Pemodelan Lip Reading Bahasa Indonesia Berbasis Visem Menggunakan VGG16 serta Jaro-Winkler Similarity dan Bigram

Henry Wicaksono, Liliana, Alvin Nathaniel Tjondrowiguno
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
E-mail: henrywicaksono1205@gmail.com, lilian@petra.ac.id, alvin.nathaniel@petra.ac.id

ABSTRAK

Lip reading merupakan teknik mengenali kata-kata hanya melalui representasi visual gerakan bibir. *Lip reading* memiliki banyak kegunaan, seperti untuk alat bantu bagi pasien laryngectomy dan alat bantu untuk penderita disabilitas pendengaran. Sebuah riset menunjukkan bahwa 2.6% penduduk Indonesia mengalami disabilitas pendengaran. Dengan demikian, *lip reading* dapat menjadi solusi yang relevan di Indonesia. Penelitian ini bertujuan untuk melakukan pemodelan sistem *lip reading* bahasa Indonesia berbasis visem.

Metode yang digunakan dalam penelitian ini adalah VGG16 yang digunakan sebagai *classifier* serta Jaro-Winkler similarity dan bigram (JW-bigram) yang digunakan sebagai *decoder*. Dataset yang digunakan terdiri atas 25 kalimat bahasa Indonesia yang tersusun atas 50 kata berbeda dan diucapkan oleh 12 orang pembicara. Hasil penelitian menunjukkan bahwa *sistem lip reading* yang dibuat dengan menggunakan VGG16 dan JW-bigram lebih efektif secara akurasi mau pun kecepatan dibandingkan dengan kombinasi metode lainnya.

Kata Kunci: *lip reading*, pemrosesan video, VGG16, Jaro-Winkler *similarity*, bigram

ABSTRACT

Lip reading is a technique used to understand spoken words through visual representation of lip movements. Lip reading has many uses, such as aids for laryngectomy patients and aids for people with hearing disabilities. A research shows that 2.6% of Indonesia's population has a hearing disability. Thus, lip reading can be a relevant solution in Indonesia. This study aims to model a viseme-based Indonesian lip reading system.

The method used in this research is VGG16 which is used as a classifier and Jaro-Winkler similarity and bigram (JW-bigram) which is used as a decoder. The dataset used consists of 25 Indonesian sentences composed of 50 different words and spoken by 12 speakers. The results showed that the lip reading system made using VGG16 and JW-bigram was more effective in terms of accuracy and speed compared to other methods combinations.

Keywords: *lip reading, video processing,* VGG16, *Jaro-Winkler similarity, bigram*

1. PENDAHULUAN

Lip reading merupakan teknik mengenali kata-kata hanya melalui representasi visual gerakan bibir [20]. Komunikasi dengan *lip reading* memiliki banyak keuntungan, seperti tidak terpengaruh gangguan *noise* dari sumber lain serta lebih mudah dalam memahami kata-kata lawan bicara saat komunikasi melalui suara kurang efektif. Oleh karena itu, *lip reading* banyak digunakan untuk berbagai kebutuhan, seperti alat bantu bagi pasien dengan keterbatasan kemampuan berbicara seperti pasien *laryngectomy* [3], alat bantu bagi penderita disabilitas pendengaran [21], serta untuk *speech recognition* [7]. Penderita disabilitas pendengaran yang menggunakan alat bantu pendengaran dan memiliki kemampuan *lip reading* mencatatkan performa 93,5% lebih baik dibanding penderita yang tidak memiliki salah satu dari kedua faktor tersebut [6]. Riset yang dilakukan oleh Kementrian Kesehatan Republik Indonesia menunjukkan bahwa sekitar 2,6% penduduk Indonesia mengalami gangguan pendengaran [10]. Dengan demikian, *lip reading* dapat menjadi solusi yang relevan di Indonesia. Namun, sebagian besar sistem *lip reading* yang telah berkembang umumnya masih terbatas pada bahasa Inggris. Salah satu contoh aplikasi bahasa Inggris tersebut adalah SRAVI (*Speech Recognition App for the Voice Impaired*) buatan Liopa yang diperuntukkan bagi pasien yang kesulitan berbicara untuk berkomunikasi dengan tenaga medis. Penelitian *lip reading* di Indonesia masih terbilang sangat minim dibandingkan dengan bahasa lain seperti bahasa Inggris.

Penelitian *lip reading* sendiri pada umumnya menggunakan 2 macam metode: *classifiers* untuk mengklasifikasikan bentuk bibir dan *decoders* untuk mendapatkan hasil akhir. Metode yang banyak digunakan sebagai *classifiers* adalah *Convolutional Neural Network* (CNN) [4, 8, 17, 23], *Multi Layer Perceptron* (MLP) [21], *Support Vector Machine* (SVM) [3]. CNN menghasilkan akurasi terbaik dibanding *classifiers* lain. Namun, akurasi tinggi baru dicapai pada *lip reading* tingkat kata. Akurasi tinggi pada tingkat kalimat sulit dicapai akibat kemiripan bentuk antar obyek yang digunakan sebagai kelas prediksi pada *classifier*. Beberapa obyek yang telah digunakan adalah suku kata [14], huruf [4], dan *frame difference* [21]. Akurasi diharapkan dapat ditingkatkan dengan menggunakan obyek yang memiliki perbedaan karakteristik visual yang jelas seperti visem [25] sebagai kelas prediksi. Penelitian ini akan membandingkan *classifier* VGG16 [30] yang merupakan salah satu model CNN dengan performa terbaik dan *Spatiotemporal* CNN (STCNN) [11] yang merupakan model *state-of-the-art* dengan menggunakan visem sebagai obyek kelas prediksi.

Ada pun metode yang banyak digunakan sebagai *decoder* adalah *Recurrent Neural Network* (RNN) seperti *Long Short-Term Memory* (LSTM) [17] dan *Bidirectional Gated Recurrent Unit* (Bi-GRU) [4]. Kelemahan RNN adalah waktu *training* yang lama serta tidak mampu memproses *sequence* sebelum fitur ekstraksi selesai dilakukan [8], sehingga sistem cenderung lambat. Kecepatan dapat ditingkatkan dengan menggunakan *decoder* yang lebih sederhana dan memiliki performa yang sama baik atau lebih baik dari RNN. Hal ini diharapkan dapat dicapai dengan kombinasi Jaro-Winkler *similarity* [15] yang dapat menilai kemiripan antar *sequence* dan

bigram [24] yang dapat menilai hubungan antar kata. Selain tidak memerlukan proses *training*, keduanya jauh lebih sederhana daripada RNN sehingga diharapkan lebih unggul dalam hal kecepatan. Akurasi dan kecepatan Jaro-Winkler *similarity* dan bigram sebagai *decoder* akan dibandingkan dengan LSTM dan Bi-GRU dalam penelitian ini.

2. PENELITIAN SEBELUMNYA

Penelitian akan menggunakan beberapa penelitian lain terkait yang telah dilakukan sebelumnya sebagai tinjauan studi. Berikut adalah penelitian-penelitian yang telah dilakukan dalam *lip reading*:

2.1 *Sentence-Level Indonesian Lip Reading with Spatiotemporal CNN and Gated RNN*

Aulia, et al [4] memanfaatkan *spatiotemporal* CNN (STCNN) dan *Bidirectional* GRU (Bi-GRU) dalam *lip reading* bahasa Indonesia. STCNN digunakan untuk mengekstrak fitur dari video, sementara Bi-GRU digunakan untuk menangkap ketergantungan linguistik antar kata, yakni bagaimana suatu kata dipengaruhi oleh pengucapan kata sebelum dan sesudahnya. Penelitian menggunakan AVID dataset yang terdiri atas 52 kata bahasa Indonesia dengan 10 orang pembicara berbeda. Hasil penelitian menunjukkan *Character Error Rate* (CER) sebesar 0.049 dan *Word Error Rate* (WER) sebesar 0.133. Kelemahan penelitian ini adalah dalam hal kecepatan Bi-GRU yang masih dapat ditingkatkan serta akurasi yang belum terlalu tinggi pada *lip reading* tingkat kalimat.

2.2 *Lip Reading using CNN and LSTM*

Garg, et al [8] membandingkan beberapa model CNN dalam *lip reading* bahasa Inggris. CNN yang berguna untuk ekstraksi fitur dipadukan dengan LSTM yang berperan sebagai *decoder* untuk mendapatkan informasi temporal dari data. Penelitian tersebut menggunakan dataset MIRACL-VCI yang terdiri atas 10 kata dan 10 frase bahasa Inggris dengan 15 orang pembicara berbeda. Akurasi tertinggi didapatkan pada model VGGNet yang menerima input berupa *stretched concatenated images* (*sequence* gambar yang disusun dalam bentuk array 2D), yakni 76%. Kelemahan penelitian ini adalah dalam hal akurasi yang masih dapat ditingkatkan, kecepatan LSTM yang berperan sebagai *decoder*, serta masih terbatas pada *lip reading* tingkat kata dan frase.

2.3 *Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory*

Lu & Li [17] memanfaatkan CNN dan LSTM dalam *lip reading* bahasa Inggris. CNN yang digunakan adalah VGG19 yang dipadukan dengan *attention-based* LSTM. VGG19 mengekstrak fitur dari data menjadi vektor dengan *length* tertentu yang kemudian akan diproses oleh LSTM. Dengan demikian, VGG19 berperan sebagai *encoder* dan LSTM berperan sebagai *decoder*. Dataset yang 8 digunakan terdiri atas 10 angka bahasa Inggris (*zero* sampai *nine*) dengan 6 orang pembicara berbeda. Akurasi tertinggi yang berhasil dicapai adalah 88,2%, lebih tinggi 3,3% dibandingkan model umum CNN-RNN. Kelemahan penelitian ini adalah dalam hal kecepatan LSTM yang berperan sebagai *decoder* serta masih terbatas pada *lip reading* tingkat kata.

3. DATASET

Dataset yang digunakan dalam penelitian ini merupakan dataset buatan sendiri yang terdiri atas dua jenis dataset: dataset video gerakan bibir dan dataset pola visem bahasa Indonesia.

3.1 Dataset Video Gerakan Bibir

Dataset ini terdiri atas video gerakan bibir 12 orang pembicara berbeda saat mengucapkan kalimat-kalimat bahasa Indonesia dengan pola yang bervariasi. 12 orang pembicara tersebut dipilih dengan mengusahakan adanya variasi daerah asal dari setiap pembicara. Dataset menggunakan 25 kalimat bahasa Indonesia yang tersusun atas 50 kata berbeda. 50 kata bahasa Indonesia tersebut dipilih dengan mempertimbangkan persebaran visem bahasa Indonesia. Setiap pembicara akan merekam 1 video untuk setiap kalimat yang telah ditentukan di atas, sehingga setiap pembicara akan merekam 25 buah video. Dengan demikian, jumlah *raw* video dalam dataset adalah 300 buah (12 pembicara x 25 kalimat). Seluruh video tersebut diproses agar memiliki *frame rate* 15 fps, resolusi 224 x 224, dan format ekstensi mp4. Durasi video-video dalam dataset tersebut berkisar antara 5.4 detik hingga 17.3 detik, dengan jumlah *frame* per video berkisar antara 81 *frame* hingga 259 *frame*. Setiap video juga akan di-*preprocess* dengan 4 metode berbeda, yaitu *grayscaling*, *edge detection*, *image sharpening*, dan *image segmentation*.

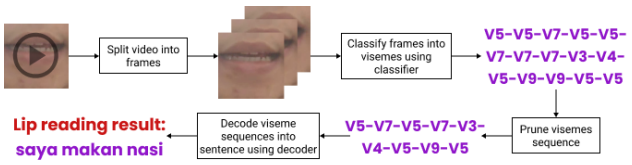
Dataset ini kemudian diproses lebih lanjut menjadi 2 jenis *training* dan *testing set*. Set pertama adalah dataset seluruh pembicara yang menggunakan video semua pembicara. *Frames* dari video pada set ini di-*undersampling* sehingga total terdapat 3,600 *frames* untuk *training* set dan 720 *frames* untuk *testing* set. Selain itu, juga terdapat 50 video utuh yang akan digunakan sebagai *testing* set. Set kedua adalah dataset pembicara terbaik yang hanya menggunakan video 3 orang pembicara terbaik berdasarkan hasil analisis. *Frames* dari video pada set ini di-*undersampling* sehingga total terdapat 1,800 *frames* untuk *training* set dan 720 *frames* untuk *testing* set. Selain itu, juga terdapat 25 video utuh yang akan digunakan sebagai *testing* set.

3.2 Dataset Pola Visem Bahasa Indonesia

Dataset ini terdiri atas pola-pola visem dari 50 kata dan 25 kalimat bahasa Indonesia yang digunakan dalam dataset video gerakan bibir. Pola visem dalam dataset ini dibuat seakan-akan seperti dihasilkan oleh *classifier* yang memiliki akurasi sempurna. Dataset ini akan digunakan untuk menguji performa *decoder* secara terpisah tanpa terpengaruh oleh performa *classifier*. Total terdapat 4,000 variasi pola visem per kata yang akan digunakan sebagai data *training* dan 500 variasi pola visem per kalimat yang akan digunakan sebagai data *testing*.

4. METODE

Metode yang digunakan dalam penelitian ini terbagi menjadi *classifier* dan *decoder*. *Classifier* berfungsi untuk melakukan klasifikasi *frames* video gerakan bibir menjadi pola visem bahasa Indonesia, sementara *decoder* berfungsi untuk melakukan *decoding* pola visem tersebut menjadi kalimat hasil prediksi. *Classifier* dan *Decoder* akan disusun dengan alur sistem seperti pada gambar 1.



Gambar 1. Alur Sistem *Lip Reading*

Sistem akan menerima input berupa video gerakan bibir. Pertama, video tersebut akan dipecah menjadi *frames* individu. Kedua, setiap *frames* akan diklasifikasikan menjadi visem bahasa Indonesia menggunakan *classifier*. Pola visem kemudian akan di-*pruning*, di mana visem tunggal yang terlalu panjang akan dipotong agar pola

visem menjadi lebih sederhana. Terakhir, pola visem akan di-*decode* menjadi kalimat bahasa Indonesia menggunakan *decoder*.

Mayoritas metode yang akan digunakan merupakan jenis ANN (*Artificial Neural Network*) [9], yakni CNN (*Convolutional Neural Network*) [22] yang umum digunakan untuk mengolah data berupa gambar serta RNN (*Recurrent Neural Network*) [16] yang umum digunakan untuk mengolah data berupa *sequence*. Secara lebih spesifik, model CNN yang akan digunakan terdiri atas VGG16 [30] dan STCNN [11], sedangkan model RNN yang akan digunakan terdiri atas LSTM [26] dan Bi-GRU [18]. Sementara itu, satu-satunya metode yang diujikan dan bukan merupakan jenis ANN adalah perpaduan Jaro-Winkler *similarity* [15] dan bigram [24].

Dalam pengujian, VGG16 dan STCNN akan digunakan sebagai *classifier*, di mana VGG16 berperan sebagai *proposed method* dan STCNN berperan sebagai metode pembanding. Sementara itu, Jaro-Winkler *similarity* dan bigram (JW-bigram), LSTM, dan Bi-GRU akan digunakan sebagai *decoder*, di mana JW-bigram berperan sebagai *proposed method* dan LSTM serta Bi-GRU berperan sebagai metode pembanding.

4.1 VGG16

VGG16 merupakan salah satu model dari CNN. Perbedaan mendasar antara VGG16 dan model CNN umum seperti AlexNet adalah pada ukuran *filter* yang digunakan pada proses konvolusi [30]. AlexNet menggunakan filter berukuran 11x11 pada *convolution layer* pertama dan 5x5 pada *convolution layer* kedua. VGG16 menggantikan *filter* berukuran besar tersebut dengan beberapa *filter* berukuran 3x3. *Layer* dan komponen yang digunakan terdiri atas:

- 13 *convolution layer* yang menggunakan *filter* berukuran 3x3 dengan *stride* 1 *pixel*. Hasil dari setiap *convolution layer* akan diubah menjadi *non-linear* dengan menggunakan ReLU *activation function*.
- 5 *max-pooling layer* yang menggunakan *filter* berukuran 2x2 dengan *stride* 2 *pixel*.
- 3 *fully-connected layer*. Hasil dari *fully-connected layer* terakhir akan diubah menjadi distribusi probabilitas dengan total 1 dengan menggunakan *softmax function*.

Pada penelitian ini, akan dilakukan *transfer learning* pada VGG16. *Weight* yang akan digunakan untuk bagian *feature extractor* model akan diambil dari *weight* model VGG16 yang telah di-*train* pada imagenet.

4.2 STCNN

Spatiotemporal CNN (STCNN) merupakan salah satu model dari CNN. Perbedaan utama dari STCNN adalah pada *convolution layers* yang digunakan [11]. CNN secara umum melakukan 2D *convolution*, sedangkan STCNN melakukan 3D *convolution* untuk dapat menangkap fitur baik secara *spatial* mau pun *temporal* dari data yang bersifat kontinyu. Selain pada *convolution layer*s yang digunakan, *layers* lain pada STCNN memiliki prinsip yang sama dengan CNN pada umumnya.

Penelitian akan menggunakan arsitektur STCNN untuk *human action recognition* [11] dengan 2 modifikasi utama untuk menyesuaikan model dengan dataset yang digunakan. Pertama, jumlah output neuron pada *fully-connected layer*s akan disesuaikan dengan jumlah kelas prediksi. Kedua, akan dilakukan penyesuaian pada setiap layer untuk dapat memproses gambar input yang memiliki ukuran 224x224x3. Selain itu, jumlah *frame* yang digunakan dalam *sequence* yang akan diproses oleh model juga akan disesuaikan dengan *nature* dari video dalam dataset.

4.3 Jaro-Winkler *Similarity* dan Bigram

Jaro-Winkler *similarity* merupakan sebuah *metric* yang dapat digunakan untuk menghitung kemiripan antar *string* [15]. Jaro-Winkler *similarity* sendiri merupakan pengembangan dari Jaro *similarity*. Oleh karena itu, konsep dari Jaro-Winkler *similarity* sangat berkaitan dengan konsep dari Jaro *similarity*. Jaro *similarity* merupakan sebuah derajat yang menunjukkan kemiripan antar dua buah *string*. Semakin besar nilai Jaro *similarity* antar dua buah *string*, maka semakin besar pula kemiripan antar kedua *string* tersebut. Jaro *similarity* menghitung kemiripan antar dua buah *string* berdasarkan jumlah minimum transposisi karakter (*insertion*, *deletion*, *substitution*) yang diperlukan untuk mengubah suatu *string* menjadi *string* lain. Terdapat beberapa variabel yang digunakan dalam perhitungan Jaro *similarity*: s1 dan s2 yang merupakan panjang dari masing-masing string, m yang merupakan jumlah dari karakter yang sama, serta t yang merupakan jumlah transposisi karakter. Rumus perhitungan Jaro *similarity* dapat dilihat pada persamaan (1).

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \quad (1)$$

Jaro-Winkler *similarity* kemudian merupakan pengembangan dari Jaro *similarity* yang memberikan nilai kemiripan lebih tinggi untuk dua *string* yang memiliki *substring* yang sama pada bagian awal kedua *string*. Jaro-Winkler *similarity* menggunakan dua buah variabel tambahan untuk melakukan perhitungan berdasarkan nilai Jaro *similarity*. Kedua variabel tersebut adalah p yang merupakan skala konstan dengan nilai standar 0.1 serta l yang merupakan panjang *substring* yang sama persis pada bagian awal kedua *string*. Rumus perhitungan Jaro-Winkler *similarity* dapat dilihat pada persamaan (2).

$$d_w = d_j + \left(l_p(1 - d_j)\right) \quad (2)$$

Sementara itu, Bigram atau 2-gram sendiri merupakan salah satu jenis n-gram *language model* [19] dengan nilai n 2. Sebuah kalimat yang terdiri atas n kata dapat dipecah menjadi n-1 buah bigram. Sebagai contoh, kalimat "aku makan nasi" yang terdiri dari 3 kata dapat dipecah menjadi 2 buah bigram: "aku makan" dan "makan nasi". Bigram memiliki kemampuan dalam merepresentasikan hubungan antar pasangan kata, sehingga dapat digunakan untuk memprediksi kata selanjutnya dari suatu *sequence* kata berdasarkan kata terakhir atau *previous word* dari *sequence* tersebut. Hal ini bisa dilakukan dengan mencari *maximum likelihood* dari probabilitas setiap kata. Probabilitas setiap kata sendiri dapat dicari dengan menggunakan persamaan (3).

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \quad (3)$$

4.4. LSTM

Long Short-Term Memory (LSTM) merupakan salah satu jenis RNN didesain untuk mengatasi masalah *vanishing gradient* [26]. LSTM mampu memilah antara informasi yang perlu disimpan dan yang perlu dibuang pada setiap *step*. Hal ini dilakukan dengan mengimplementasikan beberapa *gate* serta menambahkan sebuah *state* yang disebut sebagai *cell state* pada sel LSTM. Secara garis besar, komponen dari sebuah sel LSTM yang memungkinkan LSTM untuk dapat memilah antara informasi yang perlu dipertahankan dan informasi yang perlu dibuang terdiri atas *forget gate*, *input gate*, *cell state*, dan *output gate*.

Forget gate adalah *gate* yang menentukan seberapa banyak informasi dari *cell state* sel sebelumnya yang perlu dibuang atau

dipertahankan. *Input gate* adalah *gate* yang menentukan seberapa banyak informasi baru perlu ditambahkan pada *cell state*. *Cell state* adalah memori yang menyimpang informasi dari sel sebelumnya. Terakhir, *output gate* adalah *gate* yang menentukan nilai *hidden state* berikutnya.

4.5. Bi-GRU

Gated Recurrent Unit (GRU) merupakan salah satu jenis RNN yang juga memiliki banyak kemiripan dengan LSTM [5]. Seperti halnya pada LSTM, GRU menggunakan berbagai jenis *gate* untuk menentukan informasi yang akan disimpan dan akan dibuang. Perbedaan mendasar antara GRU dan LSTM adalah GRU tidak memiliki *cell state*. GRU hanya memanfaatkan *hidden state* dalam menyimpan informasi yang akan ditransfer. Perbedaan lain adalah pada *gate* yang digunakan. LSTM menggunakan 3 jenis *gate*, sedangkan GRU hanya menggunakan 2 jenis *gate*: *update gate* dan *reset gate*. *Update gate* adalah *gate* yang memiliki fungsi yang mirip dengan *forget gate* dan *input gate* pada LSTM. *Gate* ini bertugas untuk menentukan informasi yang harus dibuang, dipertahankan, dan ditambahkan pada *hidden state*. *Reset gate* adalah *gate* yang juga memiliki fungsi yang mirip dengan *forget gate* pada LSTM. *Gate* ini juga turut menentukan berapa banyak informasi lama yang harus dibuang atau dilupakan.

Sementara itu, *bidirectional* GRU (Bi-GRU) sendiri pada dasarnya adalah dua buah GRU yang bekerja sama untuk mengolah *sequence* dari kedua arah [18]. GRU pertama mengolah *sequence* dari awal *sequence*, sedangkan GRU kedua mengolah *sequence* dari akhir *sequence*. Dengan demikian, BiGRU mampu mempelajari ketergantungan pola dari dua arah, di mana setiap data bergantung baik pada data sebelumnya mau pun data sesudahnya.

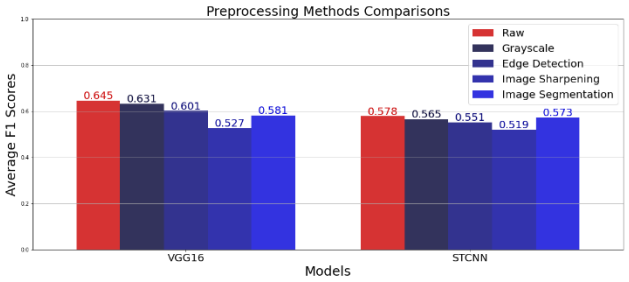
5. PENGUJIAN

Pengujian akan dilakukan dalam beberapa tahap. Pertama, akan dilakukan pengujian untuk mencari kombinasi metode *classifier*, *pruning*, dan *decoder* terbaik. Kedua, kombinasi metode terbaik tersebut akan di-*fine tuning* dengan beberapa parameter. Ketiga, kombinasi metode terbaik tersebut juga akan diuji dengan menggunakan dataset 3 pembicara terbaik untuk menguji pengaruh kualitas dataset terhadap performa sistem. Keempat, model dari pengujian kedua dan ketiga akan di-*testing* silang. Terakhir, akan dilakukan perbandingan antara performa *classifier* dengan performa sistem dari beberapa pengujian yang telah dilakukan. *Evaluation metric* yang akan digunakan pada pengujian sendiri adalah F1-*score* [27] untuk akurasi *classifier* serta WER (*word error rate*) [13] untuk akurasi *decoder* dan akurasi sistem.

5.1. Pengujian untuk Mencari Kombinasi *Classifier*, *Pruning*, dan *Decoder* Terbaik

Pengujian pertama ini memiliki dua tujuan. Tujuan pertama adalah menemukan jenis *preprocessing frame* gerakan bibir yang menghasilkan performa terbaik pada *classifier* dari 5 jenis *preprocessing* yang diujikan: *raw frame* (tanpa *preprocessing*), *grayscale* [12], *edge detection* [1], *image sharpening* [2], dan *image segmentation* [28]. Hal ini dilakukan karena penggunaan metode *preprocessing* yang berbeda akan menimbulkan efek yang berbeda pula terhadap data sehingga dapat memengaruhi performa model [29]. Tujuan kedua adalah menemukan kombinasi metode yang paling efektif baik secara akurasi mau pun kecepatan dari 12 kombinasi metode yang diujikan (12 kombinasi metode didapatkan dari kombinasi 3 *classifier*, 2 *pruning*, dan 3 *decoder*).

Langkah pertama yang dilakukan adalah melakukan *training* VGG16 dan STCNN dengan masing-masing 5 jenis *preprocessing* berbeda, sehingga total terdapat 10 kali *training* model (2 *classifier* x 5 *preprocessing*). 10 model yang telah di-*training* tersebut kemudian di-*test* performanya dalam melakukan klasifikasi *frame* gerakan bibir menjadi visem bahasa Indonesia.

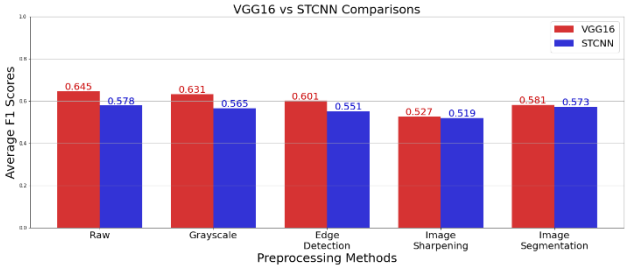


Gambar 2. Perbandingan F1-*Score* 5 *Preprocessing*

Gambar 2 menunjukkan bahwa F1-*score* untuk *raw frame* mengungguli 4 jenis *preprocessing* lainnya, baik pada VGG16 atau pun STCNN. Artinya, 4 jenis *preprocessing* tersebut justru menurunkan kemampuan model dalam mempelajari fitur visem gerakan bibir, sehingga performa model lebih buruk dibandingkan saat menggunakan *raw frame*. *Raw frame* yang menghasilkan performa terbaik pun akan dipilih untuk digunakan dalam pengujian-pengujian selanjutnya. Berikut adalah analisis penyebab penurunan performa 4 jenis *preprocessing* tersebut jika dibandingkan dengan *raw frame*:

1. *Grayscaling* dan *Image Segmentation*: Kedua proses ini memiliki efek yang mirip, yaitu menyederhanakan *frame*. Informasi yang hilang dalam proses tersebut (seperti informasi warna) ternyata dapat membantu *classifier*, sehingga performa *classifier* pun menurun saat informasi tersebut banyak yang hilang.

2. *Edge Detection* dan *Image Sharpening*: Kedua proses ini juga memiliki efek yang mirip, yaitu memperjelas *edge*. Sayangnya, kedua proses ini justru terlalu memperjelas *edge-edge* yang tidak penting dalam menentukan visem, seperti kumis, bayangan hidung, dan garis dagu. Hal-hal tersebut menyulitkan *classifier* dalam mempelajari karakteristik setiap visem.



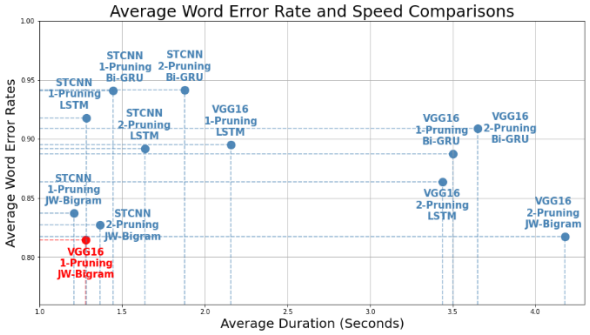
Gambar 3. Perbandingan F1-*Score* VGG16 dan STCNN

Gambar 3 lalu menunjukkan bahwa F1-*score* VGG16 mengungguli F1-*score* STCNN pada semua jenis *preprocessing*. Hal ini kembali mendukung pernyataan bahwa VGG16 lebih baik dalam mempelajari fitur dan melakukan klasifikasi *frame* gerakan bibir menjadi visem bahasa Indonesia. VGG16 berhasil mengungguli performa STCNN meski pun tidak menggunakan fitur *spatiotemporal*. Langkah berikutnya yang dilakukan adalah melakukan *training* LSTM dan Bi-GRU, lalu menguji performanya saat dibandingkan dengan JW-bigram.

Tabel 1. Perbandingan WER dan *Speed* Terbaik *Decoder*

Decoder	**JW-Bigram**	**LSTM**	**Bi-GRU**
***Best* WER**	0.003	0.294	0.473
***Best Speed* (s)**	0.614	0.345	0.310

Tabel 1 menunjukkan bahwa Jaro-Winkler *similarity* dan bigram menghasilkan *word error rate* yang jauh lebih baik daripada LSTM dan Bi-GRU, namun lebih lambat dalam melakukan *decoding*. Dengan demikian, Jaro-Winkler *similarity* dan bigram berhasil meningkatkan akurasi saat melakukan *decoding* pola visem yang 100% benar, tapi gagal memperbaiki kecepatan. Pada langkah berikutnya, dilakukan testing pada 12 kombinasi metode yang terdiri atas 2 *classifier*, 2 *pruning*, dan 3 *decoder*.



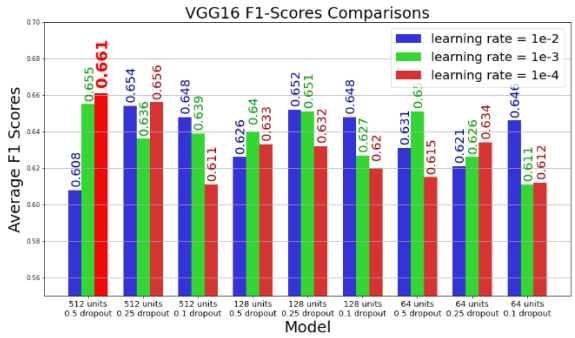
Gambar 4. Perbandingan WER dan *Speed* Sistem

Gambar 4 menunjukkan bahwa sistem VGG16+1-*Pruning*+JW-bigram memiliki WER terbaik dan sekaligus kecepatan terbaik kedua. Dengan demikian, sistem tersebut merupakan kombinasi terbaik karena paling efektif baik secara akurasi mau pun kecepatan. Sistem ini pun akan menjadi sistem yang digunakan pada pengujian-pengujian selanjutnya.

5.2. Pengujian untuk *Fine Tuning* Beberapa Parameter Metode Terbaik

Tujuan dari pengujian ini adalah untuk mencoba meningkatkan akurasi dari sistem VGG16+1-*Pruning*+JW-bigram yang telah terpilih pada pengujian 5.1 dengan cara meningkatkan F1-*score* VGG16. *Fine tuning* sendiri hanya akan dilakukan pada VGG16 sebagai *classifier* karena JW-bigram yang merupakan *decoder* terpilih bukanlah metode *machine learning*.

Langkah pertama yang dilakukan adalah melakukan *training* VGG16 dengan beberapa kombinasi *hyperparameter*. Model-model yang telah di-*training* tersebut kemudian kembali di-*test* performanya dalam melakukan klasifikasi *frame* gerakan bibir menjadi visem bahasa Indonesia.



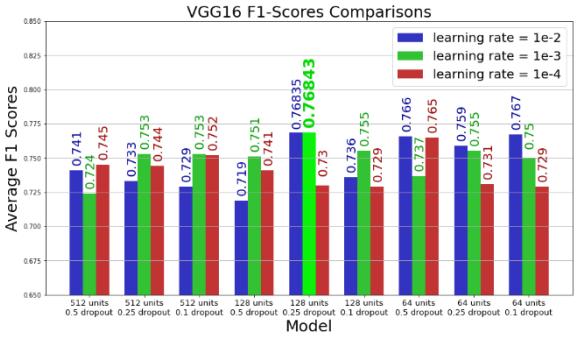
Gambar 5. Perbandingan F1-*Score* VGG16 (Seluruh Pembicara)

Gambar 5 menunjukkan bahwa F1-*score* tertinggi yang berhasil didapatkan adalah 0.661, sedikit meningkat dibandingkan dengan F1-*score* VGG16 yang digunakan pada pengujian 5.1, yakni 0.645.

Selanjutnya, dilakukan pengujian sistem dengan menggunakan VGG16 yang telah di *fine tuning* sebagai *classifier*. Hasilnya, WER sistem justru didapati mengalami kenaikan dari 0.815 menjadi 0.817. Hal ini terjadi karena sebagian *test* video mengalami kenaikan WER, sebagian tidak mengalami perubahan WER, dan sebagian mengalami penurunan WER. Kombinasi antara kenaikan dan penurunan WER inilah yang menyebabkan rata-rata WER sistem memburuk setelah *fine tuning*, meski pun angka yang dihasilkan hampir sama. Hal ini menunjukkan bahwa perubahan pada *classifier* dapat berdampak besar pada WER suatu *test* video. Perubahan berupa kenaikan F1-*score classifier* bahkan juga dapat memperburuk WER suatu *test* video pada beberapa kasus, terutama saat kenaikan F1-*score* tersebut tidak terlalu signifikan.

5.3. Pengujian dengan Menggunakan Dataset Video Pengucapan Terbaik

Tujuan pengujian ini adalah untuk menguji performa VGG16 dan keseluruhan sistem VGG16+1-*Pruning*+JW-bigram apabila menggunakan dataset dengan kualitas yang lebih baik. Dataset yang digunakan sendiri adalah dataset yang terdiri hanya atas video 3 pembicara terbaik.



Gambar 6. Perbandingan F1-*Score* VGG16 (3 Pembicara Terbaik)

Gambar 6 menunjukkan bahwa F1-*score* tertinggi yang berhasil didapatkan adalah 0.768, jauh lebih baik dibandingkan F1-*score* VGG16 terbaik saat menggunakan dataset seluruh pembicara, yakni 0.661. Perbedaan F1-*score* yang cukup signifikan ini menunjukkan bahwa kualitas dataset yang lebih baik dapat mempermudah model dalam melakukan klasifikasi *frame* gerakan bibir menjadi visem bahasa Indonesia.

Selanjutnya, kembali dilakukan pengujian sistem dengan menggunakan VGG16 yang baru saja di-*train* sebagai *classifier*. Hasilnya, sistem tersebut menghasilkan rata-rata WER 0.627. Angka ini jauh lebih baik daripada WER sistem yang menggunakan dataset seluruh pembicara (0.817). Hal ini menunjukkan bahwa peningkatan F1-*score classifier* yang signifikan dapat meningkatkan akurasi dari keseluruhan sistem. Dengan demikian, dataset dengan kualitas yang lebih baik secara tidak langsung juga dapat meningkatkan akurasi dari keseluruhan sistem.

5.4. Pengujian Silang Antara Dataset Keseluruhan dan Dataset Video Pengucapan Terbaik

Pada pengujian ini, dua sistem yang telah di-*train* dengan dataset berbeda pada pengujian 5.2 dan pengujian 5.3 akan di-*test* dengan dataset yang ditukar. Sistem yang di-*train* dengan dataset seluruh pembicara akan di-*test* dengan dataset 3 pembicara terbaik, dan sistem yang di-*train* dengan dataset 3 pembicara terbaik akan di-

test dengan dataset seluruh pembicara. Tujuan pengujian ini adalah menguji performa VGG16 dan keseluruhan sistem VGG16+1-*Pruning*+JW-bigram harus memprediksi pembicara dengan kualitas yang lebih baik atau lebih buruk daripada pembicara yang digunakan dalam *training*.

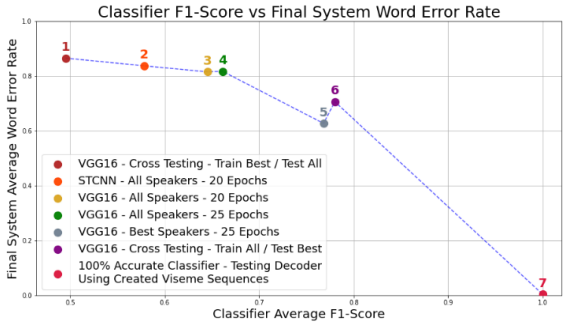
Tabel 2. Perbandingan F1-*Score* VGG16 dan WER Sistem pada Pengujian Silang

Train Set	Test Set	F1-*Score* VGG16	WER Sistem
Seluruh Pembicara	Seluruh Pembicara	0.661	0.817
3 Pembicara Terbaik	Seluruh Pembicara	0.495	0.865
Seluruh Pembicara	3 Pembicara Terbaik	0.780	0.706
3 Pembicara Terbaik	3 Pembicara Terbaik	0.768	0.627

Tabel 2 menunjukkan hasil *testing* silang. Berdasarkan hasil *testing* tersebut, ditemukan bahwa VGG16 yang di-*train* dengan dataset seluruh pembicara menghasilkan F1-*score* yang lebih baik pada *testing* kedua dataset, meski pun WER keseluruhan sistem pada dataset 3 pembicara terbaik justru lebih buruk. Artinya, VGG16 cenderung akan menghasilkan akurasi yang lebih baik saat variasi data yang digunakan dalam *training* juga semakin banyak. Bahkan, data dengan kualitas yang kurang baik pun tetap dapat membantu proses belajar dari VGG16. Oleh karena itu, penambahan variasi data dapat menjadi salah satu cara yang dapat digunakan untuk meningkatkan performa dari sistem *lip reading*, dengan catatan bahwa kenaikan F1-*score* yang dihasilkan harus cukup signifikan agar WER sistem cenderung menurun.

5.5. Perbandingan F1-*Score Classifier* dan *Word Error Rate* Sistem Beberapa Pengujian

Pada tahap ini, akan dilakukan perbandingan F1-*score classifier* dan WER sistem untuk beberapa pengujian yang telah dilakukan sebelumnya. Tujuannya adalah untuk menganalisis relasi antara kedua nilai tersebut.



Gambar 7. Perbandingan F1-*Score* VGG16 dan WER Sistem

Gambar 7 menunjukkan bagaimana WER sistem cenderung menurun seiring dengan meningkatnya F1-*score* dari *classifier*. Artinya, sistem *lip reading* yang menggunakan JW-bigram sebagai *decoder* dapat dioptimasi walau pun JW-bigram sendiri tidak dapat dioptimasi dengan cara *training* karena bukan merupakan metode *machine learning*. Salah satu cara yang dapat dilakukan untuk melakukan optimasi pada sistem tersebut adalah dengan cara melakukan optimasi pada *classifier* yang digunakan. Namun, sekali lagi perlu diingat bahwa terdapat fenomena di mana kenaikan F1-*score classifier* yang kurang signifikan justru membuat WER sistem lebih buruk, seperti terlihat pada sistem nomor 3-4 dan sistem nomor 5-6. Oleh karena itu, optimasi sistem dengan cara melakukan optimasi pada *classifier* perlu dilakukan dengan catatan bahwa kenaikan F1-*score classifier* yang dihasilkan harus cukup signifikan agar WER sistem cenderung menurun.

6. KESIMPULAN

Pada penelitian ini, dilakukan pemodelan sistem *lip reading* bahasa Indonesia tingkat kalimat dengan menggunakan VGG16 serta Jaro-Winkler *similarity* dan bigram. Pada pengujian *classifier*, didapati bahwa *raw frame* menghasilkan performa terbaik pada semua model *classifier*. Selain itu, juga didapati bahwa VGG16 selalu mengungguli F1-*score* dari STCNN, sehingga VGG16 merupakan *classifier* yang lebih unggul. Kemudian pada pengujian *decoder*, didapati bahwa Jaro-Winkler *similarity* dan bigram berhasil mengungguli LSTM dan Bi-GRU dalam hal akurasi, namun lebih buruk dalam hal kecepatan. Jaro-Winkler *similarity* dan bigram hanya berhasil mengungguli LSTM dan Bi-GRU saat telah digabungkan dengan *classifier* pada pengujian keseluruhan sistem. Pada pengujian tersebut, didapati bahwa sistem VGG16 dan JW-bigram berhasil mengungguli akurasi kombinasi metode lainnya dalam melakukan *lip reading* bahasa Indonesia tingkat kalimat. Pada pengujian 5.1, sistem dengan WER terbaik adalah VGG16+1-*Pruning*+JW-bigram dengan WER 0.815, lebih baik dari 11 kombinasi metode lainnya. Secara kecepatan pun, sistem ini juga menempati peringkat kedua terbaik dengan rata-rata durasi 1.279 detik. WER terbaik yang dicapai oleh sistem ini sendiri didapati pada pengujian 5.3 yang menggunakan dataset 3 pembicara terbaik. yakni 0.627. Oleh karena itu, sistem VGG16+1-*Pruning*+JW-bigram pun dipilih sebagai sistem terbaik karena efektif baik secara akurasi mau pun kecepatan.

Selanjutnya, ditemukan bahwa VGG16 cenderung menghasilkan akurasi yang lebih baik saat variasi data yang digunakan dalam *training* juga semakin banyak. Bahkan, data dengan kualitas yang kurang baik pun tetap dapat membantu proses belajar dari VGG16. *Testing* silang menunjukkan bahwa model yang di-*train* dengan dataset seluruh pembicara selalu mengungguli model yang di-*train* dengan dataset 3 pembicara terbaik pada semua pengujian. Oleh karena itu, penambahan variasi data dapat menjadi salah satu cara yang dapat digunakan untuk meningkatkan performa dari sistem *lip reading*. Selain itu, ditemukan bahwa F1-*score classifier* sangat berpengaruh terhadap *word error rate* sistem *lip reading*. Artinya, sistem *lip reading* yang menggunakan JW-bigram sebagai *decoder* dapat dioptimasi walaupun JW-bigram sendiri tidak dapat dioptimasi dengan cara *training* karena bukan merupakan metode *machine learning*. Salah satu cara yang dapat dilakukan untuk melakukan optimasi pada sistem tersebut adalah dengan cara melakukan optimasi pada *classifier* yang digunakan, dengan catatan bahwa kenaikan F1-*score* yang dihasilkan harus cukup signifikan agar WER sistem cenderung menurun.

7. SARAN

Berikut adalah beberapa saran yang dapat dilakukan untuk mengembangkan penelitian lebih lanjut:

1. Menyertakan data audio pada saat proses perekaman dataset agar dapat digunakan sebagai validasi dari kalimat yang diucapkan dalam video.
2. Menguji jenis CNN lain untuk digunakan sebagai *classifier* sistem *lip reading*.
3. Memodifikasi sistem *pruning* dan JW-bigram agar dapat lebih baik dalam menangani pola visem yang tidak 100% benar akibat kesalahan dari *classifier*.

4. Menambahkan jumlah kata dan jumlah variasi kalimat pada dataset.
5. Menambahkan kata-kata dengan struktur khusus dalam kalimat pada dataset, seperti kata-kata yang mengandung diftong atau kata-kata serapan dari bahasa asing.
6. Menambahkan variasi pembicara pada dataset.

8. REFERENSI

[1] Ansari, M. et al. (2017). *A comprehensive analysis of image edge detection techniques*. International Journal of Multimedia and Ubiquitous Engineering, 12(11), 1-12. DOI: 10.14257/ijmue.2017.12.11.01.

[2] Archana, J. N., & Aishwarya, P. (2016). *A review on the image sharpening algorithms using unsharp masking*. International Journal of Engi-neering Science and Computing, 6(7). DOI: 10.35940/ijitee.K2091.1081219.

[3] Arifin, F. et al. (2015). *Lip reading based on background subtraction and image projection*. 2015 International Conference on Information Technology Systems and Innovation (ICITSI), 1-3. DOI: 10.1109/ICITSI.2015. 7437727.

[4] Aulia, M. et al. (2017). *Sentence-level Indonesian lip reading with Spatiotemporal CNN and Gated RNN*. DOI: 10.1109/ICACSIS.2017.8355061.

[5] Cho, K. et al (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. DOI: 10.3115/v1/D14-1179.

[6] Dell'Aringa, A. H. B. et al. (2007). *Lip reading role in the hearing aid fitting process*. Brazilian Journal of Otorhinolaryngology, Volume 73, Issue 1, 95-99. ISSN 1808-8694. DOI: 10.1016/S1808-8694(15)31129-0.

[7] Estellers, V. & Thiran, J. (2012). *Multi-pose lipreading and audio-visual speech recognition*. EURASIP Journal on Advances in Signal Processing, 2012(1), 1-23. DOI: 10.1186/1687-6180- 2012-51.

[8] Garg, A. et al. (2016). *Lip reading using CNN and LSTM*. Retrieved from http://cs231n.stanford.edu/reports/2016/pdfs/ 217_Report.pdf.

[9] Grossi, E. & Buscema, M. (2008). *Introduction to artificial neural networks*. European Journal of Gastroenterology & Hepatology, 19(12), 1046-54. DOI: 10.1097/MEG. 0b013e3282f198a0.

[10] Harpini, A. (2019). *Disabilitas rungu di Indonesia*, p. 3. ISSN: 2442-7659.

[11] Ji, S. et al. (2010). *3D Convolutional Neural Networks for human action recognition*. Pattern Analysis and Machine Intelligence, 35(1), 495-502. DOI: 10.1109/TPAMI.2012.59.

[12] Kanan, C & Cottrell, G. W. (2012). *Color-to-grayscale: does the method matter in image recognition?*. PloS one, 7(1), e29740. DOI: 10.1371/journal.pone.0029740.

[13] Klakow, D., & Peters, J. (2002). *Testing the correlation of word error rate and perplexity*. Speech Communication, 38(1-2), 19-28. DOI: 10.1016/S0167-6393(01)00041-3.

[14] Kurniawan, A. & Suyanto, S. (2020). *Syllable-based Indonesian lip reading model*. DOI: 10.1109/ ICoICT49345.2020.9166217.

[15] Leonardo, B., & Hansun, S. (2017). *Text documents plagiarism detection using Rabin-Karp and Jaro-Winkler distance algorithms*. Indonesian Journal of Electrical Engineering and Computer Science, 5(2), 462-471. DOI: 10.11591/ijeecs.v5.i2.pp462-471.

[16] Lipton, Z. C. et al. (2015). *A critical review of recurrent neural networks for sequence learning*. arXiv preprint arXiv:1506.00019. DOI: 10.48550/arXiv.1506.00019.

[17] Lu, Y. & Li, H. (2019). *Automatic lip-reading system based on deep Convolutional Neural Network and attention-based Long Short-Term Memory*. Applied Sciences, 9(8), 1599. DOI: 10.3390/app9081599.

[18] Lynn, H. et al. (2019). *A deep Bidirectional GRU Network model for biometric electrocardiogram classification based on Recurrent Neural Networks*. IEEE Access, 7, 145395-145405. DOI: 10.1109/ACCESS.2019.2939947.

[19] Martin, S. et al. (1998). *Algorithms for bigram and trigram word clustering*. Speech communication, 24(1), 19-37. DOI: 10.1016/S0167-6393(97)00062-9.

[20] Murthy, N. & Rudregowda, S. (2020). *Lip-reading techniques: A review*. International journal of scientific & technology research, 9(02), 4378-4383. ISSN: 2277-8616.

[21] Nasuha, A. et al. (2017). *Automatic lip reading for daily Indonesian words based on frame difference and horizontal-vertical image projection*, 95(2), 393-402. ISSN: 1992-8645.

[22] O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks*. arXiv preprint arXiv:1511.08458. DOI: 10.48550/arXiv.1511.08458.

[23] Özcan, T. & Basturk, A. (2019). *Lip reading using Convolutional Neural Networks with and without pre-trained models*. Balkan Journal of Electrical and Computer Engineering, 7(2), 195-201. DOI: 10.17694/bajece.479891.

[24] Paskin, M. (2004). *Grammatical bigrams*. Advances in Neural Information Processing Systems, 14. DOI: 10.1.1.24.8418

[25] Setyati, E. et al. (2015). *Phoneme-Viseme mapping for Indonesian language based on blend shape animation*. IAENG International Journal of Computer Science, 42(3), 1-12. DOI: 10.22146/ijitee.47577.

[26] Sherstinsky, A. (2020). *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network*. Physica D: Nonlinear Phenomena, 404, 132306. DOI: 10.1016/j.physd.2019.132306.

[27] Stehman, S. V. (1997). *Selecting and interpreting measures of thematic classification accuracy*. Remote sensing of Environment, 62(1), 77-89. DOI: 10.1016/S0034-4257(97)00083-7.

[28] Yuheng, S., & Hao, Y. (2017). *Image segmentation algorithms overview*. arXiv preprint arXiv:1707.02051. DOI: 10.48550/arXiv.1707.02051.

[29] Zhu, C., & Gao, D. (2016). *Influence of data preprocessing*. Journal of Computing Science and Engineering, 10(2), 51-57. DOI: 10.5626/JCSE.2016.10.2.51.

[30] Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv: 1409.1556. DOI: 10.48550/arXiv.1409.1556.