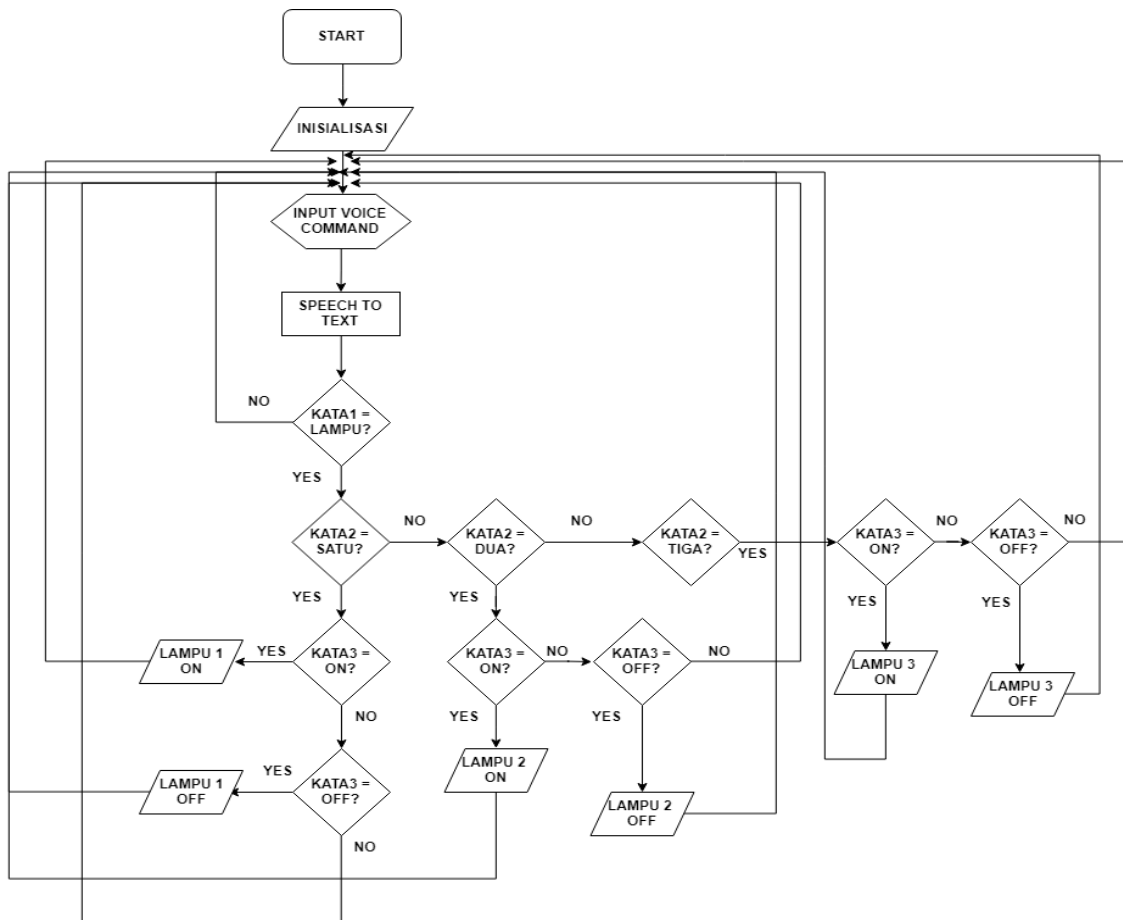


Gambar 3 Perancangan sistem perangkat keras.

Gambar 3 merupakan perancangan sistem yang dibangun pada penelitian ini. Pada sistem ini, *voice command* akan berperan sebagai *input*. *User* akan memberikan perintah untuk menghidupkan atau mematikan lampu. Sinyal suara tersebut akan ditangkap oleh *microphone* yang sudah terhubung dengan Raspberry Pi. Suara akan diproses dengan mengubah suara berupa sinyal audio menjadi suatu teks (*Speech to Text*). Proses ini menggunakan algoritma *Recurrent Neural Network* dan *Long-Short Term Memory* (LSTM) yang sudah tersedia pada *Google Cloud Speech API* yang sudah terintegrasi di *Raspberry Pi*. Setelah diubah menjadi sebuah teks, selanjutnya teks tersebut akan diproses untuk mengidentifikasi perintah apa yang diberikan oleh *user*. Perintah ini akan diteruskan ke *Relay* yang berperan sebagai *aktuator*. *Relay* akan menghidupkan lampu atau mematikan lampu, sesuai dengan perintah yang diberikan.

2.8 Perancangan Sistem Perangkat Lunak Raspberry Pi

Sistem *smart lamp* merupakan sistem yang selalu aktif untuk mendeteksi suara-suara secara *real time*. Sistem ini akan berhenti hanya ketika perangkat-perangkat sistem ini *off*. Berikut *flowchart* sistem yang diajukan.



Gambar 4 Flowchart perancangan sistem pada Raspberry Pi.

Gambar 4 merupakan *flowchart* dari sistem perancangan yang dibangun pada Raspberry Pi. Raspberry Pi berfungsi sebagai *microcontroller* untuk melakukan proses pengolahan dan pengenalan *Voice Command*. Pengolahan *Voice Command* disini diolah dengan menggunakan Google API yang sudah terintegrasi pada Raspberry Pi.

III. Hasil Pengujian dan Pembahasan

Pengujian menggunakan tiga *device* lampu dengan jarak dan intensitas suara tertentu. Masing-masing *device* diuji untuk dihidupkan dan dimatikan. Percobaan dilakukan sebanyak 10 kali untuk masing-masing *device*. Kemudian dihitung waktu yang dibutuhkan untuk memproses suara dan menghasilkan *output* yang diinginkan.

Untuk lingkup lingkungan pengujian dilakukan di ruangan laboratorium dengan kondisi ideal yaitu tenang dan tanpa *bising/noise*. Hasil pengujian dapat dilihat pada tabel berikut ini.

Tabel 1 Hasil pengujian lampu I.

PERCOBAAN	PERCOBAAN						
	HASIL				JARAK (cm)	INTENSITAS SUARA (dB)	
	ON	WAKTU (s)	OFF	WAKTU (s)		TERBACA SISTEM	TIDAK TERBACA SISTEM
1							
2	1	3	1	2,6	12	63,6	39,9
3	1	3,6	1	4,24	24	64,0	40,65
4	1	3,17	1	3,28	36	62,8	41,1
5	1	3,14	1	2,98	48	60,6	43,3
6	1	3,1	1	3,6	60	61,4	44,8

7	1	3,15	1	2,7	72	61,8	46,3
8	1	3,8	1	3,2	84	61,5	47,4
9	1	3,20	1	3,8	96	63,8	49,7
10	1	3,17	1	4,14	108	61,2	51,4

Tabel 2 Hasil pengujian lampu II

PERCOBAAN							
PERCOBAAN	HASIL				JARAK (cm)	INTENSITAS SUARA (dB)	
	ON	WAKTU (s)	OFF	WAKTU (s)		TERBACA SISTEM	TIDAK TERBACA SISTEM
1							
2	1	3,2	1	2,9	12	61,7	40,3
3	1	3,12	1	3,3	24	60,8	42,2
4	1	3,38	1	3,9	36	61,4	41,8
5	1	3,56	1	3,16	48	61,28	43,5
6	1	2,78	1	3,25	60	63,29	45,8
7	1	3,52	1	3,86	72	62,24	46,4
8	1	3,53	1	4,21	84	61,09	47,1
9	1	3,82	1	3,76	96	63,55	48,6
10	1	3,76	1	3,92	108	63,42	50,8

Tabel 3 Hasil Pengujian Lampu III

PERCOBAAN							
PERCOBAAN	HASIL				JARAK (cm)	INTENSITAS SUARA (dB)	
	ON	WAKTU (s)	OFF	WAKTU (s)		TERBACA SISTEM	TIDAK TERBACA SISTEM
1							
2	1	3,2	1	3,1	12	62,6	38,9
3	1	3,4	1	3,28	24	64,12	40,45
4	1	3,2	1	3,31	36	63,8	41,19
5	1	3,6	1	3	48	61,32	42,3
6	1	3,1	1	3,5	60	62,42	45,35
7	1	3,45	1	2,9	72	61,51	46,66
8	1	3,77	1	3,24	84	63,5	47,84
9	1	3,2	1	3,6	96	64,8	49,23
10	1	3,15	1	3,9	108	62,16	51,52

Dari pengujian yang dilakukan, dapat disimpulkan bahwa rata-rata waktu yang diperlukan sistem untuk menghidupkan dan mematikan Lampu I secara berturut-turut adalah 3,3 *second* dan 3,28 *second* dengan intensitas suara minimal 60,6 dB. Untuk menghidupkan dan mematikan Lampu II secara berturut-turut adalah 3,43 *second* dan 3,61 *second* dengan intensitas suara minimal 60,8 dB. Sedangkan untuk menghidupkan dan mematikan Lampu III secara berturut-turut adalah 3,32 *second* dan 3,39 *second* dengan intensitas suara minimal 61,32 dB. Ketiga lampu dapat dikendalikan dengan jarak terjauh adalah 1,2 meter.

IV. Kesimpulan

Berdasarkan hasil dari perancangan dan pengujian pada sistem *smart lamp*, dapat disimpulkan bahwa sistem *smart lamp* yang dibangun dalam penelitian ini adalah sistem yang

berbasis *voice command* dan menggunakan algoritma Recurrent Neural Networks (RNN), *Long Short-Term Memory Networks* (LSTM) yang sudah terintegrasi pada *Google API* dalam mengolah suara menjadi *output* yang diinginkan. Dalam pengujian sistem, untuk jarak terjauh *user* sebagai sumber suara terhadap sistem adalah 1,2 meter dan didapatkan hasil pengujian yang dilakukan pada Lampu I secara berturut-turut adalah 3,3 *second* dan 3,28 *second* dengan intensitas suara minimal 60,6 dB. Untuk menghidupkan dan mematikan Lampu II secara berturut-turut adalah 3,43 *second* dan 3,61 *second* dengan intensitas suara minimal 60,8 dB. Sedangkan untuk menghidupkan dan mematikan Lampu III secara berturut-turut adalah 3,32 *second* dan 3,39 *second* dengan intensitas suara minimal 61,32 dB.

Penelitian ini memiliki prospek-prospek yang harus dikembangkan untuk penelitian kedepannya. Salah satunya adalah mengembangkan sistem *voice recognition*, yaitu pengenalan *speaker*. Dengan adanya pengenalan *speaker*, sistem yang dibangun akan semakin lengkap, efektif, dan keamanannya akan lebih terjaga. Selain itu, dapat dikembangkan juga sistem *monitoring* status lampu berbasis *Internet of Things* yang dapat memudahkan *user* dalam mengendalikan dan memantau lampu ruangnya walaupun dari jarak jauh.

Referensi

- [1] M. Muslihudin, "Implementasi Aplikasi Rumah Pintar Berbasis Android dengan Arduino Microcontroller," *J. Keteknikan dan Sains*, vol. 1, no. 1, pp. 23–31, 2018.
- [2] D. W. Sudiharto, A. Gautama, P. Satwiko, and R. L. Haposan, "Implementasi Prototipe Automasi Pintu dan Jendela Rumah Melalui Perintah Suara," p. 20, 2016.
- [3] F. Z. Rahman, "Smart Home Berbasis IoT," in *Seminar Nasional Inovasi Teknologi Terapan*, 2017, pp. 369–374.
- [4] M. Sidiq, T. A. B. W, and S. Sa'adah, "Desain dan Implementasi Voice Command Menggunakan Metode MFCC dan HMMs," *Epoch*, vol. 2, no. 1, pp. 1–10, 2012.
- [5] F. Elkusnandi, U. N. Wisesty, F. Informatika, U. Telkom, and A. Buzo, "Implementasi Sistem Pengenalan Ucapan Bahasa Indonesia Menggunakan Kombinasi MFCC dan PCA Berbasis HMM," vol. 5, no. 2, pp. 3608–3622, 2018.
- [6] G. K. Berdibaeva, O. N. Bodin, V. V. Kozlov, D. I. Nefed'Ev, K. A. Ozhikenov, and Y. A. Pizhonkov, "Pre-processing voice signals for voice recognition systems," *Int. Conf. Young Spec. Micro/Nanotechnologies Electron Devices, EDM*, pp. 242–245, 2017.
- [7] A. Ravulavaru, *AI Services Quick Start Guide*. Packt Publishing Ltd, 2018.
- [8] R. Dorf and R. Bishop, *Modern Control System*, 12th ed. London: Pearson Education, 2010.
- [9] M. F. Wicaksono, *Mudah Belajar Raspberry Pi*. Bandung: Informatika, 2018.
- [10] M. Saleh and M. Haryanti, "Jurnal Teknologi Elektro, Universitas Mercu Buana ISSN : 2086 - 9479," *J. Teknol. Elektro, Univ. Buana*, vol. 8, no. 2, pp. 87–94, 2017.
- [11] G. Cloud, "AI & Machine Learning Products," *Google*, 2008. [Online]. Available: <https://cloud.google.com/speech-to-text/>. [Accessed: 02-Sep-2019].
- [12] A. Sandi, "Mengenal Apa itu Web API," *Code Politan*, 2017. [Online]. Available: <https://www.codepolitian.com/mengenal-apa-itu-web-api-5a0c2855799c8>. [Accessed: 13-Aug-2019].
- [13] B. Prijono, "Pengenalan Recurrent Neural Network (RNN) – Bagian 1," 2018. [Online]. Available: <https://indoml.com/2018/04/04/pengenalan-rnn-bag-1/>. [Accessed: 22-Nov-2019].
- [14] K. Dutta and K. K. Sarma, "Multiple feature extraction for RNN-based Assamese speech recognition for speech to text conversion application," *Proc. 2012 Int. Conf. Commun. Devices Intell. Syst. CODIS 2012*, pp. 600–603, 2012.
- [15] B. Prijono, "Pengenalan Long Short Term Memory (LSTM) dan Gated Recurrent Unit (GRU) – RNN Bagian 2," 2018. [Online]. Available: <https://indoml.com/2018/04/13/pengenalan-long-short-term-memory-lstm-dan-gated-recurrent-unit-gru-rnn-bagian-2>. [Accessed: 22-Nov-2019].
- [16] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "LSTM time and frequency recurrence for automatic speech recognition," *2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc.*, pp. 187–191, 2016.