SIBI (Sign System Indonesian Language) Text-to-3D Animation Translation Mobile Application

Erdefi Rakun, Sultan Muzahidin, IGM Surya A. Darmana, Wikan Setiaji
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia

***Abstract*—This research proposed a mobile application prototype to translate Indonesian text into SIBI (Sign System for the Indonesian Language) 3D gestures animation to bridge the communication gap between the deaf and the other. To communicate in sign language, the signer will use his/her hands and fingers to demonstrate the word gesture, and at the same time, his/her mouth will pronounce the word being expressed. Therefore, the proposed mobile application needs two animation generator components: the hand gesture and the lip movement generator. Hand gestures are made using a motion capture sensor. Mouth movements are created for all syllables available in the SIBI dictionary using the Dirichlet Free-Form Deformation (DFFD) method. The subsequent challenging work is synchronizing these two components and adding transitional gestures. A transitional gesture done by the cross-fading method is needed to make a word gesture that can smoothly connect with the next word gesture. The Mean Opinion Score (MOS) test was run to measure the mouth movements in 3D animation. The MOS score is 4.422. There are four surveys conducted to measure user satisfaction. The surveys showed that the animation generated did not significantly differ from the original video. The Sistem Usability Score (SUS) is 76.25. The score means that prototype is in the GOOD category. The average time needed to generate an animation from Indonesian input text is less than 100ms.**

***Keywords*—SIBI sign language; sequence generation; visual speech; animation**

I. Introduction

Sign language is a non-verbal language used to help people with hearing impairment communicate. Sign language represents a word with hand gestures and mouth movements. Communication with sign language uses a combination of hand, finger, and mouth movements representing words [1]. Indonesia has two sign languages acknowledged by the government: SIBI (Sign System for the Indonesian Language) and BISINDO (Indonesian Sign Language).

SIBI is a sign language that the Ministry of Education and Culture officially acknowledged in Indonesia in 1994. SIBI follows the Indonesian language grammar and has been used formally in the School for special needs students. The characteristic of SIBI is that SIBI applies Indonesian grammar in organizing word gestures in a sentence [2]. Indonesian words are written using the Latin-Roman alphabet and categorized into four elements: subject, verb, noun, and adverb. Indonesian also has inflectional words that attach prefixes, suffixes, and affixes to the root word. With these affixes, the root word has additional meaning.

BISINDO is a sign language that developed naturally through the deaf community in Indonesia. BISINDO does not follow Indonesian grammar and is commonly used in conversation. BISINDO prioritizes the meaning of the gestures carried out rather than the language structure of the gestures.

Unfortunately, not many people master sign language to communicate with the Deaf. This research proposes to bridge the communication gap between the deaf and others by building a mobile application to translate Indonesian text to 3D SIBI gesture animation.

SIBI differs from other sign languages such as American Sign Language (ASL) and British Sign Language (BSL) in terms of their gestures and method of arranging gestures in a sentence. Gestures in SIBI are arranged according to the rules in Indonesian grammar. Another difference lies in how the inflectional gesture is formed. Inflectional gestures are formed by combining the root word and affixes in the inflectional words [3].

Constructing a SIBI sentence gesture that differs from other sign languages needs different ways to generate 3D animation of a SIBI sentence gesture. This research faces several challenges. First, the Indonesian input sentence must be deconstructed into its components according to the SIBI rules to generate animated gestures. Fig. 1 shows an example of how an Indonesian sentence is deconstructed into word components: An inflectional word will be split into components affixes, and the root word ("mengatakan" = claim, will be separated into to prefix "me" + root word "kata" + suffix "kan"); Name will be changed to its alphabets ("William" will be split into w+i+l+l+i+a+m); numbers will be split into their essential number components ("6023" becomes "6"+"thousand"+"20"+3) [4].

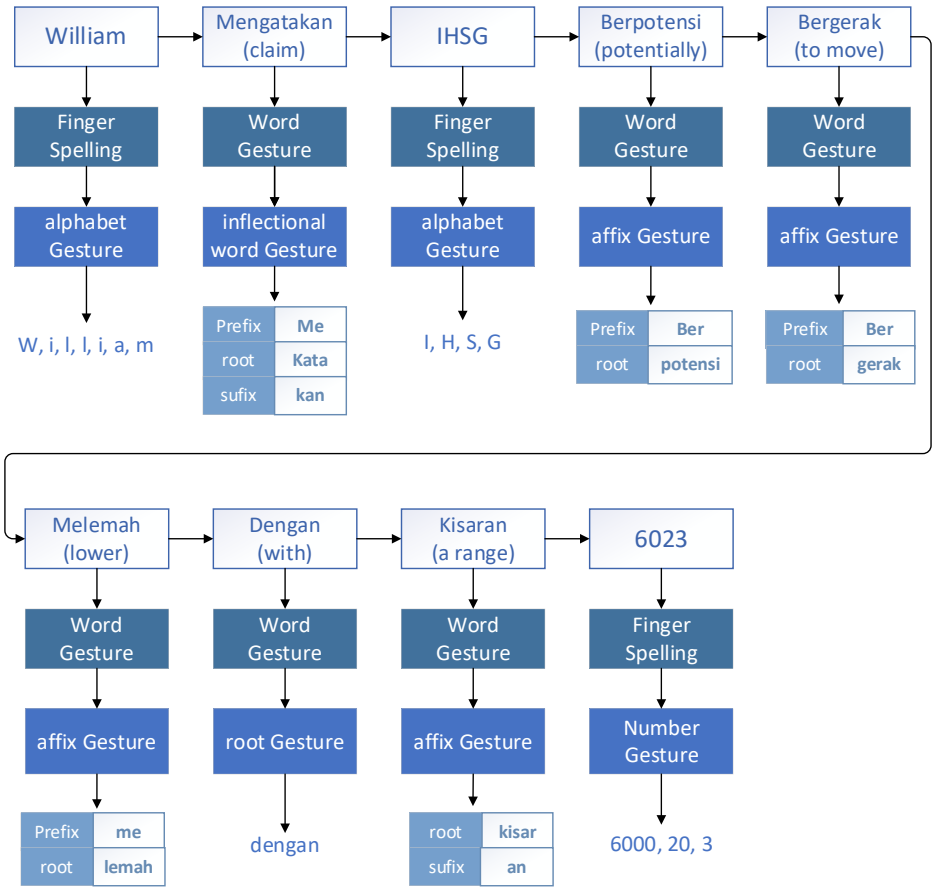


Fig. 1. Deconstruction of an Indonesian Input Sentence.

Second, to communicate in sign language, the signer will use his/her hands and fingers to demonstrate the word gesture, and at the same time, his/her mouth will pronounce the word being expressed. Therefore, building a 3D SIBI gesture animation has to be divided into two steps: building hand gestures and mouth movements. After the hand and mouth movement components are completed, two more challenges need to be solved by this research: how to make hand movements synchronize with mouth movements and how to connect the word components in a sentence into a single, smooth movement.

In conclusion, this paper discusses how to solve the four challenges in building an application to translate Indonesian input text into a 3D animation of the SIBI gesture.

The remainder of this paper is organized as follows: Section 2 states other research on generating text-to-gesture translation systems and mouth movement; Section 3 explains the proposed method for Indonesian text-to-3D SIBI gesture animation, dataset, and evaluation metric; Section 4 evaluates and analyzes the evaluation results; Section 5 closes this paper with the conclusion and future works.

II. Related Works

This section discusses other research on generating text-to-gesture translation systems and mouth movement. Table I shows some research on gesture generation, while Table II shows the research on mouth movements.

Table I shows that a common way to generate gestures from text is to create a sign language script and then generate gestures based on the script. Sign language scripts commonly used are Sutton SignWritting and HamNoSys notation. Sutton SignWritting notation is sign language that transcribes the signed gestures spatially, in two-dimensional canvas, as they are visually perceived. Furthermore, HamNoSys notation is an alphabetic system that describes a sign, primarily phonetic. HamNoSys notation is designed as a markup language foundation used to transcribe all sign languages worldwide. It does not depend on the conventions of each country, such as gestures for spelling the finger alphabet [5][6][7]. The lack of documentation of the HamNoSys and Sutton SignWritting corpus available for sign language in Indonesia causes the HamNoSys, and Sutton SignWritting cannot be implemented in SIBI's text-to-gesture translation system. SIBI is a sign system that follows Indonesian grammar [8]. The similarity between the SIBI structure and the Indonesian language structure is an advantage that can be used in SIBI's text-to-gesture translation system [4]. This research proposed an Indonesian Language stemming method to find word components in SIBI sentences. Table I also shows that other research focuses on building web-based text-to-gesture translation systems. Meanwhile, this research focuses on developing a text-to-gesture translation system as an Android Mobile Application.

TABLE I. RESEARCH ON GENERATING GESTURES IN SEVERAL COUNTRIES

Author	Sign Language	Platform	Method
(Karpouzis et al., 2007) [5]	Greek Sign language	Web-Based	1. Scripting Technology for Embodied Personalanguage (STEP) 2. HamNoSys 3. 3D Animation
(Bouzid & Jemni, 2014) [28]	Tunisian Sign Language	Web-Based	1. SWML (SignWriting Markup Language) 2. Sutton SignWriting notation 3. 3D Animation
(Boulares & Jemni, 2012) [6]	American Sign Language	Web-Based and Android	1. XML based 2. HamNoSys 3. Video Animation
(Efthimiou et al., 2009)[7]	Dicta-Sign: Greek, British, German, and French Sign Language	Web-Based	1. Signing Gesture Markup Language (SiGML) 2. HamNoSys 3. 3D Animation

Table II shows the research on mouth movements. Mouth movement research usually uses the form of viseme, which is a visual form of pronunciation. Three approaches to generating Viseme derived automatically from pronunciation are key-frame interpolation, model-based, and concatenative [9]. The Key-frame interpolation connects through interpolation the pre-define lip shape of all viseme that appear in a word or sentence [10], [11]. The model-based approach creates viseme from each pronunciation done by the human model [12]. The concatenative approach is a combination of key-frame and model-based approaches. Research by [13] and [14] tracks viseme on the human face to create a database of viseme animations. To generate mouth movement animation, all the viseme that appear in a word or sentence will be taken from the database and then connected through interpolation. The concatenative approach creates a realistic speech animation because it uses actual human face data as a model. Therefore, this research uses the concatenative approach for SIBI mouth movement animation.

Research related to text-to-gesture translation systems generally focuses on generating hand movements only. So far, no system has been found to generate hand movements and mouth movements from text input. This research proposes a combined SIBI hand and mouth movements from Indonesian text.

TABLE II. THE GENERATION OF MOUTH MOVEMENTS RESEARCH

Author	Language	Input	Method
(Setyati et al., 2017) [10]	Indonesian	Text	Hidden Markov Model, 2D Animation, key-frame interpolation
(Haryanto & Sumpeno, 2018) [11]	Indonesian	Text	Morphing Viseme, Syllable Concatenation, FACS, Key-frame Interpolation
(Yu & Wang, 2015) [12]	Mandarin Chinese	Video, Voice, dan Text	AAM, RBF Interpolation,Model-Based
(Ni & Liu, 2019) [13]	Chinese	Voice	DFFD, Concatenative
(Taylor et al., 2017) [14]	English	Voice	AAM, Concatenative

Note: FACS = Facial Action Coding System, AAM = Active Appearance Model, RBF = Radial Basis Function, DFFD = Dirichlet Free-Form Deformation

III. PROPOSED METHOD

This section discusses the proposed method to generate 3D animation from Indonesian sentence text. The discussion is divided into six sub-sections: overall system design, how to do SIBI sentence deconstruction, how to make 3D animation for hand and mouth movements, how to synchronize hand and mouth movements, and how to connect each word component in sentences by inserting transitional movements, and evaluations carried out to measure system performance.

A. Application Overview

Fig. 2 is the architecture used to build the Indonesian text to SIBI's 3D animation. The application consists of two main modules: text parser and animation engine. Text parser consists of two types: deconstructing Indonesian sentences into word components using hand gesture text parser and deconstructing Indonesian sentences into syllables using mouth movement text parser.

The other process is to generate movement based on the text parser. This process occurs in the animation engine module, resulting a 3D animation. This module is divided into two, namely, hand gestures and mouth movement animation engines. The hand gesture animation engine develops 3D animation of hand and finger movements based on data obtained from sensors placed on the body of the SIBI expert. Meanwhile, the mouth movement animation engine develops mouth movements using the facial data of the SIBI expert model. The two-generation processes will run in parallel. The synchronization process equalizes the speed between the hand gesture and mouth animation movements.

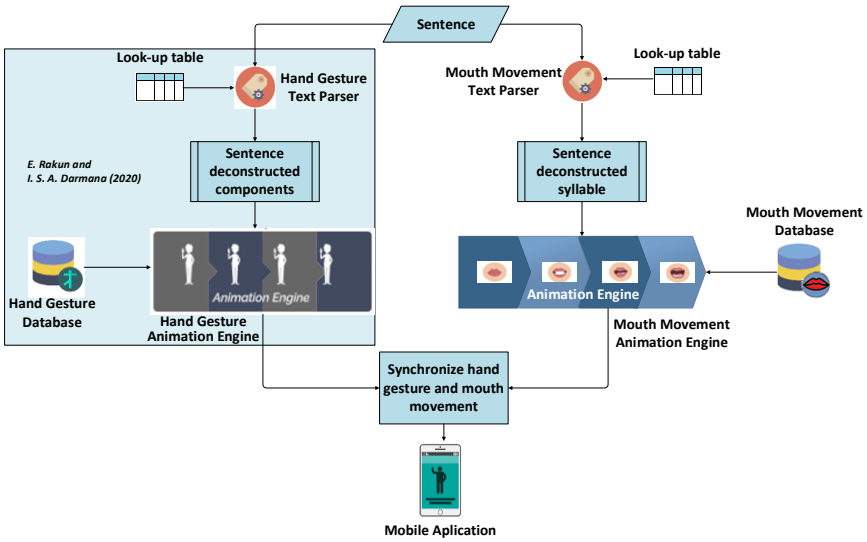


Fig. 2. SIBI 3D Animation Application Architecture.

The previous study (Rakun & Darmana 2020) [15] discusses deconstructing Indonesian sentences into word components and the hand gesture animation engine, so these two topics are discussed briefly here. This paper will discuss the mouth movement text parser, the mouth movement animation engine, and how to synchronize hand and mouth animations.

B. SIBI Sentence Deconstruction

This section discusses breaking down Indonesian sentences into the components needed to generate 3D animation for hand movements (1) and mouth movements (2). The text parsing result of both hand and mouth in (3).

1) Hand gesture text parser: SIBI uses the standard Indonesian grammar and has two types of gestures: word and finger gestures (Fig. 3). The word components are obtained with the help of a look-up table consisting of all inflectional words available in the SIBI dictionary. The look-up table contains each inflectional word's affix and root word components [4]. In addition to the look-up table consisting of inflectional words, there is also a look-up table consisting of slang words, which will later be used to correct the input word.

Fig. 4 shows the implementation of the hand gesture text parser [15]. In-text parsing splits sentences into word components. This process starts with the text tokenizer, splitting sentences based on spaces between words. Next is miss-spelling correction using slang word table look-up. This process corrects the slang words into root words. Furthermore, after the sentence is split into word components and corrected, the Word Mapper process will use the look-up table created previously based on the SIBI dictionary to split any inflectional word into its affix and root word.

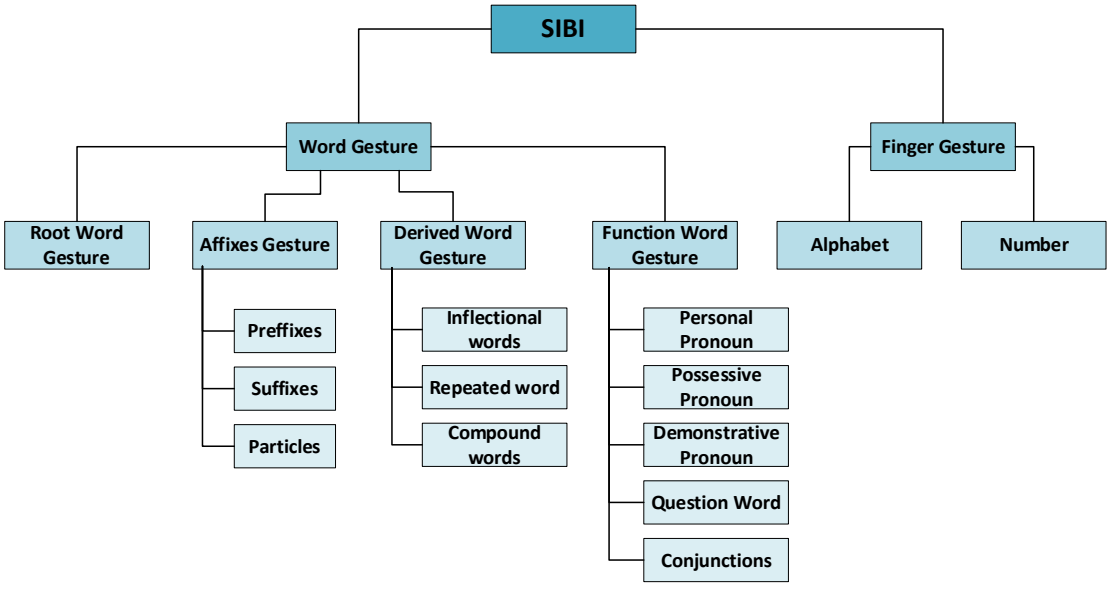


Fig. 3. Gestures in SIBI.

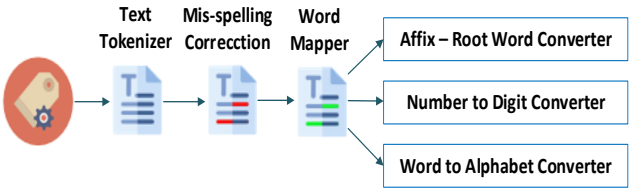


Fig. 4. Hand Gesture Text Parser Module.

The output of the hand gesture text parser, for example, is the splitting of the inflectional word "mengatakan" (= to say/ to tell) into prefix "me" + root word "kata" + suffix "kan." Meanwhile, the names and numbers in the sentence will be split into finger gestures. Names will be split according to the alphabet in the name; for example, William will be split into "w," "i," "l," "l," "i," "a," and "m." At the same time, the numbers will be split into the essential components of numbers, such as ten, hundred, thousand, and million: for example number "6203" will be split into "6" + "ribu" (= thousand) + "2" + "ratus" (= hundred) + "3" as seen in Fig. 4 [15].

2) Mouth text parser: The mouth text parser is a module used to break a sentence input into syllables. There are four steps to breaking the input sentence into syllables:

- Removing symbols and punctuation marks from the input sentence.
- Identifying the words and numbers in the sentence.
- Changing them to lowercase.
- Breaking the word into syllables according to the syllable look-up table (Table III below shows part of the syllable look-up table).

TABLE III. LIST OF INDONESIAN SYLLABLES AND THEIR EXAMPLES

Syllable	Example
V	**a**-tau, **i**-kan, **u**-ang, **e**-lang
CV	**ba-ca, du-ka, ko-ta**
VC	**an**-da, **il**-mu, ku-**il, in**-dah
CVC	**tam-bah, sam-bal, tum-pah**
CCV	**pri**-a, pu-**tra** pu-**tri, tri**-o
VCC	**eks**-tra, bu-**ang**
CVCC	**teks,**
CCVC	**Stig**-ma
CCCV	In-**stru**-men
CCCVC	**Struk**-tur
CCVCC	Kom-**pleks,** ke-**nyang,** me-**nyang**-kut

Note: C = consonant, V = vowel

Forming the mouth movements will use as many syllables as the syllables in the input sentence.

A syllable is a part of a word articulated in one breath. A syllable consists of one or more phonemes combined. Each syllable always contains a vowel phoneme [16][17]. Table III shows some examples of syllables. The actual syllable look-up table consists of all syllables in the SIBI dictionary.

In the previous study (Muzahidin and Rakun, 2020) [18], the respondents gave suggestions and criticisms of the lack of tongue movement. The tongue is essential in pronouncing a word [19]. Therefore, this study improves the mouth movement animation by adding tongue movement. With the addition of tongue movements, respondents can better distinguish words with similar lip movements. Tongue movements are formed separately from the formation of lip movements. The lip movement is formed based on videos of SIBI experts pronouncing words. On the other hand, the tongue movement data was formed according to the rules in the Bina Talk book, as shown in Table IV [20].

TABLE IV. TONGUE MOVEMENT

No	Name	Description	Example
1	Apiko dental	The tip of the tongue at the base of the upper teeth touches the front alveolar (upper gum).	/t/, /d/, /n/
2	Apiko alveolar	The tip of the tongue meets the arch of the tooth (alveolar)	/s/ and /z/
3	Dorso velar	Attach the back of the tongue to the area (soft palate)	/ng/, /g/, /k/, and /x/
4	Fronto platal	The center of the tongue is the articulator, and the palate is the articulation	/j/, /c/, and /y/
5	Lateral	Lifting the tongue to the palate	/l/
6	Vibration	Attaching the tongue to the alveolar (gums) and so on repeatedly	/r/

3) Module text parser results: The hand gesture and mouth movement text parser will produce different sentence fragments. Table V shows examples of output from hand gestures and mouth movement text parsers. Next, the output of each text parser will be used to generate hand gesture and mouth movement animations.

TABLE V. TEXT PARSER RESULT

Word	Hand Gesture	Mouth Movement	Represent
Saya	Saya	Sa, ya	Root Word
Surya	S, u, r, y, a	S, u, r, y, a	Fingerspelling (name)
45	Empat, puluh, lima	Em, pat, pu, luh, li, ma	Fingerspelling (number)
Abu-abu	Abu-abu	A, bu, a, bu	Repeated Word
Memakai	Me-, pakai	Me, pa, kai	Prefix + root word
Pakaian	Pakai, -an	Pa, kai, an	Root word + suffix
Memakaikan	Me-, pakai, -kan	Me, pa, kai, an	Prefix + root word + suffix

C. 3D Animation Generation

This section discusses how to generate hand gesture animation based on a hand gesture text parser (1) and how to generate mouth movement animation based on a mouth movement text parser (2).

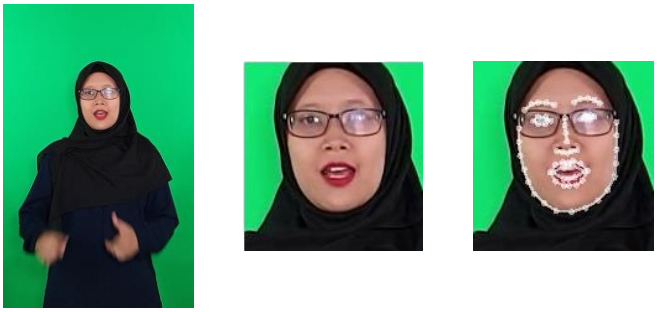
1) Hand gesture animation engine: Hand gesture data creation begins with hand movement data collection using

sensors from Perception Neuron v2. This study used 25 sensors: 3 upper-body, 1 left-shoulder, 3 left-hand, 7 right-fingers, 1 right-shoulder, 3 right-hand, 7 right-fingers. These sensors attached to a SIBI expert model will record data in coordinates of the position and rotation of the human body joints. All 3,100 words available in the SIBI dictionary were recorded. The recorded sensor data are then stored as skeletal animation. The data generated by the sensor is not perfect. Fig. 5 shows the difference between the hand movements made by the SIBI expert (right image) and the hand gestures generated by the sensor (left image). The AxisNeuron application is used to check and correct every skeletal data generated. In this process, the skeletal animation clip needs to be readjusted with references to the SIBI dictionary. The corrected skeletal animation clips were exported in fbx format and are used by Unity3D to generate hand gesture animation.

Fig. 5. Data Sensor Recording Result.

2) Mouth animation engine: The mouth animation movement was created based on SIBI expert facial data detected using the OpenPose[1][2] library. Then, the Dirichlet Free-Form Deformation (DFFD) uses the coordinate data from the face detection to build mouth movements. The data used in this process is data for mouth movements generation only. In addition, it is also necessary to develop tongue movements based on the Bina Bicara book.

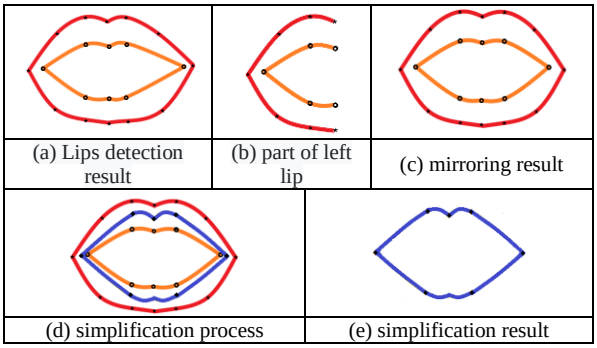
a) Mouth Movement: The process of generating mouth movements starts by recording the SIBI expert pronouncing every word from the SIBI dictionary. Using actual video data can produce more realistic lip movements [21]. The steps to obtain the face coordinates to drive the three-dimensional mouth movement animation are as follows (as shown in Fig. 6 and Fig. 7):



(a) Data Video (b) Face Detection (c) Coordinate Detection

Fig. 6. Obtaining Face Coordinate Steps.

[1]https://github.com/CMU-Perceptual-Computing-Lab/openpose



(a) Lips detection result (b) part of left lip (c) mirroring result

(d) simplification process (e) simplification result

★/ red line = outer coordinates (po); ✪/ orange line = inner coordinates (pi); ◆/ Blue line = middle coordinates (pm)

Fig. 7. Simplification Lip Point.

- Take sign language gesture with its pronunciation video performed by a SIBI expert [Fig. 6(a)].
- Crop the video to focus on the face only [Fig. 6(b)].
- Implement the coordinate detection process on the face using OpenPose [Fig. 6(c)].
- Get coordinates lip point [Fig. 7(a)].
- Cut the lips in half, and take a left part [Fig. 7(b)].
- Discard the right half of the lip and replace it with the mirror of the left lip to form symmetrical lips [Fig. 7(c)].
- Do the lips simplification process by averaging the outer and inner lips [Fig. 7(d)].
- The result of the lips simplification will be used for mouth movement animation [Fig. 7(e)].

The mirroring process is implemented because the shape of the human lips is not symmetrical between the left and right parts, caused by various factors such as teeth, cheeks, and face shape. The lip coordinates are then identified as the outer coordinates(po) and the inner coordinates (pi). Then, find the middle coordinates (pm) using equation 1 [22]. Fig. 7(c) shows the results of the lip point simplification. These lip simplification coordinates will generate lip movement animation [18].

$$pm_x = \frac{(po_x + pi_x)}{2} \quad (1)$$

b) Dirichlet Free-Form Deformation (DFFD): In this study, the Dirichlet Free-Form Deformation (DFFD) method is used to deform the 3D model. The application of DFFD in this research is by the following process:

- Apply the point of mouth movement form [Fig. 7(e)] into the 3D animation.
- Using the DFFD method to make mouth movements.
- Do this process for all the syllables in the SIBI dictionary
- Save this mouth movement into the database

The DFFD process changes the coordinates of a control point in an area that affects changes around another control point. Fig. 8 is the result of the deformation of the DFFD where the change in coordinates at one point has a local deformation area that reaches its surrounding points. All coordinate points move each other towards the specified coordinates. The results of these coordinates create changes in the shape of the lips according to the input data.

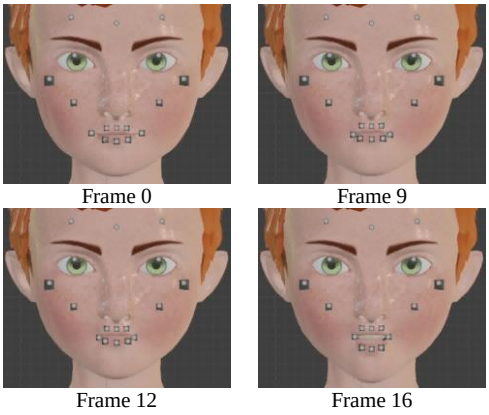


Fig. 8. 3D Animation Mouth Movement Syllable "Bu".

Next is transferring the lip movement data into a 3D animation. Each lip part that will be moved has to be defined and coded for each syllable. All animation clips of Indonesian viseme syllables will be available after doing this process for all syllables. These visemes syllables animation will be stored as a database that the SIBI application can use.

The determining factor in understanding the word pronounced depends not only on the movement of the lips but also on teeth, tongue, and expression. Until now, no research has proven that people can catch a syllable or a word by looking at lip movements alone. Providing the 3D models with teeth and the tongue movement corresponding to each syllable will make it easier for the deaf to catch each syllable spoken.

D. Synchronizing Hand Gestures and Mouth Movement

Hand gestures and mouth movements are called based on the input sentence. The input sentence is deconstructed into words and syllables to generate hand gestures and mouth movements. The hand gestures and mouth movements will be generated simultaneously, but they do not take the same amount of time. So to synchronize hand gestures and mouth movements, it is necessary to accelerate or decelerate the movement of hand gestures. The speed of hand movement will follow the speed of the mouth movements. Generally, humans need 1 to 3 seconds to pronounce a word (depending on the length of the spoken word). Usually, words with only two syllables will be pronounced in one second. The gestures in the hands will follow the speed of the word's pronunciation. A one-second (two syllables) video is converted into frames, which is 30 frames per second or equal to 15 frames per syllable. So the length of the hand gesture is 15 frames multiplied by the number of syllables of the word input. The algorithm for synchronizing hand gestures and mouth movements can be seen in Algorithm 1 below. The 3D animation will be displayed in a full model with synchronized pronunciation and hand gestures.

Algorithm 1 Synchronizing Hand and Mouth Movement

```
program start
initialize variable word_input
initialize variable mouth
initialize variable handsign
initialize variable frame = 15
start
    call function splittosyllable with word_input
    splittosyllable return value mouth
    output  mouth

    handsign = mouth*frame
    output handsign

    call function runanimationmouth withinput mouth
    call function runanimationhandsign withinput
handsign
end
```

E. Insertion of Transition Movement using Cross Fade

This section explains how to create animated hand gestures. Hand gesture data creation begins with collecting hand and fingers movement coordinates for each word in the SIBI dictionary using a sensor, Perception Neuron v2. The coordinates of the hand and fingers are stored in a database. Creating hand gestures from an input sentence is done by retrieving the hand and fingers coordinates of each word in the sentence from the gesture database. Next is to insert transition gestures between words to create smooth, unified 3D animation sentence gestures. Fig. 9 shows the position of transition gestures in a sentence. This research uses the cross-fade method implemented using the Animancer API [15][18] to create transition gestures.

This research uses interpolation to generate a smoother transition movement between word gestures. Linear interpolation, also known as LERP, is the method to interpolate the positions and rotation values from the last frame of a word animation to the first frame of the following word animation linearly. It has the advantages of easy implementation and short execution time as the animation method traditionally used in animation [23]. Linear interpolation is a parametric curve defined as a straight line function can be seen in the following equation 2:

$$Q(u) = P_0 + u(P_1 - P_0) \quad (2)$$

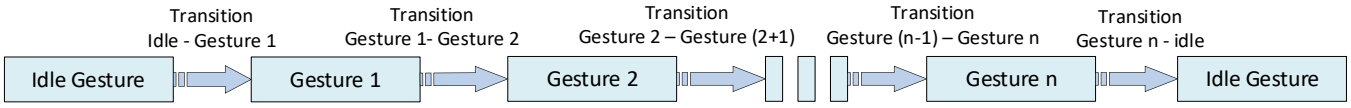


Fig. 9. Transition Motion between Word Gestures.

Equation (2) can also be written as:

$$Q(u) = (1-u)P_0 + uP_1 \quad (3)$$

The value of *u* is used to set the interpolation to be built. If the value of u is 0, then *Q(u)* will be equal to the starting point of P_0, whereas if the value of u is 1, then *Q(u)* will be equal to the endpoint of P_1. If the value of *u* is between 0 and 1, then it will produce a point on the line $\overline{P_0P_1}$. Interpolation occurs if the value of *u* is in the interval [0,1]. If the value of *u* is outside the interval, it will not be interpolated.

Cross-fading is used to combine two LERP-based animated clips [24]. This technique can generate a smooth transition between animation clips. Cross-fade works by stacking the timeline of two animated clips, as seen in Fig. 10. The merging process requires a blend percentage β of the clips to be merged. β starts at 0 at time t_{start}. The meeting time between clip A and clip B is when a cross-fade process occurs. The value of β is then slowly incremented to 1 until time t_{end}. At that time, only the animation of clip B will appear. The time interval when the cross-fade is in progress is called the *blend time* ($\Delta t_{blend} = t_{end} - t_{start}$).

F. Evaluation Metric

There are three tests carried out to measure the performance of the system being built. The first test is intended to measure the mouth movement animation when using syllables. The second test is carried out to measure the usability of this system. The third test is to measure the execution time.

1) Evaluation of mouth movement: Evaluation of mouth movement is done by calculating the Mean Opinion Score (MOS). Four online questionnaires (done during the Covid-19 pandemic locked down) need to be filled in by the respondents to check how well the 3D animation works: In Questionnaire 1, the animation pronounces 40 Indonesian words available in the SIBI dictionary; In Questionnaire 2, the animation pronounces 25 sentences long sentences; Questionnaire 3 compares the 3D animation with the original videos; In questionnaire 4, combine the mouth movement with hand gestures simultaneously. MOS respondents consisted of three deaf students and three SIBI teachers from the School for special needs SLB Santi Rama.

The evaluation uses subjective values from the teachers and deaf students because there is still no ground truth to test the correctness of lip movements. This application is intended for the Deaf, so it requires a direct assessment of the intended target. The MOS assessment uses a scale from 1 to 5 [2]. Calculations using MOS can be seen in the following equation 4:

$$MOS = \sum_{i=1} \frac{x(i).k}{N} \quad (4)$$

Where:

$x(i)$ = sample number i

k = weight value

N = Number of Respondents

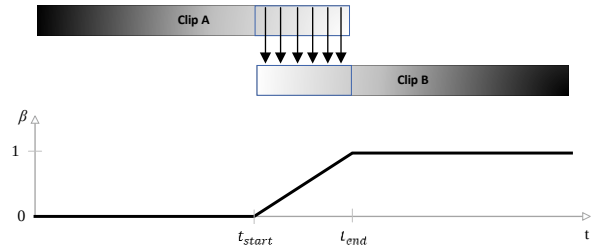


Fig. 10. Smooth Transition on Cross-Fading.

2) Application survey measurement: The System Usability Scale (SUS) test, a qualitative research tool, is used to assess and improve the usability of this system. Usability testing is carried out using a usability testing sheet containing tasks and scenarios the respondent must do during the test. In each task and scenario, the respondent will be assessed on whether he/she has succeeded in carrying out the task, the ease and difficulty the respondent faced, things the respondent likes and dislikes, and any suggestions from the respondent [25]. In each question, respondents will be asked to determine their rating on a scale of 1-5 based on their experience after using an interactive system design. A low score indicates disagreement from the respondent, while a high score indicates the respondent's agreement with the questions. SUS Score Assessment using the scoring formula [26]. Table VI is a list of questions of SUS taken from statements from [26] that have been translated into Indonesian [27]. The SUS itself consists of 10 items, the odd numbers are for positive items and the even numbers for negative. For positive items, the score contribution is the scale position minus 1 and for the negative items, the score contribution is 5 minus the scale position. The overall SUS score is the result of the sum of item score contributions multiply by 2.5, range from 0 to 100.

TABLE VI. TEXT PARSER RESULT

No	Question (Indonesian)	Question (English)
1.	Saya berfikir akan menggunakan sistem ini lagi	I think that I will use this system again
2.	Saya merasa sistem ini rumit untuk digunakan	I found the system to be unnecessarily complex
3.	Saya merasa sistem ini mudah untuk digunakan	I found the system easy to use
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini	I think that I would need the support of a technical person to be able to use this system
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya	I found the various features in this system were well integrated
6.	Saya merasa ada banyak hal yang tidak konsisten(tidak serasi) pada sistem ini	I thought there was too much inconsistency in this system
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat	I feel that most people would learn to use this system very quickly
8.	Saya merasa sistem ini membingungkan	I found the system very cumbersome to use
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	I found no obstacles in the usage of this system
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini	I needed to learn a lot of things before I could get going with this system

IV. EXPERIMENT RESULTS

A. Mouth Movement Development

This research used deconstruction of words into syllables to generate mouth movements. Each syllable is developed based on lip movements and stored in a database. The lip movement of 3D animation starts from a silent mouth position labeled "idle." Then the lip movements are generated sequentially according to the word's syllables. At the end of the movement of the syllable, the mouth returns to the "idle" position. Fig. 11 shows an example of mouth movement generation. Furthermore, a transition motion that connects one syllable to the next using the cross-fading technique was added. This cross-fading technique is the same technique when generating transitions in hand movements.

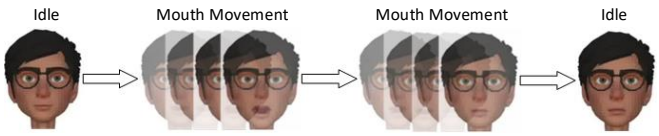


Fig. 11. Mouth Movement Generation.

B. Text-to-Gesture Application System Analysis

The text-to-gesture application system analysis measures the mean opinion score (MOS) on four questionnaires, System Usability Scale (SUS), and execution time. The MOS testing is used to test how well the mouth movement is in pronouncing words. Usability testing is a test to assess the user interface of the text-to-gesture application. Furthermore, execution time tests the time it takes to run the text-to-gesture application.

1) Mean opinion score result: The qualitative testing to measure the performance of the mouth movements generation on syllables was done by distributing four questionnaires to three SIBI teachers and three students from the School for special needs students, SLB Santi Rama. Questionnaire 1, that test the mouth movements animation in word pronunciation, yields a score of 4.025. Questionnaire 2, which tests the mouth movements animation to pronounce a sentence, got a score of 4.025. Then, questionnaire 3, to test user understanding of the animation when hand gestures and mouth movements were combined, got a score of 4.422. Finally, questionnaire 4 tested the similarity between the animation and the original video and got a score of 4.282.

From the MOS results, respondents found it easier to catch words spoken solely (4.025) than in sentences (3.96). The combination of hand gesture and mouth movement animation can improve animation realism and understanding of the gestures demonstrated in 3D animation (3.96 vs. 4.422).

2) System usability scale (SUS): Google Form application is used to help design the questionnaire for SUS. The respondents will answer ten questions by choosing a scale from 1 – 5 for each item. This questionnaire was distributed for seven days and obtained 72 respondents. These respondents consisted of various backgrounds (sign language teachers, deaf people, and ordinary people), ranging from 19 to 61 years, and comprised 28 women and 44 men. The results of the SUS test obtained a score of 76.25 which means that it is categorized as GOOD and considered an application that users generally can accept [26].

3) Execution time: In building the Text-to-3D animation application, three processes involve the use of data, namely (1) sentence translation, (2) storage of application settings, and (3) dictionary search. In each of these processes, the execution time was tested 100 times. Furthermore, from the 100 results, a 95% confidence interval was calculated to find the actual execution time value interval. The 95% confidence interval results for each process successfully met the requirements to react instantaneously based on [23], which has an execution time of under 100 ms. The results of the execution time test can be seen in Table VII.

TABLE VII. EXECUTION TIME

Process	*Confidence interval* 95% execution time	Qualified react instantaneously (< 100 ms) based on research by Nielsen (1993)
Sentence translation	36.5 ± 2.65 ms	Yes
Storage of application settings	0.23 ± 0.039 ms	Yes
Dictionary search	2.21 ± 0.196 ms	Yes

V. CONCLUSION

This study aims to build a SIBI Text-to-3D animation translator application system. The output of this application is SIBI 3D gesture animation of every input word in an Indonesian sentence. The 3D animation consists of hand gestures and mouth movements' animation. Hand gesture data creation begins with hand movement data collection using sensors from Perception Neuron v2. The recorded sensor data are then stored as skeletal animation. The mouth animation movement was created based on SIBI expert facial data detected using the OpenPose library. Then, the Dirichlet Free-Form Deformation (DFFD) uses the coordinate data from face detection to build mouth movements. Hand gestures and mouth movements are called based on the input sentence. The input sentence is deconstructed into words to generate hand gestures and syllables to generate mouth movements.

It is necessary to accelerate or decelerate the movement of hand gestures to synchronize hand gestures and mouth movements. This research uses Cross-Fade interpolation to generate a smoother transition movement between word gestures. There are three tests carried out to measure the performance of the system being built. The first test is intended to measure the mouth movement animation when using syllables by calculating MOS. The second test is carried out to measure system's usability by using the SUS test. The third test measures the execution time by calculating the time needed by processes involving data. From the MOS results, respondents found it easier to catch words spoken solely (4.025) than in sentences (3.96). The combination of hand gestures and mouth movement animation can improve animation realism and understanding of the gestures demonstrated in 3D animation (3.96 vs. 4.422). The resulting animation is quite similar to the original video (4.282). SUS score is 76.25, which means this application is in the GOOD category. The execution time of all processes that involved data (sentence translation, storage of application settings, and dictionary search) are less than 100ms, which means it met the application requirements to react instantaneously. Hopefully, this application can be used to solve the communication problems between the deaf and the people around them. Because the number of words available in the SIBI dictionary (around 3100 words) is far less than the Indonesian words, some words need to be fingerspelled or replaced by similar words available in the SIBI dictionary. In the future, a synonym table can be added to this application. The synonym table will speed up the process of replacing a word with its synonym from the SIBI dictionary. This synonym table must be updated regularly to cover as many Indonesian words as possible. Another thing that can be done to improve this application is to add facial expressions to the 3D model. In sign language, facial expressions are used to strengthen the meaning of a sentence, just like intonation in spoken language.

ACKNOWLEDGMENT

This work is supported by The National Research and Innovation Agency (BRIN) Implementation Grant, Number PKS-175/UN2.INV/HKP.05/2021. This support is gratefully received and acknowledged.

REFERENCES

[1] V. Khetani, Y. Gandhi, and R. R. Patil, "A Study on Different Sign Language Recognition Techniques," in 2021 International Conference on Computing, Communication and Green Engineering (CCGE), 2021, pp. 1–4.

[2] S. Sumpeno, M. Hariadi, and A. M. Syarif, "Development of Indonesian Text-to-Audiovisual Synthesis System Using Syllable Concatenation Approach to Support Indonesian Learning," pp. 166–184, 2017.

[3] K. Anggraini, E. Rakun, and L. Y. Stefanus, "Recognizing The Components of Inflectional Word Gestures in Indonesian Sign System known as SIBI (Sistem Isyarat Bahasa Indonesia) by using Lip Motion," in 2019 International Conference on Electrical Engineering and Informatics (ICEEI), 2019, pp. 384–389.

[4] M. Adriani, J. Asian, B. Nazief, S. M. M. Tahaghoghi, and H. Williams, "Stemming Indonesian: A Confix-Stripping Approach.," ACM Trans. Asian Lang. Inf. Process., vol. 6, 2007.

[5] K. Karpouzis, G. Caridakis, S.-E. Fotinea, and E. Efthimiou, "Educational Resources and Implementation of a Greek Sign Language Synthesis Architecture," Comput. Educ., vol. 49, no. 1, pp. 54–74, Aug. 2007.

[6] M. Boulares and M. Jemni, "Mobile Sign Language Translation System For Deaf Community 1–4.," 2012.

[7] E. Efthimiou et al., "Sign Language Recognition, Generation, and Modelling: A Research Effort with Applications in Deaf Communication," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5614 LNCS, no. PART 1, pp. 21–30, 2009.

[8] S. Siswomartono, "Cara Mudah Belajar SIBI (Sistem Isyarat Bahasa Indonesia)," 2007.

[9] A. Thangthai, B. Milner, and S. Taylor, "Synthesising Visual Speech Using Dynamic Visemes and Deep Learning Architectures," Comput. Speech Lang., vol. 55, pp. 101–119, 2019.

[10] E. Setyati, O. Susandono, L. Zaman, Y. M. Pranoto, S. Sumpeno, and M. H. Purnomo, "Establishment of Indonesian Viseme Sequences Using Hidden Markov Model Based on Affection," 2017 Int. Semin. Intell. Technol. Its Appl. Strength. Link Between Univ. Res. Ind. to Support ASEAN Energy Sect. ISITIA 2017 - Proceeding, vol. 2017-Janua, pp. 275–280, 2017.

[11] H. Haryanto and S. Sumpeno, "A Realistic Visual Speech Synthesis for Indonesian Using a Combination of Morphing Viseme and Syllable Concatenation Approach to Support Pronunciation Learning," vol. 13, no. 8, pp. 19–37, 2018.

[12] J. Yu and Z. F. Wang, "A Video, Text, and Speech-Driven Realistic 3-D Virtual Head for Human-Machine Interface," IEEE Trans. Cybern., vol. 45, no. 5, pp. 977–988, 2015.

[13] H. Ni and J. Liu, "3D Face Dynamic Expression Synthesis System Based on DFFD," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 1125–1128.

[14] S. Taylor et al., "A Deep Learning Approach for Generalized Speech Animation," ACM Trans. Graph., vol. 36, no. 4, 2017.

[15] E. Rakun and I. S. A. Darmana, "Generating of SIBI Animated Gestures from Indonesian Text," PervasiveHealth Pervasive Comput. Technol. Healthc., pp. 256–264, 2020.

[16] S. Suyanto, "Incorporating Syllabification Points into a Model of Grapheme-to-Phoneme Conversion," Int. J. Speech Technol., vol. 22, no. 2, pp. 459–470, 2019.

[17] S. Suyanto, "Flipping Onsets to Enhance Syllabification," Int. J. Speech Technol., vol. 22, no. 4, pp. 1031–1038, 2019.

[18] S. Muzahidin and E. Rakun, "Text-Driven Talking Head Using Dynamic Viseme and DFFD for SIBI," in 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 2020, pp. 173–178.

[19] L. Zhao and L. Czap, "Visemes of Chinese Shaanxi xi'an Dialect Talking Head," Acta Polytech. Hungarica, vol. 16, no. 5, pp. 173–193, 2019.

[20] E. Sadja'ah, Bina Bicara, Persepsi Bunyi dan Irama. Bandung: PT. Refika Aditama, 2013.

[21] I. R. Ali, H. Kolivand, and M. H. Alkawaz, "Lip Syncing Method for Realistic Expressive 3D Face Model," Multimed. Tools Appl., vol. 77, no. 5, pp. 5323–5366, 2018.

[22] M. Liyanthy, H. Nugroho, and W. Maharani, "Realistic Facial Animation of Speech Synchronization for Indonesian Language," 2015 3rd Int. Conf. Inf. Commun. Technol. ICoICT 2015, pp. 563–567, 2015.

[23] S. Lim, "Linear Interpolation Transition of Character Animation for Immediate 3D Response to User Motion," Int. J. Contents, vol. 11, no. 1, pp. 15–20, 2015.

[24] J. Gregory, Game Engine Architecture, 3rd ed. CRC Press, 2019.

[25] A. K. Darmawan, M. A. Hamzah, B. Bakir, M. Walid, A. Anwari, and I. Santosa, "Exploring Usability Dimension of Smart Regency Service with Indonesian Adaptation of The System Usability Scale (SUS) and User Experience Questionnaire (UEQ)," in 2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), 2021, pp. 74–79.

[26] Nielsen, Usability Engineering. San Diego, 1993.

[27] Z. Sharfina and H. B. Santoso, "An Indonesian adaptation of the System Usability Scale (SUS)," in 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2016, pp. 145–148.

[28] Y. Bouzid and M. Jemni, "TuniSigner: A Virtual Interpreter to Learn sign Writing," Proc. - IEEE 14th Int. Conf. Adv. Learn. Technol. ICALT 2014, pp. 601–605, 2014.