Implementation of Machine Learning Model to Detect Sign Language Movement in SIBI Learning Media

Leni Fitriani[1*], Dede Kurniadi[2], Ilham Syahidatul Rajab[3]

[1,2,3]Department of Computer Science, Institut Teknologi Garut, Garut, West Java, Indonesia
E-mail: [1*]leni.fitriani@itg.ac.id, [2]dede.kurniadi@itg.ac.id, [3]ilhamsyahidatulr@gmail.com

(Received: 27 Nov 2024, revised: 10 Dec 2024, accepted: 11 Dec 2024)

Abstract

This research focuses on the development of a web-based Indonesian Sign Language System (SIBI) learning application with motion detection to improve the precision of sign language practice. Despite the government's introduction of SIBI as an official system, existing platforms lack tools to validate the accuracy of hand movements. Using the Design Sprint methodology—comprising Understand, Define, Sketch, Decide, Prototype, and Validate phases—this study employs Microsoft Azure Machine Learning to create a motion detection model capable of recognizing SIBI gestures. The application offers an interactive learning experience, allowing users to practice and receive real-time feedback on their accuracy. Initial trials demonstrated high prediction accuracy, achieving 99.82% on public datasets and 96.4% on private datasets. Beta testing revealed an 86% satisfaction rate among users, indicating the application's effectiveness in enhancing the learning process. By providing accessibility through standard web browsers and incorporating advanced motion detection, this application contributes to inclusivity, facilitating broader public understanding and interest in learning sign language.

Keywords: Azure, Design Sprint, Learning Media, Machine Learning, SIBI

I. INTRODUCTION

Sign language is a visual language of communication used by the deaf and mute through hand gestures, facial expressions, and body movements to represent letters, numbers, and words [1]. In Indonesia, there are two sign languages: Indonesian Sign Language (BISINDO) and Indonesian Sign Language System (SIBI) [2]. SIBI is based on the American Sign Language (ASL) and is officially used in Special Needs Schools (SLB) under the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek). Nasir's research (2021) showed that students adapt differently to SIBI than their peers [3]. However, sign language communication is hindered by limitations in sign language proficiency [4]. While literature teaches sign language, the focus is on its usage [1]. The government has created the pmpk.kemdikbud.go.id website for SIBI learning through videos, but it needs motion detection. Hence, a machine learning application is required to detect Indonesian sign language hand gestures.

The first referenced study is based on the arrangement of smartphone photos over time, causing users to forget to search for similar photos when capturing an image. This study resulted in an application that categorizes photos using a model with an accuracy of 96%, precision of 93%, and success rate of 93% [5], [6]. The second referenced study is based on deaf students who understand words through images and require supplementary aids. This study produced a video-based educational application to assist the learning process [7]. The third study is motivated by non-verbal communication challenges among some individuals. Sign language is used to communicate with disabled individuals. This study created a prototype application using Microsoft Kinect to detect hand movements with an accuracy ranging from 75% to 87.5% [8]. The fourth study discusses the challenges disabled individuals face in communicating and interacting with others due to varying sign language comprehension. It resulted in an application that detects movements and provides text and audio outputs. The model achieved an accuracy of 94%, precision of 99.9%, and recall of 100% [9]. The fifth reference study is based on the need for intermediaries in translating sign language to facilitate communication between disabled individuals and others. This study employed the Haar Classifier method with the K-Nearest Neighbors (K-NN) algorithm, achieving an average test result of 91.8% and the ability to translate 24 letters [10].

Previous studies on motion detection and sign language learning provide valuable insights into addressing these limitations. For instance, systems like Leap Motion combined with Hidden Markov Models (HMC) have demonstrated gesture recognition accuracy of 86.1% for ASL, although requiring specialized hardware poses accessibility challenges [11]. Real-time recognition systems, such as one for Indian Sign Language using MediaPipe and LSTM models, achieved an impressive 92% accuracy and emphasized the need for adaptable and accessible solutions [12]. Additionally, transformer-based architectures for generating detailed sign language motions have showcased potential for improved gesture precision and contextual accuracy [13].

Despite these advancements, challenges remain. The reliance on specialized devices like Leap Motion or Kinect reduces accessibility, while datasets often lack diversity to account for dynamic gestures such as "Z" or individual physical differences [14]. This study addresses these issues by developing a web-based SIBI learning application incorporating motion detection powered by Microsoft Azure Machine Learning. The proposed system is designed to improve inclusivity and accuracy by allowing gesture validation via standard webcams, thus bridging the gap in accessibility while enhancing the learning process for the Indonesian Sign Language System.

Based on the above background and research, it can be concluded that sign language is crucial to learn as a means of nonverbal communication. Educational media can serve as a tool in sign language learning. Therefore, this study proposes the implementation of Microsoft Azure Machine Learning to Detect Sign Language Movements in the SIBI Learning Media, aiming to facilitate the public in learning and practicing the Indonesian Sign Language System.

II. METHOD

In system design, several stages of activities are outlined in the conceptual framework that illustrates the logical flow of the research. The following Figure 1 is the conceptual framework of this study.

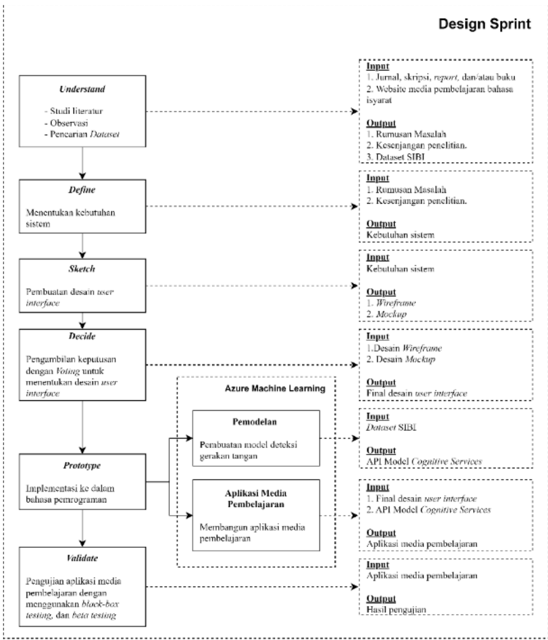

Figure 1. Conceptual Framework

In detail, the research process depicted in the diagram in Figure 1 consists of several steps explained as follows:

1. **Conceptual Understanding**
   Achieved through literature review and observation. Literature review and observation were conducted to understand sign language-based learning media. Data was collected from diverse sources such as journals, theses, reports, and relevant books. Observations included a review of existing sign language learning platforms. The dataset used for training and testing the model was retrieved from Kaggle (https://www.kaggle.com/mlanangafkaar/datasets-lemlitbang-sibi-alphabets). It contains 1,312 labeled images of SIBI alphabets: 1,092 images were allocated for training the model, and 220 were reserved for validation. Each image represents static hand gestures corresponding to letters in SIBI, as shown in Figure 2 for the letter 'A.'

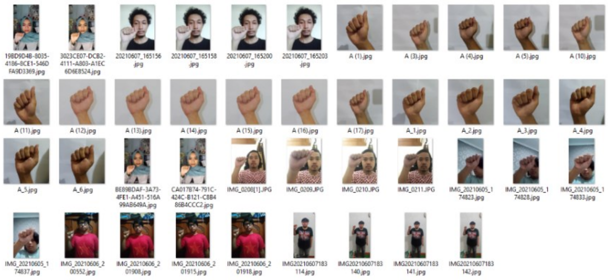

Figure 2. Dataset for the Letter 'A'

2. **Defining System Requirements**
   System requirements were derived from insights obtained during the conceptual stage. This included identifying the application's objectives, target users, required devices, and specifications for the learning media website. The system requirement document also detailed the menu structure design, ensuring clarity and usability for users.Sketch, the initial design of the learning media application is created based on the previous system requirements. The main output is user interface design in the form of wireframes and application mockups.
3. **Sketching the Initial Design.**
   Based on the established system requirements, wireframes and application mockups were developed to design the user interface (UI). These sketches served as the blueprint for creating an intuitive and user-friendly learning experience.
4. **Decision-Making on UI Design**.
   After creating the wireframes, decision-making was facilitated through voting or polling among stakeholders and potential users. The most preferred UI design was selected for implementation.
5. **Prototyping and Machine Learning Model.**
   The motion detection model was developed using Microsoft Azure Machine Learning. Cognitive Services and Azure's Custom Vision were used to create and train a model that recognizes SIBI gestures. The dataset images were preprocessed to enhance contrast and remove noise.

ISSN 2549-8037, EISSN 2549-8045
DOI: 10.34148/teknika.v14i1.1159

Model Training and Export: The model was trained using the Kaggle dataset, applying supervised learning techniques. Once trained, the model achieved a prediction accuracy of 99.82% on the public dataset and 96.4% on private validation data. It was then exported to TensorFlow.js for web-based deployment.

Model Evaluation: Evaluation metrics such as precision, recall, and F1-score were calculated to assess the model's robustness. Additional stress tests were performed under varying lighting and background conditions to ensure real-world applicability.

6. **Validation and Testing**

Black-Box Testing: The application's functionality was rigorously tested to identify any bugs or errors.

Beta Testing: The application was distributed to a group of users, and feedback was collected through online questionnaires. The results indicated an **86% satisfaction rate**, affirming the application's effectiveness in enhancing SIBI learning.

Real-World Scenarios: Tests were conducted in different environments to validate the model's ability to detect gestures accurately under diverse conditions, such as varying hand sizes, skin tones, and lighting.

III. RESULT AND DISCUSSION

In system design, several stages of activities are outlined in the conceptual framework that illustrates the logical flow of the research. The following is the conceptual framework of this study. In designing the system, several stages of activities are outlined in the conceptual framework that illustrates the logical flow of the research. The following is the conceptual framework of this study:

1. Understand

The initial step involves conceptual understanding through literature study and observation. This includes collecting data from previous research such as journals, theses, and relevant books and observing sign language learning platforms based on websites and applications. Subsequently, the Indonesian Sign Language System (SIBI) dataset is searched from public sources like Kaggle. The dataset used is from 'Datasets SIBI Sign Language Alphabets' by M. Lanang Afkaar, containing 1,312 sign language movement photos for 26 letters, with 1,092 pictures for training data and 220 images for validation.

2. Define

The second stage involves determining the system requirements, including application goals, user targets, device specifications, and menu structure. The application aims to create a sign language learning platform with machine learning integration for direct practice and movement assessment. The target users for the application are the public. The application's Menu Structure consists of a Home Page, a choice letter Page, a Learning Page, and an About Us Page. Details can be found in the Table 1 below.

Table 1. Summary of Tips & Tricks for a Good Scientific Article

No.	Page	Explanation
1	Home Page	Displays the website logo, button to access letter selection, and some website information
2	Selection Page	Presents letter cards to navigate to the learning page
3	Learning Page	Shows images, videos, and gesture detection of sign language
4	About us Page	Features creators, mentors, and used assets for the website

3. Sketch

In the third stage, a rough design of the sign language learning website application is created based on the defined system requirements. During this stage, several designs are crafted.

A. Wireframe

Wireframes are used as the foundation for UI design. The wireframe designs created are depicted in Figure 3.

Figure 3. Wireframes

B. Mockups

Mockups involve creating UI while considering the layout of the wireframes. The mockup designs created are shown in Figure 4.

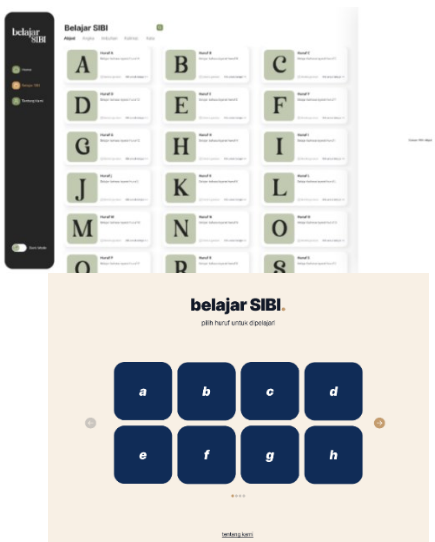

Figure 4. Mockups

4. Decide

In the fourth phase, UI design was selected through a survey using a Google Form filled out by 84 respondents (63.1% female, 36.9% male) aged 7-30 years. The design with the most votes was chosen as the app's UI. The design selection results can be seen in the Table 2 below.

Table 2. Decide Phase Questionnaire

Questions	UI Website 1	UI Website 2	UI Website 3
I'm more interested in the website design's color combination.	12	20	52
I'm more interested in the website's font usage.	23	18	43
I'm more interested in the website layout design.	13	31	40
I'm more interested in the use of images in the website design.	19	15	50
I'm more interested in the use of images in the website design.	10	25	49

From the table, most potential users preferred the third UI design for implementation. The questionnaire also indicated high interest in learning sign language, with 25% strongly agreeing, 54.8% agreeing, 16.7% somewhat disagreeing, and 3.6% disagreeing.

5. Prototype

In the fifth stage, the activities involve creating a model using Microsoft Azure Machine Learning and implementing the model into a programming language.

A. Create a model

Creating a model in Microsoft Azure Machine Learning involves the following steps: Firstly, create a resource in Azure through portal.azure.com and select "Create a resource." Then, locate the Custom Vision service. After setting up the Custom Vision resource in Azure, the next step is to create a project in Custom Vision (customvision.ai) and fill in the project details. For instance, the "Name" section is filled with "belajarSIBI," and the "Description" provides a project overview. Selecting a "Classification" project type, with "Classification Types" set to Multiclass for image classification. The "domain" section is set to General Compact for model export to Tensorflow.js. The dataset is then uploaded to the project, with images of at least 1,000px x 1,000px resolution in .jpg format. After uploading the dataset, the model is trained using the "Train" button and Advanced Training options. Post-training evaluation yields precision and recall rates of 99.1% and Additional Performance reaching 100%, as seen in Figure 5.

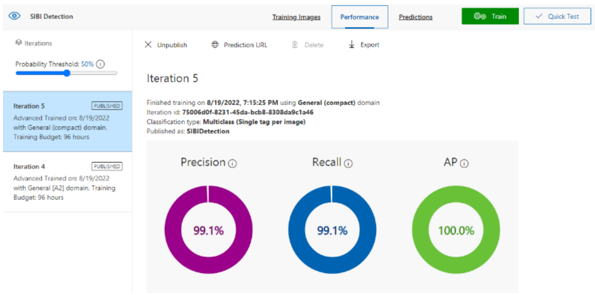

Figure 5. Model Evaluation

Testing on public datasets shows an average prediction rate of 99.82%, while private dataset testing, as shown in Figure 6, reaches 96.4%.

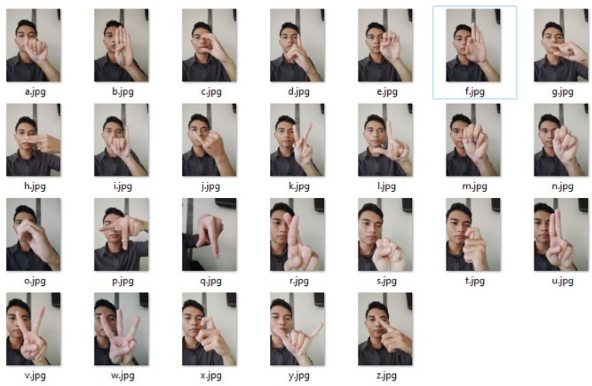

Figure 6. Personal Dataset

Here are the test results of the model as shown in Table 3 below.

Table 3. Test Result of the Model

Data	Prediction Accuracy From SIBI Dataset	Prediction Accuracy From Personal Dataset
A	100%	97.9%
B	100%	99.9%
C	100%	99.9%
D	100%	99.9%
E	99.9%	96.9%
F	100%	87.6%
G	100%	86.7%
H	100%	99.4%
I	100%	99.9%
J	100%	99.9%
K	100%	99.6%
L	100%	86.7%
M	99%	93.1%
N	99.9%	96.8%
O	100%	94.9%
P	100%	100%
Q	100%	100%
R	100%	86.7%
S	100%	99%
T	100%	99.9%
U	99.9%	99.9%

ISSN 2549-8037, EISSN 2549-8045
DOI: 10.34148/teknika.v14i1.1159

Data	Prediction Accuracy From SIBI Dataset	Prediction Accuracy From Personal Dataset
V	96.5%	86.7%
W	100%	99.9%
X	100%	94.8%
Y	100%	99.9%
Z	100%	99.3%

B. Implementation

Implement the Model into the Programming Language, from system requirements and detection model to the chosen UI. The display of the implementation results in the programming language using the Next.js and Tailwind CSS framework can be seen in Figures 7 through 11.

Figure 7. Home Page

Figure 8. Selection Page

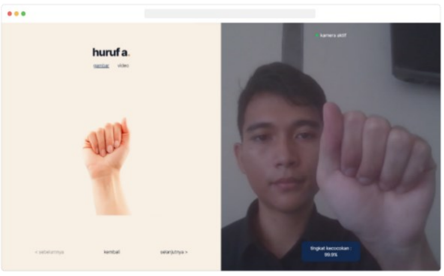

Figure 9. Image Learning Page

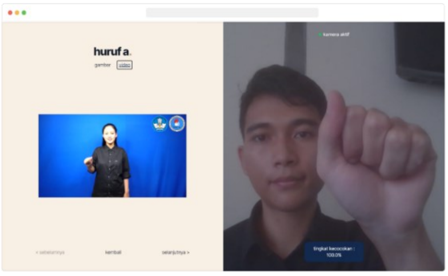

Figure 10. Video Learning Page

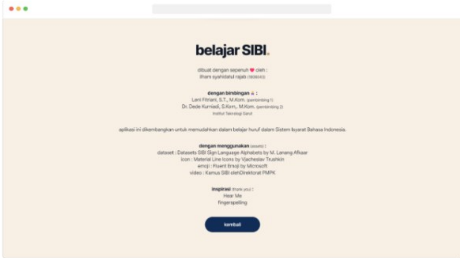

Figure 11. About Us Page

The hand gesture detection model results are presented in the Table 4 below.

Table 4. Results of the Hand Gesture Detection

Data	Successful/Not Successful (with a matching rate >85%)
A	Success
B	Success
C	Success
D	Success
E	Success
F	Success
G	Success
H	Success
I	Success
J	Success
K	Success
L	Success
M	Success
N	Success
O	Success
P	Success
Q	Success
R	Success
S	Success
T	Success
U	Success
V	Success
W	Success
X	Success
Y	Success
Z	Fail

From the table above, the results are promising, except for the letter Z, which wasn't detected due to its dynamic movement involving the index finger. Despite being dynamic, the letter J was classified correctly because the dataset includes images of its final hand position, aiding the model's detection.

6. Validate

The next step is testing the educational media application using black-box and beta testing methods. This stage aims to identify errors, weaknesses, and the alignment of the application with the design objectives.

A. Black-Box Testing

DOI: 10.34148/teknika.v14i1.1159

The application developer conducts black-box testing to ensure the smooth operation of the application without any errors. The results of the Black-box testing are presented in the Table 5 below.

Table 5. Black-Box Testing

Activity	Test Class	Testing Scenario	Expected Outcome	Result
Website Access	Correct URL (belajarsibi.vercel.app)	Entering the complete URL	Homepage will appear	Correct
	Incorrect URL (belajarsibi.vercel.app/test)	Entering an incomplete URL	404 Page will appear	Correct
Home Page	Mulai Belajar Button	Pressing the Mulai Belajar Button	Move to the select letter page	Correct
Selection Page	Data Loaded Successfully	Loading data	Display loading and show all data	Correct
	Data Failed to Load	Failed to load data	Display a failed-to-load data message	Correct
	Pagination	Navigating to find other letters	Letter selection changes	Correct
	Tentang Kami Button	Pressing the Tentang Kami Button	Move to the About page	Correct
	Letter Selection Cards	Clicking on a card	Move to the learning page	Correct
About Page	Displaying Asset Data	Clicking on an assets link	The page will switch to the respective assets' URL	Correct
	Back Button	Pressing the Back Button	Move back to the select letter page	Correct
Learning Page	Webcam Access Approved	User approves webcam access	The camera will appear, and a green indicator will indicate that the camera is active.	Correct
	Webcam Access Denied	The user denies webcam access.	The camera area will turn black, and a red indicator with "camera not active" text will appear, along with steps to activate the camera.	Correct
	Loading Image and Video Data	Loading data	Display loading and show data	Correct
	Failed to Load Image and Video Data	Failed to load data	Display a was unable to load data message	Correct
	Video Button	Pressing the Video Button	Display a video corresponding to the letter	Correct
	Gambar Button	Pressing the Gambar Button	Display the image corresponding to the letter	Correct
	Back Button	Pressing the Back Button	Move to the select letter page	Correct
	Next Button	Pressing the Next Button	Move to the next letter learning page	Correct
	Next Button is on Letter Z	Pressing the Next Button if on Letter Z	Nothing will happen	Correct
	Previous Button	Pressin the Previous Button	Move to the last learning letter page	Correct
	Previous Button is on Letter A	Pressing the Previous Button if on Letter A	Nothing will happen	Correct
	Requested Gesture Match Level	Performing the gesture corresponding to the letter	The matching level will increase	Correct
	Incorrect Gesture Match Level	Performing a gesture not corresponding to the letter	The matching level won't increase or will be less than 30%	Correct

B. Beta Testing

Beta testing, the application is distributed, and a questionnaire is uploaded online via Google Forms to a group of users. The goal is to assess the application's compatibility with user needs and gather feedback for improvements. The testing involves 30 respondents, adhering to an appropriate sample size [15]. The questionnaire consists of 10 questions using a Likert scale from 1 to 4. Here are the Likert scale assessment results for each question in the questionnaire. The percentage for each answer is calculated using the Formula 1 [16]:

$$Y = \frac{\Sigma(N \cdot R)}{ideal\ score} \times 100\% \quad (1)$$

Here are the beta testing results conducted with input from 30 respondents (Table 6).

DOI: 10.34148/teknika.v14i1.1159

Table 6. Beta Testing

Beta Testing	Answer	N	R	N·R	Y
What is your impression of the appearance or User Interface of the SIBI app?	Strongly Agree	4	17	68	86%
	Agree	3	10	30	
	Disagree	2	2	4	
	Strongly Disagree	1	1	1	
Do you find the app's user interface easy to understand and navigate?	Strongly Agree	4	21	84	88%
	Agree	3	4	12	
	Disagree	2	4	8	
	Strongly Disagree	1	1	1	
How comfortable do you feel using this app's sign language motion detection feature?	Strongly Agree	4	15	60	85%
	Agree	3	13	39	
	Disagree	2	1	2	
	Strongly Disagree	1	1	1	
Does the sign language motion detection feature help you better understand sign language movements?	Strongly Agree	4	14	56	84%
	Agree	3	14	42	
	Disagree	2	1	2	
	Strongly Disagree	1	1	1	
In your opinion, how accurate is the sign language motion detection feature in recognizing sign language movements?	Strongly Agree	4	11	44	82%
	Agree	3	17	51	
	Disagree	2	1	2	
	Strongly Disagree	1	1	1	
Does the sign language motion detection feature enrich your sign language learning experience?	Strongly Agree	4	21	84	91%
	Agree	3	8	24	
	Disagree	2	0	0	
	Strongly Disagree	1	1	1	
The displayed hand motion images help me practice sign language.	Strongly Agree	4	19	76	89%
	Agree	3	10	30	
	Disagree	2	0	0	
	Strongly Disagree	1	1	1	
The displayed hand motion videos help me practice sign language.	Strongly Agree	4	22	88	91%
	Agree	3	6	18	
	Disagree	2	1	2	
	Strongly Disagree	1	1	1	
Do you believe this app has the potential to support individuals with hearing impairments in learning sign language?	Strongly Agree	4	17	68	88%
	Agree	3	12	36	
	Disagree	2	0	0	
	Strongly Disagree	1	1	1	
After trying the SIBI learning app, I am interested in learning sign language further.	Strongly Agree	4	12	48	81%
	Agree	3	14	42	
	Disagree	2	3	6	
	Strongly Disagree	1	1	1	

The results of the Beta Testing in the table above indicate that the SIBI learning application aligns with its objectives, featuring a well-designed User Interface, a practical sign language gesture detection feature with sufficiently high accuracy, and user enthusiasm to study sign language further. With an average questionnaire score of 86%, the SIBI application has demonstrated excellence in its suitability as an educational medium, as presented in the Table 7 below [17].

Table 7. Range of Media Feasibility Percentages

Interval	Criteria
85% - 100%	Excellent
69% - 84%	Good
53% - 68%	Satisfactory
37% - 52%	Fair
20% - 36%	Poor

C. User Trials and Feedback

User trials were conducted with a group of 30 participants, including students and educators from Special Needs Schools (SLBs). The participants used the SIBI learning application for gesture recognition and learning over a period of one week. Feedback was collected through questionnaires and interviews to assess user experience, learning outcomes, and motion detection performance. Key Findings from User Trials:

DOI: 10.34148/teknika.v14i1.1159

1. Motion Detection Accuracy: The system achieved an average detection accuracy of 96.4% across real-world scenarios, consistent with validation results from training. Dynamic gestures (e.g., "Z" and "J") presented marginally lower accuracy due to dataset limitations.
2. User Satisfaction: 86% of participants expressed satisfaction with the application, citing its intuitive interface and real-time feedback as major strengths.
3. Learning Outcomes: 70% of users demonstrated improved accuracy in performing SIBI gestures after two learning sessions, showing the application's potential for effective skill development.

Comparison with Existing Applications or Methods

The proposed SIBI learning application was benchmarked against three existing platforms and methods (Table 8), focusing on motion detection accuracy, user interface, and learning outcomes.

Table 8. Comparison with Existing Applications or Methods

Feature	Proposed Application	Existing Applications	Description
Motion Detection Accuracy	96.4% (real-world)	92% for MediaPipe with LSTM [12]	MediaPipe models demonstrate strong accuracy but are less accessible due to the requirement for complex libraries and setup.
User Interface	Modern, web-based, accessible on devices with webcams	Hardware-dependent interfaces (e.g., Leap Motion, Kinect) [11]	The proposed system is accessible via standard devices (e.g., webcams) and does not require additional hardware, enhancing usability.
Learning Outcomes	70% improvement in gesture accuracy	Limited feedback, no real-time validation [11]	Provides interactive learning with real-time feedback, enhancing engagement.

Discussion

The results highlight the effectiveness of the proposed system in improving motion detection and learning outcomes compared to existing methods. Its web-based design ensures accessibility for a wide range of users without the need for specialized hardware. However, dynamic gestures remain an area for improvement, suggesting the need for expanded datasets and further model refinement. User feedback underscored the importance of intuitive interfaces and real-time validation in driving learning engagement.

This study sets a new standard in technology-assisted SIBI learning by integrating high-accuracy motion detection with an accessible platform. Future work should focus on enriching gesture datasets and incorporating features to adapt to diverse user needs and environments.

IV. CONCLUSION

This study successfully developed a motion detection model using Microsoft Azure Machine Learning, achieving a precision of 99.1%, recall of 99.1%, and additional performance of 100%. The model demonstrated high accuracy, with public dataset testing yielding an average prediction rate of 99.82% and private dataset testing reaching 96.4%. The resulting web-based SIBI learning application was praised for its user-friendly interface and appealing visual design. However, challenges were noted in detecting dynamic gestures, particularly the letter "Z," highlighting limitations in the dataset and model’s adaptability. Additionally, the application requires a webcam for gesture detection, which may limit accessibility in environments with inconsistent lighting or low-resource settings. Environmental factors such as lighting variability, diverse backgrounds, and differences in user hand shapes also reduced detection accuracy in some real-world scenarios.

Future research should focus on addressing these limitations to enhance the application’s effectiveness and adaptability. Expanding the dataset to include dynamic gestures, numbers in SIBI, and diverse user demographics (e.g., different skin tones and hand sizes) would improve the model’s generalizability. Incorporating synthetic data generation techniques could help address gaps in real-world variability. To further improve accuracy in real-world conditions, algorithms should be developed to adapt to environmental changes, such as lighting normalization and background removal. Additionally, extending the application to support other sign languages like BISINDO or ASL could broaden its usability. Enhancing the user experience by incorporating gamification and personalized feedback features would make learning more engaging and effective. Cross-platform accessibility, including a mobile version of the application, would ensure broader reach, particularly in low-resource settings. Finally, integration with assistive technologies such as speech-to-text and text-to-speech features could facilitate two-way communication between hearing and non-hearing users. Addressing these recommendations will enable future iterations of the application to support diverse learners and environments mor effectively.

REFERENCES

[1] R. I. Borman and B. Priyopradono, “Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA),” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 103–108, 2018.

[2] D. Yolanda, K. Gunadi, and E. Setyati, “Pengenalan alfabet bahasa isyarat tangan secara real-time dengan menggunakan metode Convolutional Neural Network dan Recurrent Neural Network,” *J. Infra*, vol. 8, no. 1, pp. 203–208, 2020.

[3] A. D. Saputra, J. Jayanta, and A. B. Pangaribuan, “Klasifikasi Alfabet Bahasa Isyarat Indonesia (Bisindo) Dengan Metode Template Matching Dan K-Nearest

DOI: 10.34148/teknika.v14i1.1159

Neighbors (Knn)," in *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2020, pp. 747–760. Accessed: Dec. 10, 2024. [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/563

[4] N. A. M. Amin and F. Pribadi, "Urgensi Bahasa Isyarat dalam Pendidikan Formal Sebagai Media Komunikasi dan Transmisi Informasi Penyandang Disabilitas Rungu dan Wicara," *J. Sos.*, vol. 77, 2022.

[5] L. Fitriani, D. Tresnawati, and M. B. Sukriyansah, "Image Classification On Garutan Batik Using Convolutional Neural Network with Data Augmentation," *JUITA J Inf.*, vol. 11, no. 1, p. 107, 2023.

[6] C. B. Takapente, S. R. Sompie, and V. C. Poekoel, "Implementasi Azure Cognitive Service Untuk Aplikasi Pengkategorian Foto," *J. Tek. Inform.*, vol. 13, no. 4, 2018, Accessed: Dec. 10, 2024. [Online]. Available: https://ejournal.unsrat.ac.id/index.php/informatika/article/view/28093

[7] Z. Nikolawatin, P. Setyosari, and S. Ulfa, "Pengembangan media tutorial bahasa isyarat untuk siswa tunarungu SLB BC Kepanjen," *Jinotep J. Inov. Dan Teknol. Pembelajaran Kaji. Dan Ris. Dalam Teknol. Pembelajaran*, vol. 6, no. 1, pp. 15–22, 2019.

[8] A. B. Yunanda, F. Mandita, and A. P. Armin, "Pengenalan bahasa isyarat indonesia (bisindo) untuk karakter huruf dengan menggunakan microsoft kinect," *Fountain Inform. J.*, vol. 3, no. 2, pp. 41–45, 2018.

[9] S. Hidayatullah, "Rancang Bangun Penerjemah Bahasa Isyarat Menggunakan Pengolahan Citra Dengan Metode You Only Look Once (YOLO)," PhD Thesis, Politeknik Perkapalan Negeri Surabaya, 2021. Accessed: Dec. 10, 2024. [Online]. Available: http://repository.ppns.ac.id/4239/

[10] R. I. Borman, B. Priopradono, and A. R. Syah, "Klasifikasi Objek Kode Tangan pada Pengenalan Isyarat Alphabet Bahasa Isyarat Indonesia (Bisindo)," in *SNIA (Seminar Nasional Informatika dan Aplikasinya)*, 2019, p. D-1. Accessed: Dec. 10, 2024. [Online]. Available: https://snia.unjani.ac.id/web/index.php/snia/article/view/87

[11] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak, "Recognition of American sign language gestures in a virtual reality using leap motion," *Appl. Sci.*, vol. 9, no. 3, p. 445, 2019.

[12] A. Ganpatye and S. Mane, "Motion Based Indian Sign Language Recognition using Deep Learning," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Jun. 2022, pp. 1–6. doi: 10.1109/CONIT55038.2022.9848275.

[13] M. Gil-Martín, M. Villa-Monedero, A. Pomirski, D. Sáez-Trigueros, and R. San-Segundo, "Sign Language Motion Generation from Sign Characteristics," *Sensors*, vol. 23, no. 23, p. 9365, 2023.

[14] H. Luqman, "An efficient two-stream network for isolated sign language recognition using accumulative video motion," *IEEE Access*, vol. 10, pp. 93785–93798, 2022.

[15] S. Sholichin, "Pengembangan dan pengujian aplikasi pemesanan makanan berbasis website menggunakan metode waterfall," *J. Comput. Sci. Eng. JCSE*, vol. 2, no. 1, pp. 40–50, 2021.

[16] A. Rosano, "Pengujian Alpha dan Beta pada Pengembangan Sistem Internet Banking (Ibank) PT Bank Mega, Tbk," *REMIK Ris. Dan E-J. Manaj. Inform. Komput.*, vol. 3, no. 2, pp. 34–40, 2019.

[17] F. F. Dewi and S. L. Handayani, "Pengembangan media pembelajaran video animasi en-alter sources berbasis aplikasi powtoon materi sumber energi alternatif sekolah dasar," *J. Basicedu*, vol. 5, no. 4, pp. 2530–2540, 2021.