Automatic speech recognition for indonesian medical dictation in cloud environment

Asril Jarin, Agung Santosa, Mohammad Teduh Uliniansyah, Lyla Ruslana Aini, Elvira Nurfadhilah, Gunarso
Research Center for Data and Information Sciences (PRSDI), National Research and Information Agency (BRIN), KST Samaun Samadikun Bandung, Dago, Indonesia

Article Info

Article history:

Received Aug 30, 2023
Revised Oct 12, 2023
Accepted Nov 7, 2023

Keywords:

Indonesia speech corpus
Kaldi automatic speech recognition
Medical dictation
Pychain
Speech recognition

ABSTRACT

This paper introduces *Sistem Pengenalan Wicara untuk Pendiktean Medis* (SPWPM), an automatic speech recognition (ASR) system designed specifically for Indonesian medical dictation. The main objective of SPWPM is to assist medical professionals in producing medical reports and diagnosing patients. Deployed within a cloud computing service architecture, SPWPM strives to achieve a minimum speech recognition accuracy of 95%. The ASR model of SPWPM is developed using Kaldi and PyChain technologies-creating a comprehensive training dataset involving collaboration with Labs247 Company and *Harapan Kita* Heart and Blood Vessel Hospital. Several optimization techniques were applied, including language modeling with smoothing, lexicon generation using the Grapheme-to-Phoneme Converter, and data augmentation. The readiness of this technology to assist hospital users was assessed through two evaluations: the SPWPM architecture test and the SPWPM speech recognition test. The results demonstrate the system's preparedness in accurately transcribing medical dictation, showcasing its potential to enhance medical reporting for healthcare professionals in hospital environments.

Corresponding Author:

Asril Jarin
Research Center for Data and Information Sciences (PRSDI)
National Research and Information Agency (BRIN)
KST Samaun Samadikun Bandung, West Java, 40135, Indonesia
Email: asri003@brin.go.id

1. INTRODUCTION

During the 1990s, automatic speech recognition (ASR) technology gained popularity in US hospitals, allowing doctors to directly convert their speech into notes in electronic health record (EHR) systems [1], [2]. It helped the process of completing patient records, providing doctors with a comfortable and efficient experience. However, the adoption of ASR technology posed challenges due to the need for high accuracy. Researchers have continuously worked on improving ASR accuracy to gain user trust and meet their expectations [3], [4]. A survey conducted by Goss *et al.* [5] revealed that enhanced ASR accuracy resulted in faster and more cost-effective completion of medical reports. The survey, which included 1731 doctors utilizing ASR, found that over 70% of respondents were satisfied with the technology and acknowledged its potential to enhance efficiency. Additional studies conducted by Zuchowski and Goeller [6] demonstrated that using ASR for medical documentation reduced the time required to complete the formula to an average of 5.11 minutes, compared to 8.9 minutes when using traditional typing methods.

In contrast, while English ASR technology has reached human-level accuracy [7], [8], the development of Indonesian ASR technology is still actively pursued by Indonesian researchers and faces significant challenges in reaching a comparable performance level [9]–[13]. Particularly in medical applications, where a minimum accuracy of 95% is required, notable hindrances stemming from regional dialect influences on pronunciation and the absence of standardized medical terminology in Bahasa Indonesia. A critical issue pertains to the scarcity of available Indonesian medical speech corpora. In comparison to the abundance of English resources [14], [15], the current medical corpora [16], [17] for Indonesian are relatively undersized, underscoring the pressing necessity for an expanded and comprehensive Indonesian medical speech corpus.

This paper introduces an innovative automatic speech recognition system for Indonesian medical dictation named *Sistem Pengenalan Wicara untuk Pendiktean Medis* (**SPWPM**, in English: Speech Recognition System for Medical Dictation). This study aimed to produce an **SPWPM's ASR** with a word error rate (**WER**) below 5%, primarily to recognize cardiac surgery medical reports. We are working with Labs247 Company and *Harapan Kita* Heart and Blood Vessel Hospital (RSJPDHK) to make a medical speech corpus. Labs247 carried out the recording process from several speakers who were determined and composed based on dialect, age, and gender. At the same time, RSJPDHK provided training dataset materials from cardiac surgery medical reports. The collection and use of data from hospitals must go through ethical clearance procedures.

The SPWPM's ASR modeling approach leverages the PyChain algorithm, an implementation based on PyTorch. This algorithm enables lattice-free maximum mutual information (LF-MMI) training used in chain models of the Kaldi ASR toolkit [18] as shown in Figure 1. Apart from the training algorithm, several techniques were employed to enhance accuracy, including data augmentation, language model smoothing, and the creation of lexicons using the grapheme-to-phoneme tool. These techniques were implemented to achieve the targeted WER value.

This paper is organized into five chapters. Chapter 2 explains the stages of developing SPWPM's ASR and developing SPWPM into cloud computing services. Chapter 3 describes the evaluation results obtained from two types of SPWPM testing: architectural testing and ASR SPWPM testing by hospital users. Finally, Section 4 concludes the paper.

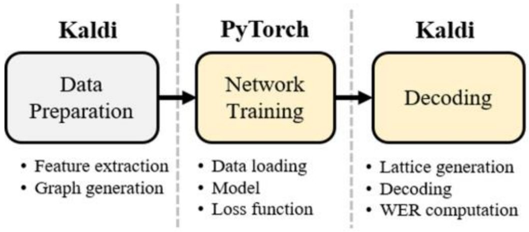

Figure 1. The pipeline of end2end LF-MMI training with PyChain [18]

2. RESEARCH METHOD

The methodology employed in this study, depicted in Figure 2, encompasses two main development paths. The first path involves the ASR development flow for SPWPM speech recognition, encompassing dataset preparation, language model creation, pronunciation list or lexicon generation, data augmentation, ASR modeling using Kaldi and PyChain, and deployment of ASR to the SPWPM service. Evaluations are conducted at various stages to provide feedback for improvement and optimization of preceding steps. For instance, the evaluation of modeling through testing and measurement of WER value yields insights for improving and optimizing language and lexicon models. The outcome of this flow is selecting the best ASR model to be integrated into the SPWPM service system. The second flow pertains to developing the SPWPM as a Service System, involving the specification of system requirements, design of system architecture, implementation and testing of the system, and integration with hospital user applications for field testing.

2.1. ASR development for SPWPM service

The SPWPM's ASR model undergoes multiple development stages, as depicted in Figure 2. First, data preparation is performed, including the collection of speech and medical text corpora. A language model is then built with a smoothing technique, and a pronunciation list is created using graphene-to-phoneme converter software. The speech datasets are augmented with variations in noise and reverberation. The ASR model is trained using Kaldi and PyChain, and its testing and evaluation are carried out iteratively by measuring the WER value. During the evaluation process, the alignment of transcripts and speech phonemes recognized by ASR is visually inspected using an evaluation tool [19]. The evaluation results provide feedback to improve the data preparation process, optimize the language model specification and lexicon, and refine data augmentation techniques. All of them aim to get the best ASR model. Finally, the best ASR model is implemented into the SPWPM service system. Below is a detailed explanation of each of these stages.

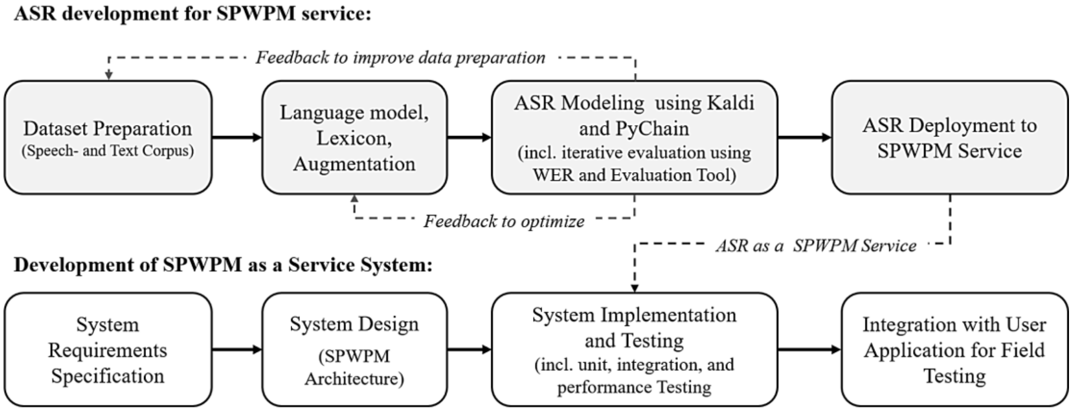
ASR development for SPWPM service:
Feedback to improve data preparation
Dataset Preparation (Speech- and Text Corpus)
Language model, Lexicon, Augmentation
ASR Modeling using Kaldi and PyChain (incl. iterative evaluation using WER and Evaluation Tool)
ASR Deployment to SPWPM Service
Feedback to optimize
ASR as a SPWPM Service
Development of SPWPM as a Service System:
System Requirements Specification
System Design (SPWPM Architecture)
System Implementation and Testing (incl. unit, integration, and performance Testing)
Integration with User Application for Field Testing

Figure 2. Speech recognition system development methodology for medical dictation

2.1.1. Data preparation: speech- and text corpus

The ASR modeling dataset consists of two types of data: the speech corpus and the text corpus. The speech corpus is used for acoustic modeling and involves recording multiple speakers. It is essential to have a diverse range of data and a sufficient dataset size to represent the population accurately. Creating the corpus involves essential tasks such as selecting recorded transcripts from various sources and organizing them based on speaker profiles, including age, gender, and dialect. Supervised recordings are then conducted to meet the specific requirements of the training data, and the recordings are processed to generate training data for ASR acoustic modeling.

This study developed a medical speech corpus for ASR training, comprising 197.5 hours of recordings with over 5,000 unique medical sentences. It included 218 speakers from seven dialect groups, with Javanese, National Standard, and Sundanese being the most prominent. The corpus had an equal gender distribution and covered different age ranges. A portion of this medical speech corpus has been evaluated as a good training dataset for ASR modeling, and the results were published in 2021 in the paper [20]. The medical speech corpus was combined with a general speech corpus from previous studies to create a training dataset [17]. The training dataset for the acoustic model was 225 hours, and a separate 15-hour test dataset was prepared for internal testing. A text corpus was also created to model word order (language model), consisting of sentences from various sources, including transcriptions, question-and-answer sentences, and medical reports. The text corpus contained numerous unique questions, answers, and sentences from doctors' medical reports.

2.1.2. Creating a language model, including smoothing

The effectiveness of ASR systems depends on the language model utilized. Creating a language model involves utilizing appropriate materials like text corpus and audio transcripts, which are preprocessed to ensure consistency, including addressing spelling variations. A smoothing technique is applied to handle Out-of-Vocabullary (OOV). The Modified Kneser-Ney algorithm, developed by James [21], is a widely recognized and effective method for smoothing n-grams, and it was employed in this study for that purpose. To determine the optimal language model, we evaluated two toolkits, Stanford Research Insititute Language Modeling Toolkit (SRILM) [22] and Kenneth Heafield Language Modeling Toolkit (KenLM) [23]. Based on our experimental results comparing the ASR performance using each toolkit, the KenLM-generated language model, incorporating the Modified Kneser-Ney smoothing technique, exhibited superior performance as shown in Table 1.

2.1.3. Creating a pronunciation list using the grapheme-to-phoneme conversion (G2P)

The ASR system also requires a pronunciation dictionary, known as a lexicon, which contains word pronunciations. With thousands of unique words in the language model corpus, manual creation of the lexicon becomes impractical. To address this, we utilize grapheme-to-phoneme conversion (G2P) techniques. We experimented with two G2P toolkits: Sequitur G2P, which employs joint-sequence models [24], and

Phonetisaurus, which utilizes weighted finite-state transducers [25]. Based on our experimental results comparing the ASR performance using each toolkit, the Sequitur G2P demonstrated slightly better performance as shown in Table 2.

To optimize the lexicon generated by Sequitur G2P, we take several steps. We standardize word pronunciation by mapping different phonetic variations, add words not included in the language model using Phonetisaurus, and ensure consistency in surface words to reduce ambiguity. These optimization techniques refine the lexicon, improving the ASR system's accuracy and performance.

Table 1. Comparing ASR Results: SRILM vs. KenLM Based on WER

Experimental ASR model	WER (%)
ASR using KenLM with Smoothing	6.94
ASR using SRILM with Smoothing	7.66

Table 2. Comparing ASR Results: Sequitur G2P vs. Phonetisaurus based on WER

Experimental ASR model	WER (%)
ASR with SEQUITUR based Lexicon	13.59
ASR with PHONETISAURUS based Lexicon	13.61

2.1.4. Data augmentation

Data augmentation is a general strategy adapted to ASR modeling to increase the quantity of training data, prevent overfitting, and increase the robustness of the ASR model. A study by Ragni *et.al.* [26] provides insights that data augmentation is beneficial for speech recognition of low-resource language. Research by Ko *et al.* [27] shows that reverberation and noise data augmentation can reduce WER from 40.9% to 24.6%.

In several ways, we augmented the data with a target of ten times the original data. Several augmentation experiments were done based on several techniques, including variations in the speed of sound; variations in sound volume; addition of noise with the MUSAN corpus (A Music Speech and Noise Corpus) [28]; addition of reverberation sound; and the parcelmouth augmentation [29]. From all those experiments, we decided for our final model only to use the combined augmentation of noise and reverberation.

2.1.5. Modeling speech recognition using the Kaldi ASR toolkit and PyChain

Due to the limited training data, the ASR model for SPWPM was constructed using Kaldi and PyChain [18], [30]. Kaldi is an open-source ASR toolkit with C++ libraries and executables and offers a wide range of sample scripts for building ASR models. Since its introduction in 2011, Kaldi has gained popularity as a framework for developing advanced ASR models, aligning with the current trend of utilizing deep learning platforms for ASR, like PyTorch [31]. Among the successful efforts to bridge Kaldi and PyTorch is PyChain, a flexible and lightweight PyTorch algorithm designed explicitly for lattice-free maximum mutual information (LF-MMI) training.

During the modeling process, multiple iterations were conducted, incorporating variations in the lexicon and text corpus. Nevertheless, this paper explicitly highlights the modeling process that yielded the most favorable outcomes. As depicted in Figure 1, the modeling pipeline involves both Kaldi and PyTorch. Kaldi handles data preparation (feature extraction and graph generation) and decoding (lattice computation generation, decoding, and word error rate calculation) for efficiency and consistency. On the other hand, PyTorch is responsible for loading the data generated by Kaldi and conducting end-to-end model training using a deep neural network. The trained model, saved as a PyTorch model, generates phoneme sequence posteriors. These posteriors are then forwarded to Kaldi for quick decoding or saved to disk for later decoding.

In PyTorch, network modeling utilizes default parameters, including a time delay neural network (TDNN) architecture featuring six convolution layers, an input dimension of 40 using Mel-frequency cepstral coefficients (MFCC), the Adam optimizer method, a learning rate 0.001, and a training duration of 40 epochs. In line with Shao's method [18], the development of the PyChain model encompassed several key steps. First, we partitioned the speech data (in WAV format) into Training, Validation, and Testing Data. Subsequently, we employed Kaldi to process the speech data, extract MFCC features, and then normalize these features using Cepstral mean and variance normalization (CMVN) techniques. Concurrently, we processed the transcript and mono-lingual corpus to construct a Language Model and word lists. Using Kaldi again, we transformed the transcript data, language model, and lexicon into the numerator/denominator graph (FSTs). The heart of the PyChain model lay in the DNN architecture, which was trained using the features obtained from the Training Data and Validation Data prepared in the previous step. Lastly, to determine the optimal model, we rigorously tested it against the Test Data, evaluating the WER value to make the final selection.

2.1.6. ASR performance evaluation with WER

We evaluated ASR performance several times internally by preparing test data and measuring the word error rate (WER) value. WER is a metric used to evaluate the performance of an ASR system, which measures the difference between the recognized words and the actual words spoken by the speaker. It is calculated by dividing the total number of word errors (insertions, deletions, and substitutions) by the total number of words spoken. A lower WER indicates better performance of the ASR system, while a higher WER suggests that the system has more errors in recognizing spoken words [32].

The test data consist of WAV audio data from four testers at *Harapan Kita* Heart and Blood Vessel Hospital in 2021, WAV data on medical reports from 10 speakers who are not part of the recorded speakers for the training dataset, with the data being recorded in 2022, and WAV data from 10 transcripts of surgical medical reports for 2021. WER measurement involves running a pipeline depicted in Figure 3. Initially, the test data undergoes feature extraction, generating input for posteriors generation. Next, PyChain's ASR models process these extracted features to produce posteriors. Subsequently, KALDI tools are used to decode the posterior values, generating text transcripts. The WER value is then calculated by comparing these text transcript results with the reference text transcripts.

2.1.7. Evaluating ASR with a Kaldi-based ASR evaluation tool

To analyze and understand the reasons behind ASR inaccuracies, researchers need a tool that enables visual examination and monitoring of the data generated during the ASR recognition process, including transcripts and speech alignment. A tool developed by Jarin *et al.* [19] serves this purpose, providing researchers with assistance. This tool utilizes the PyKaldi [33] and PyChain modules to display speech waveforms, and MFCC features on an HTML5 canvas.

2.1.8. ASR deployment to SPWPM service

The final ASR model is embedded inside ASR engine and the ASR engine is deployed to the SPWPM service system as a docker in the last phase of developing machine learning-based applications. The ASR functionality is integrated into the SPWPM Backend components by establishing interfaces with other SPWPM components. Within the SPWPM Backend, ASR functions as a service worker, handling the recognition of speech data submitted by users online as shown in Figure 4.

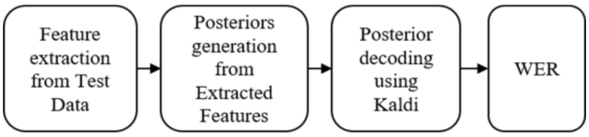
Feature extraction from Test Data
Posteriors generation from Extracted Features
Posterior decoding using Kaldi
WER

Figure 3. The PyChain model WER measurement pipeline

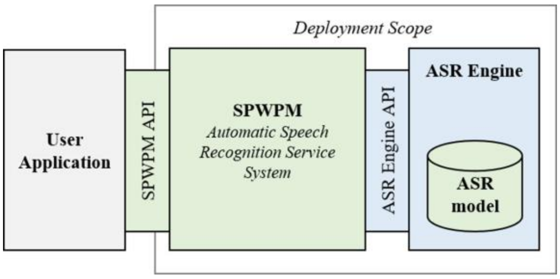

Figure 4. Deployment ASR as a worker from the SPWPM service

2.2. Development of SPWPM as a service system

The development of SPWPM as a service system involves several stages, starting with formulating system requirements and performance expectations. These requirements are documented in the software requirements specification (SRS), a reference for designing and implementing the SPWPM architecture. The architecture is designed to interact with user applications through an application programming interface (API), such as hospital applications. The system components are implemented using multiple docker containers managed and orchestrated on a docker swarm cloud platform. Once the implementation is complete, the

SPWPM service system undergoes testing and integration with hospital user applications, including MEDIS247 and ASRi applications developed by Labs247 Company.

2.2.1. Requirements specification for SPWPM

The system and performance requirements for SPWPM are specified in the SRS document, a living document. SPWPM is designed to integrate seamlessly with user application systems that provide speech recognition services, allowing users to conveniently dictate medical reports using their device's microphone. The system utilizes an ASR Engine API to provide services to user applications. External interface requirements include a TCP/IP network connection and specific criteria for voice data submission, ensuring compatibility. Documentation is provided to guide users in effectively using the ASR engine API. SPWPM ensures a smooth user experience through REST/HTTP APIs with defined endpoints and structured data exchange in JSON format. It can handle multiple connections simultaneously, enhancing flexibility and efficiency. The ASR API converts voice data into transcription text, seamlessly integrating it into user applications. Pronunciation requirements, such as speech rate and volume adjustments, optimize accuracy and resemble the news reporter's voice.

2.2.2. System design: SPWPM architecture

SPWPM architecture as illustrated in Figure 5 is designed to integrate with hospital user applications, providing a speech-based data entry service feature. SPWPM offers an API that user applications can utilize to access ASR services. The backend of SPWPM includes an ASR engine responsible for speech recognition.

The system architecture comprises Middleware (MW), Backend (BE), and Supporting components. Middleware serves as the interface between user applications and the ASR Engine in the Backend. It receives user service requests via RESTful/HTTP APIs, forwards them to the ASR Engine, and sends back transcription results through RESTful/HTTP responses. The Backend works with the ASR engine, waiting for service requests through the messaging system, processing them, and returning the transcription results. Supporting Components include the database, messaging system, and cloud platform. The Database uses PostgreSQL, the messaging system is based on NATS, and the cloud platform chosen is docker swarm, facilitating efficient system management on a simplified infrastructure.

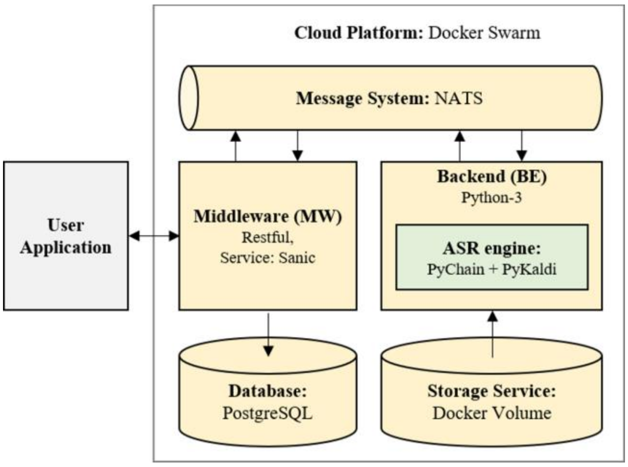

Figure 5. SPWPM architecture

2.2.3. System implementation

In the SPWPM implementation, the MW and BE functions are separated to accommodate their different workload and resource requirements characteristics. MW is designed to operate with limited resources and relies more on networking capabilities. At the same time, BE handles computationally intensive tasks and requires significant computational and storage resources to store ASR models. The separation of MW and BE also allows for independent scalability of the worker nodes in BE, regardless of the number of MW.

NATS serves as the messaging system, facilitating communication between MW and BE by decoupling their interactions. With NATS, MW and BE can communicate without knowing each other's IP addresses. This setup is especially advantageous in a container orchestrator environment like docker swarm. NATS handles the functions of message reception, forwarding, and sharing.

Docker swarm is utilized as the cloud platform because it functions as a container orchestrator, providing management services for active containers. It simplifies the management of the number of active workers as needed. Docker swarm is easy to deploy since it is included in the default docker engine deployment, eliminating the need for complex installation processes. Activating docker swarm only requires running a command on the manager node and the worker node machines.

2.2.4. Testing of SPWPM as a prototype

Before field testing with hospital users, the SPWPM application undergoes several testing stages to ensure compliance with the predefined system requirements. The testing process consists of three main parts: unit testing, system and integration testing, and performance, load, and stress testing. The performance, load, and stress testing results, which include response time and various combinations of workers and clients, will be discussed in the results and discussion section of the research to facilitate further research discussions.

2.2.5. Integration with user application for field testing

The SPWPM service is accessed through a RESTful/HTTP endpoint using the POST method, where messages are sent according to the provided API documentation. Integration of user applications with SPWPM requires a TCP/IP network connection, either through an intranet or the internet, as SPWPM does not have a direct user interface. Users can configure SPWPM using a JSON format configuration file that can be edited using a text editor.

We integrated SPWPM's ASR service to evaluate its performance with the MEDIS247 and ASRi applications. A field test was conducted with four cardiologists from *Harapan Kita* Heart and Blood Vessel Hospital, as shown in Figure 6. The SPWPM system enables medical professionals to transcribe dictations online. It is connected to the user application server through a Restful API with public IP access. Our test used the MEDIS247 and ASRi applications developed by Labs247 Company as user applications, allowing doctors and medical personnel to access the ASR feature integrated into these applications.

The testing process involved three scenarios: two conditioned and one unconditioned test. The conditioned tests were performed by an RS examiner who read pre-prepared transcripts. In the first scenario, the examiner read 50 sentences from a patient's diagnosis report, and in the second scenario, the examiner read five transcripts of surgical medical reports. The unconditioned test was conducted by a *Harapan Kita* Heart and Blood Vessel Hospital examiner who narrated surgical medical reports recorded using the mobile phones of each examining doctor and read a surgical report provided by the examining doctor. The results of the SPWPM field test with four doctors at *Harapan Kita* Heart and Blood Vessel Hospital will be discussed in the Results and Discussion Section, providing valuable material for research discussions.

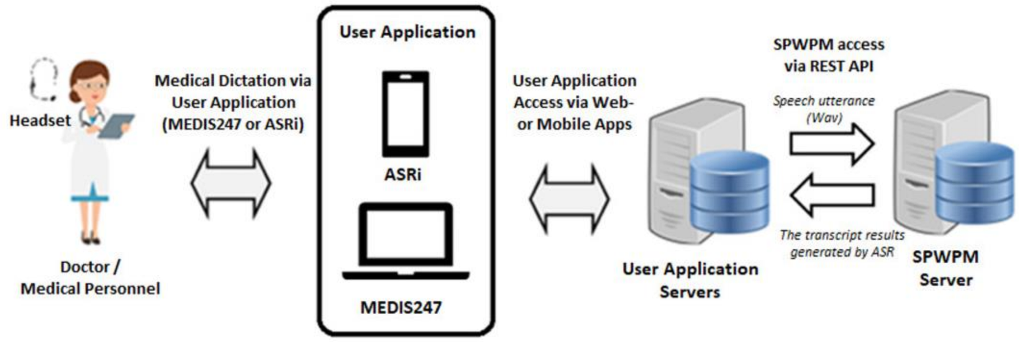
Headset
Doctor / Medical Personnel
Medical Dictation via User Application (MEDIS247 or ASRi)
User Application
ASRi
MEDIS247
User Application Access via Web- or Mobile Apps
User Application Servers
SPWPM access via REST API
Speech utterance (Wav)
The transcript results generated by ASR
SPWPM Server

Figure 6. SPWPM test environment for field testing

3. RESULTS AND DISCUSSION

3.1. SPWPM architecture testing

To evaluate the performance of SPWPM architectures, we conducted an experiment utilizing a set of clients and a deployed SPWPM. The clients were implemented as small Python scripts that utilized the SPWPM API to issue 10 consecutive requests with speech data sourced from a pre-defined collection of WAV files. In addition, each client recorded response time by subtracting the time of the response from SPWPM with the recorded time when the request was sent. The resulting response times were utilized to generate a histogram that displays the distribution of response times for the experimental setup. We evaluated the performance of

SPWPM with configurations of one, two, and three automatic speech recognition (ASR) workers, and we conducted four separate trials, each with a different number of clients (10, 20, 50, and 100).

We generated four response time distribution plots utilizing the approach described above. Specifically, Figure 7 display the response time distribution when handling 10, 20, 50, and 100 clients using 1, 2, and 3 ASR workers. Based on the distribution plots, we observed that for up to 20 clients, a single ASR worker is sufficient. However, for 50 clients, two workers are necessary, and for up to 100 clients, three workers are required. Notably, the mode of the response time from the distribution was around 300 ms. Furthermore, the testing demonstrated the flexibility of adjusting the number of workers in SPWPM deployed on docker swarm.

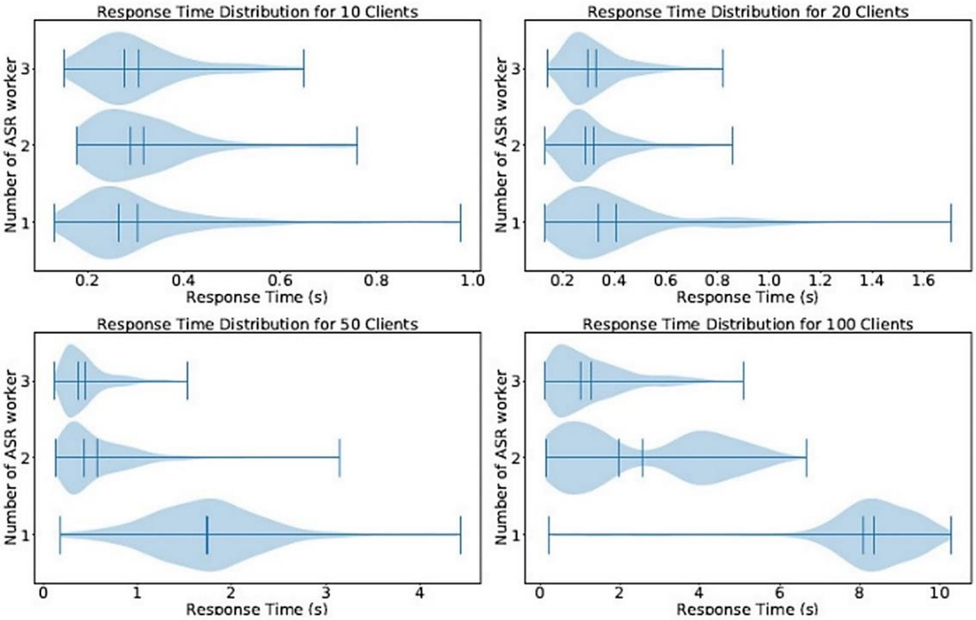

Figure 7. Distribution of response time of 10 to 100 concurrent clients with up to 3 ASR workers

3.2. SPWPM field testing

As explained in the Methodology section and illustrated in Figure 6, after SPWPM was integrated with the user application, we evaluated ASR services with the help of four doctors from *Harapan Kita* Heart and Blood Vessel Hospital through three test scenarios. The results of the three test scenarios are described in Tables 3, 4, and 5. In addition to showing the WER value of each test item, we include data on the duration and speed of the examiner's pronunciation. WER results from all test scenarios show a value below 5%, which is by the ASR performance target we have planned.

Based on the testing results, a scatter plot displayed in Figure 8 illustrates the relationship between utterance speeds and word error rate (WER). The scatter plot reveals a positive correlation between utterance speeds and WER. The calculated R2 score for this correlation is 0.561, indicating a moderately positive relationship. Furthermore, by analyzing the linear model derived from the data, represented by the line in 8, it can be estimated that utterance speeds of up to 140 words per minute (or two words per second) may still result in a WER below 5%.

Table 3. Conditioned testing results with 50 common diagnostic sentences

Doctor	Duration in minutes	Words per minute	Words per second	WER (%)
Male_1	6.2	123.5	2.1	4.57
Male_2	6.7	114.3	1.9	4.47
Female_1	7.8	98.2	1.6	1.95
Female_2	5.6	136.7	2.3	4.30

We did assessment the statistical significance of the relationship between gender and WER using the Pearson Test. The resulting correlation coefficient was 0.20029, indicating a weak positive correlation between

WER scores and gender. Regarding direction, the positive correlation implies that Male speakers show slightly higher WER values than Female speakers. Thus, based on the data analyzed, it can be concluded that gender has a little influenced the performance of SPWPM's ASR.

Table 4. Conditioned testing results for 5 medical surgery reports

Doctor	Duration in minutes	Words per minute	Words per second	WER (%)
Male_1	22.0	75.3	1.3	2.42
Male_2	16.5	100.4	1.7	4.58
Female_1	14.7	112.7	1.9	4.70
Female_2	16.6	99.8	1.7	2.11

Table 5. Unconditioned testing results using own report

Doctor	Duration in minutes	Words per minute	Words per second	WER (%)
Male_1	310.9	52.1	0.9	1.57
Male_2	217.0	72.46	1.21	4.78
Female_1	163.0	92.0	1.5	3.76
Female_2	271.5	49.1	0.8	2.72

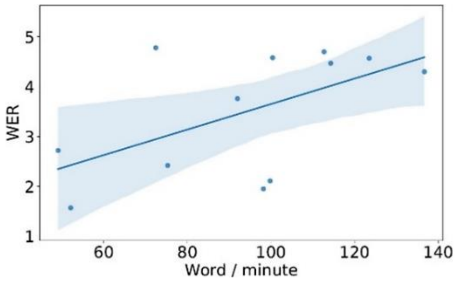

Figure 8. Correlation of pronunciation speed and WER of all test results

4. CONCLUSION

The primary goal of this research was to develop SPWPM, an automatic speech recognition system designed explicitly for Indonesian medical dictation in a cloud computing environment. Our main objective was to assess the performance and feasibility of the system in a hospital setting by achieving a word error rate (WER) below 5%. We employed various ASR enhancement techniques, including creating medical datasets, implementing data augmentation, applying language modeling with smoothing techniques, and utilizing the advanced deep learning algorithm PyChain from Kaldi. Rigorous testing involving conditioned and unconditioned scenarios demonstrated that SPWPM achieved an average WER value below the desired threshold, indicating its suitability for hospital operational use. These results showcase the significant progress made in enhancing the efficiency and accuracy of speech recognition technology for medical dictation. Future research should focus on expanding the SPWPM system's applicability to various medical domains through domain-specific efforts like dataset collection, language model fine-tuning, and adaptation to specialized terminology and speech patterns, enhancing its utility in the medical field.

ACKNOWLEDGEMENTS

This paper acknowledges the support from the Productive Innovative Research Grant (RISPRO) LPDP Competition, Ministry of Finance of the Republic of Indonesia (Letter of Agreement No. PRJ-31/LPDP/2020). We also extend our appreciation to Labs247 Company and the *Harapan Kita* Heart and Blood Vessel Hospital for their valuable cooperation.

REFERENCES

[1] N. Tyler, "Voice recognition," in *New Electronics*, vol. 51, no. 21, Springer New York, 2018, pp. 12–14.

[2] I. Hammana, L. Lepanto, T. Poder, C. Bellemare, and M.-S. Ly, "Speech recognition in the radiology department: A systematic review," *Health Information Management Journal*, vol. 44, no. 2, pp. 4–10, Jun. 2015, doi: 10.1177/183335831504400201.

[3] S. Ajami, "Use of speech-to-text technology for documentation by healthcare providers," *National Medical Journal of India*, vol. 29, no. 3, pp. 148–152, 2016.

[4] M. Johnson *et al.*, "A systematic review of speech recognition technology in health care," *BMC Medical Informatics and Decision Making*, vol. 14, no. 1, Oct. 2014, doi: 10.1186/1472-6947-14-94.

[5] F. R. Goss *et al.*, "A clinician survey of using speech recognition for clinical documentation in the electronic health record," *International Journal of Medical Informatics*, vol. 130, p. 103938, Oct. 2019, doi: 10.1016/j.ijmedinf.2019.07.017.

[6] M. Zuchowski and A. Göller, "Speech recognition for medical documentation: An analysis of time, cost efficiency and acceptance in a clinical setting," *British Journal of Health Care Management*, vol. 28, no. 1, pp. 30–36, Jan. 2022, doi: 10.12968/bjhc.2021.0074.

[7] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Apr. 2018, vol. 2018-April, pp. 5934–5938, doi: 10.1109/ICASSP.2018.8461870.

[8] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," *33rd International Conference on Machine Learning, ICML 2016*, vol. 1, no. December, pp. 312–321, 2015, [Online]. Available: http://arxiv.org/abs/1512.02595.

[9] A. Santosa, A. Jarin, E. M. Yuniarno, H. Riza, and M. H. Purnomo, "OOV handling using partial lemma-based language model in LF-MMI based ASR for Bahasa Indonesia," in *Proceeding of the International Conference on Computer Engineering, Network and Intelligent Multimedia, CENIM 2022*, Nov. 2022, pp. 167–171, doi: 10.1109/CENIM56801.2022.10037479.

[10] K. Ramli and A. Jarin, "New ns-3-based emulation platform for performance evaluation of TCP-based speech recognition," *International Journal of Technology*, vol. 9, no. 4, pp. 852–861, Jul. 2018, doi: 10.14716/ijtech.v9i4.80.

[11] K. Ramli, A. Jarin, and Suryadi, "A real-time application framework for speech recognition using HTTP/2 and SSE," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1230–1238, Dec. 2018, doi: 10.11591/ijeecs.v12.i3.pp1230-1238.

[12] A. Jarin, S. Suryadi, and K. Ramli, "Packet delay distribution model for investigating delay of network speech recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 1, pp. 11–18, Jan. 2017, doi: 10.11591/ijeecs.v5.i1.pp11-18.

[13] D. Hoesen, C. H. Satriawan, D. P. Lestari, and M. L. Khodra, "Towards robust Indonesian speech recognition with spontaneous-speech adapted acoustic models," *Procedia Computer Science*, vol. 81, pp. 167–173, 2016, doi: 10.1016/j.procs.2016.04.045.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Apr. 2015, vol. 2015-August, pp. 5206–5210, doi: 10.1109/ICASSP.2015.7178964.

[15] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text," *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*, pp. 69–71, 2004.

[16] M. R. Qorib and M. Adriani, "Building MEDISCO: Indonesian speech corpus for medical domain," in *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, Nov. 2019, pp. 133–138, doi: 10.1109/IALP.2018.8629259.

[17] H. Riza, E. Nurfadhilah, M. Teduh Uliniansyah, A. Santosa, and L. R. Aini, "An overview of BPPT's Indonesian Language resources," *Alr*, pp. 73–77, 2016.

[18] Y. Shao, Y. Wang, D. Povey, and S. Khudanpur, "PYCHAIN: A fully parallelized pytorch implementation of LF-MMI for end-to-end ASR," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Oct. 2020, vol. 2020-October, pp. 561–565, doi: 10.21437/Interspeech.2020-3053.

[19] A. Jarin *et al.*, "A visual inspection tool for evaluation of ASR model using PyKaldi and PyCHAIN," in *Proceedings - 2022 9th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2022*, Aug. 2022, pp. 61–65, doi: 10.1109/ICITACEE55701.2022.9924132.

[20] E. Nurfadhilah *et al.*, "Evaluating the BPPT medical speech corpus for an ASR medical record transcription system," in *2021 9th International Conference on Information and Communication Technology, ICoICT 2021*, Aug. 2021, pp. 657–661, doi: 10.1109/ICoICT52021.2021.9527450.

[21] F. James, "Modified kneser-ney smoothing of n-gram models," *Research Institute for Advanced Computer Science, Tech. Rep. 00.07*, 2000.

[22] A. Stolcke, "SRILM - An extensible language modeling toolkit," in *7th International Conference on Spoken Language Processing, ICSLP 2002*, Sep. 2002, pp. 901–904, doi: 10.21437/icslp.2002-303.

[23] K. Heafield, "KenLM: Faster and smaller language model queries," *WMT 2011 - 6thWorkshop on Statistical Machine Translation, Proceedings of the Workshop*, pp. 187–197, 2011.

[24] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008, doi: 10.1016/j.specom.2008.01.002.

[25] J. R. Novak, N. Minematsu, and K. Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, pp. 907–938, Sep. 2016, doi: 10.1017/S1351324915000315.

[26] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gales, "Data augmentation for low resource languages," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Sep. 2014, pp. 810–814, doi: 10.21437/interspeech.2014-207.

[27] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Mar. 2017, pp. 5220–5224, doi: 10.1109/ICASSP.2017.7953152.

[28] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, [Online]. Available: http://arxiv.org/abs/1510.08484.

[29] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, Nov. 2018, doi: 10.1016/j.wocn.2018.07.001.

[30] D. Povey, G. Boulianne, L. Burget, P. Motlicek, and P. Schwarz, "The Kaldi speech recognition," *IEEE 2011 workshop on automatic speech recognition and understanding*, 1920.

[31] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1–2, pp. 19–28, Sep. 2002, doi: 10.1016/S0167-6393(01)00041-3.

[33] D. Can, V. R. Martinez, P. Papadopoulos, and S. S. Narayanan, "Pykaldi: A python wrapper for kaldi," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Apr. 2018, vol. 2018-April, pp. 5889–5893, doi: 10.1109/ICASSP.2018.8462463.

BIOGRAPHIES OF AUTHORS

Asril Jarin holds a Doctor of Electrical Engineering from the University of Indonesia in 2017. He also received his Dipl.Ing from Georg-Simon-Ohm Fachhochschule Nuernberg and M.Sc. from Hochschule Darmstadt, Germany, in 2001 and 2003 respectively. Presently, he serves as an associate researcher at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Since 2021, he has been an active researcher in the domain of artificial intelligence, specifically focusing on speech and natural language processing. His research includes speech recognition, speech analytics, sentiment analysis, and natural language processing. Over time, he has contributed with numerous publications in international conferences and journals. For correspondence, please reach out via email: asri003@brin.go.id.

Agung Santosa received the BSEE from University of Toledo, USA, in 1992 and Master of Computer Science from University of Indonesia, Indonesia, in 2015. He is an associate researcher at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Since 2021, he has been an active researcher in the domain of artificial intelligence, specifically focusing on speech and natural language processing. His research includes speech recognition, speech analytics, sentiment analysis, and natural language processing. He can be contacted at email: agun006@brin.go.id.

Mohammad Teduh Uliniansyah received a B.Eng. degree from Shibaura Institute of Technology, Japan, in 1991, a Master of Computer Science degree from Oklahoma State University, USA, in 1998, and a Ph.D. from Keio University, Japan, in 2007. He holds the associate researcher position at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Since 1992, he has been actively engaged in research within artificial intelligence, particularly on speech and natural language processing. His research encompasses speech recognition and analysis, sentiment analysis, and natural language processing. His email address is mted001@brin.go.id.

Lyla Ruslana Aini earned a Bachelor's degree in Computer Science from Sebelas Maret University (Surakarta, Indonesia) in 2012. Lyla is an associate researcher at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Her research interests include speech and natural language processing. Her current projects are related to automatic speech recognition, sentiment analysis, neural machine translation for sign language, and knowledge-graph. Her email address is lyla001@brin.go.id

Elvira Nurfadhilah received the Bachelor and Master of Computer Science from IPB University, Indonesia, in 2011 and 2015. She is an associate researcher at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Since 2016, she has been an active researcher in artificial intelligence, specifically focusing on speech and natural language processing. Her research includes speech recognition and analysis, sentiment analysis, and natural language processing. She can be contacted at email: elvi003@brin.go.id.

Gunarso Graduated in Electrical Engineering from Gadjah Mada University (UGM) Indonesia in 1988. He is an associate researcher at the Research Center for Data and Information Sciences, affiliated with the National Research and Innovation Agency (BRIN) in Indonesia. Since 2021, he has been an active researcher in the domain of artificial intelligence, specifically focusing on speech and natural language processing. His research includes speech recognition, sentiment analysis, and natural language processing. He can be contacted at email: gunarso@brin.go.id