Jurnal Politeknik Caltex Riau
Terbit Online pada laman https://jurnal.pcr.ac.id/index.php/jkt/
| e- ISSN : 2460-5255 (Online) | p- ISSN : 2443-4159 (Print) |

Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow

Nasha Hikmatia A.E.[1], Muhammad Ihsan Zul[2]

[1]Politeknik Caltex Riau, Teknik Informatika, email: nasha@alumni.pcr.ac.id
[2]Politeknik Caltex Riau, Teknik Informatika, email: ihsan@pcr.ac.id

Abstrak

Bahasa Isyarat Indonesia atau BISINDO digunakan oleh masyarakat yang memiliki keterbatasan berbicara atau mendengar namun tidak bagi masyarakat lainnya. Hal ini menyebabkan pengguna BISINDO kesulitan menyampaikan informasi karena hanya beberapa masyarakat yang mengerti BISINDO. Oleh karena itu, dikembangkan sebuah aplikasi untuk membantu komunikasi antara pengguna BISINDO dan Bahasa Indonesia secara realtime. Klasifikasi BISINDO dilakukan dengan metode Convolutional Neural Network dan arsiktektur MobilenetV2 menggunakan tensorflow. Hasil klasifikasi digunakan sebagai model pada android untuk selanjutnya dikonversi menjadi suara. Berdasarkan pengujian model, tingkat akurasi yang dihasilkan mencapai 54,8% dalam mengklasifikasi 30 bahasa isyarat. Sehingga, performa dari model dapat dikatakan belum optimal dalam mengklasifikasikan. Berdasarkan pengujian aplikasi kepada 30% responden diperoleh hasil responden sangat setuju dengan adanya aplikasi ini dengan nilai rata-rata sebesar 83,95%.

Kata kunci: *BISINDO, Tensorflow, Convolutional Neural Network, MobileNetV2, Android.*

Abstract

Indonesian Sign Language or BISINDO is used by people who have limited speech or hearing, but not for other communities. This causes BISINDO users have difficulties in conveying information because only a few people understand BISINDO. Therefore, an application was developed to help communication between BISINDO users and Indonesian in realtime. BISINDO classification is carried out using the Convolutional Neural Network method and the MobilenetV2 architecture using tensorflow. The classification results are used as a model for android which is then used as a sound. Based on model testing, the resulting accuracy rate resulted in 54.8% in the classification of 30 specified languages. Thus, the performance of the model can be said to be not optimal in classifying. Based on the application testers to 30% of respondents, it was found that respondents strongly agreed with this application with an average value of 83.95%.

Keywords: *BISINDO, Tensorflow, Convolutional Neural Network, MobileNetV2, Android.*

1. Pendahuluan

Indonesia memiliki dua jenis bahasa isyarat yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI menggunakan abjad sebagai panduan bahasa isyarat

Dokumen diterima pada 14 April, 2021
Dipublikasikan pada 31 Mei, 2021

tangan satu dan representasi dari Bahasa Indonesia yang sesuai dengan Pedoman Umum Ejaan Bahasa Indonesia (PUEBI). Sedangkan BISINDO menggunakan gerakan dua tangan yang digunakan oleh tuna rungu/tuna wicara sebagai upaya komunikasi antar pengguna bahasa isyarat.

Berdasarkan hasil kuisioner dan wawancara yang dilakukan, 91% dari 100 orang responden tunarungu di berbagai daerah telah menggunakan BISINDO dalam komunikasi. Selain itu, penyandang tuna rungu juga mendukung penggunan BISINDO sebagai media komunikasi [1]. Di sisi lain, terdapat kesulitan penggunaan BISINDO di masyarakat dikarenakan banyak masyarakat yang tidak mengerti BISINDO. Hal ini membuat tuna rungu/tuna wicara sulit menyampaikan informasi dengan akurat.

Penelitian ini bertujuan untuk mengembangkan aplikasi penerjemah BISINDO menjadi suara guna mengatasi permasalahan komunikasi antara pengguna Bahasa Isyarat Indonesia dan Bahasa Indonesia. Aplikasi yang dibangun pada platform android ini akan menerjemahkan setiap gerakan bahasa isyarat menjadi suara. Penerjemahan gerakan dilakukan dengan menggunakan pengolahan gambar.

2. Tinjauan Pustaka

2.1 Penelitian Terdahulu

Bahasa Isyarat adalah sistem yang kompleks. Suatu istilah dapat memiliki arti yang berbeda dan tidak adanya standar internasional untuk Bahasa Isyarat. Penelitian ini menggunakan Bahasa Isyarat Indonesia atau BISINDO dan Microsoft Kinect Xbox sebagai sensor. Data diklasifikasi dan diuji menggunakan metode Hidden Markov Model dengan kerapatan Gaussian. Selanjutnya data akan dievaluasi menggunakan *K-Fold Cross-Validation* selama tiga kali dengan K=10. Akurasi penelitian yang didapatkan sekitar 60%-70% mengenali BISINDO [2].

Penggunaan teknologi dapat mempermudah pengertian bahasa isyarat untuk tuna rungu wicara maupun manusia normal lainnya. Teknologi yang digunakan berupa aplikasi berbasis android yang dapat mengubah bahasa isyarat tersebut menjadi suara. Adapun metode yang digunakan dalam *Object Detection* adalah *Haar Cascade Classifier*. Aplikasi ini dapat berjalan dengan baik namun memiliki beberapa batasan yaitu *background*, pencahayaan, dan *hardware* [3].

Penelitian selanjutnya bertujuan untuk merancang sistem yang dapat mendeteksi bentuk tangan melalui citra yang ditangkap oleh *IP Camera*. Pengenalan bentuk tangan menggunakan metode *feature extraction* yaitu *Edge Detection, Vector Analysis and Pixel to Pixel Distance Analysis.* Nama dari bentuk tangan akan diklasifikasi dengan menggunakan algoritma *k-NN*. Sistem ini berhasil mendeteksi 53 bentuk tangan dengan *error rate* 11,67% [4].

Penelitian selanjutnya bertujuan untuk merancang sebuah aplikasi yang dapat mendukung orang-orang berkomunikasi dengan BISINDO. Aplikasi ini dapat menerjemahkan isyarat BISINDO menjadi sebuah teks. Adapun metode yang digunakan yaitu *Convolutional Neural Network* untuk mengidentifikasi gerakan tangan secara *real time* dan *MobileNet* untuk melatih *dataset* gambar. Tingkat keberhasilan identifikasi aplikasi mencapai 100% dengan rata-rata kinerja model tertinggi mencapai 95,13% pada 23 kategori isyarat tangan [5].

2.2 BISINDO

BISINDO atau Bahasa Isyarat Indonesia merupakan bahasa isyarat alami yang muncul dan berkembang dalam komunitas tuli [6]. BISINDO memiliki keragaman isyarat di tiap daerah karena terbentuk melalui pengaruh dari kebiasaan, nilai, dan budaya setempat. BISINDO

memiliki 2 jenis isyarat tangan yaitu isyarat statis dan dinamis(bergerak). Isyarat tangan pada BISINDO digunakan untuk abjad, angka dan kata yang langsung merujuk pada suatu hal. Pada penelitian ini, isyarat yang digunakan yaitu isyarat statis dan dinamis yang terdiri dari abjad dan kata (saya, kamu, nama, siapa).

2.3 MobileNetV2

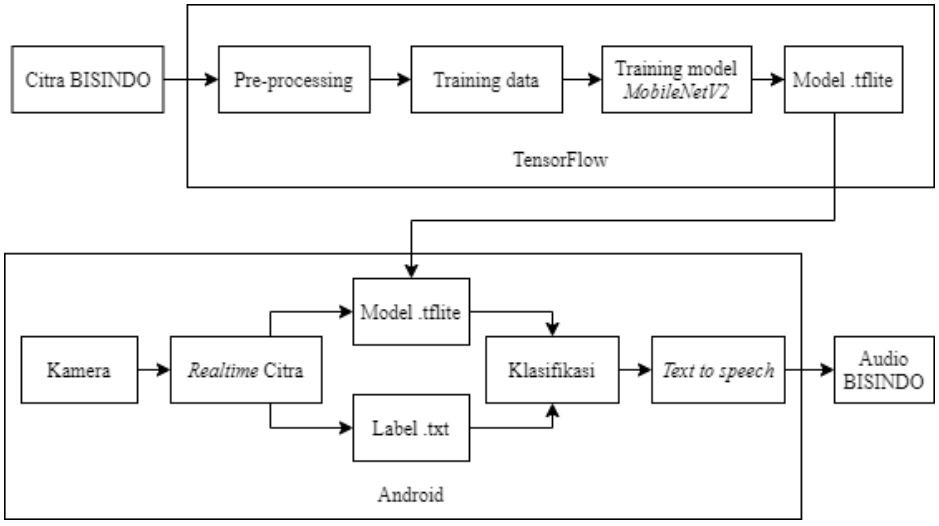
MobileNet V2 adalah salah satu arsitektur *convolutional neural network* (CNN) berbasis ponsel yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih. *MobileNet V2* merupakan penyempurnaan dari arsitektur *MobileNet. Arsitektur MobileNet* dan arsitektur CNN pada umumnya memiliki perbedaan pada penggunaan lapisan atau *convolution layer. Convolution layer* pada *MobileNetV2* menggunakan ketebalan filter yang sesuai dengan ketebalan dari input image. MobileNetV2 menggunakan *depthwise convolution, pointwise convolution, linear bottleneck* dan *shortcut connections* antar *bottlenecks* [7].

2.4 TensorFlow

TensorFlow adalah kerangka kerja komputasi untuk membangun model pembelajaran mesin. *TensorFlow* menyediakan berbagai *toolkit* yang memungkinkan untuk membuat model pada tingkat abstraksi yang disukai dan dapat menjalankan grafik pada beberapa *platform hardware*, termasuk CPU, GPU, dan TPU.

3. Metodologi Penelitian

3.1 Gambaran Umum Penelitian



Gambar 1 Blok Diagram

Pada penelitian ini terdapat beberapa tahapan yang terdiri dari pembuatan model dan penggembangan aplikasi. Proses pembuatan model dilakukan pada tensorflow yang diawali dengan akuisisi citra, pelatihan model menggunakan *MobileNetV2* yang merupakan arsitektur *Convolutional Neural Network* dan menyimpan model dalam format *tensorflow lite (.tflite).* Model .tflite ini akan digunakan pada aplikasi *android* untuk mengidentifikasi bahasa isyarat secara *realtime.* Bahasa isyarat yang diidentifikasi oleh model akan mengeluarkan *output* berupa teks. Selanjurnya teks ini diolah menggunakan *text to speech* sehingga menghasilkan *output berupa suara.*

3.2 Akuisisi Citra

Akuisisi citra yang dilakukan adalah akuisisi citra bergerak(video). Citra bergerak dipilih agar menghasilkan banyak citra untuk satu isyarat tanpa harus mengambil gambar secara berulang-ulang. Citra direkam menggunakan kamera *android* dengan kondisi cukup cahaya dan dalam jarak yang ditentukan.

```
import cv2
vidcap = cv2.VideoCapture('A.mp4')
success,image = vidcap.read()
count = 0
success = True
while success:
  cv2.imwrite("A%d.jpg" % count, image)
  success,image = vidcap.read()
  count += 1
```

Gambar 2 *Script* Konversi

Selanjutnya citra di konversi menjadi citra diam tiap frame menggunakan *python script.* Citra di simpan dalam sebuah *folder* berdasarkan isyarat dengan format .jpg. Total isyarat yang digunakan pada adalah 30 isyarat dengan masing-masing memiliki 240 gambar isyarat tangan. Adapun hasil *preprocessing* citra dapat dilihat pada Gambar 3.

Gambar 3 Hasil Akuisisi Citra

3.3 *Preprocessing*

Preprocessing citra terbagi dalam beberapa tahap. Tahap pertama yaitu menghapus citra yang tidak diperlukan secara manual. Tahap selanjutnya yaitu mengatur skala citra ke [-1,1]. Skala ini digunakan karena *MobilenetV2* mengharapkan inputnya dinormalisasi ke kisaran [-1,1]. Tahap berikutnya melakukan augmentasi citra untuk menghasilkan data pelatihan tambahan dengan menggunakan *ImageDataGenerator*. Setelah itu dilakukan pemetaan fungsi *preprocessing.* Pada tahap ini juga mengatur ukuran gambar, *batch size* dan metode pemilihan klasifikasi menggunakan *categorical*. Resolusi gambar yang didukung *MobileNetV2* adalah 128, 160, 192, atau 224px. Hasil *preprocessing* citra dapat dilihat pada Gambar 4.

Gambar 4 Hasil *Prepocessing* Citra

3.4 Pelatihan Model

Proses pelatihan diawali dengan membuat model dasar dengan metode *transfer learning*. Model yang digunakan adalah *MobileNetV2*. Model ini dikembangkan di Google dan telah dilatih pada

dataset ImageNet yang memiliki 1000 kelas. Lapisan *MobileNetV2* yang digunakan untuk ekstraksi fitur adalah lapisan terakhir sebelum *flatten operation* (lapisan *bottleneck*). *MobileNetV2* dimuat menggunakan 3 saluran input dengan bobot *Imagenet*. *Include_top = False* digunakan karena lapisan yang digunakan untuk ekstraksi fitur adalah lapisan terakhir sebelum *flatten operation* (lapisan *bottleneck*). Semua lapisan pada *MobileNetV2* akan dibekukan untuk mencegah bobot lapisan tertentu diperbarui selama pelatihan [8].

Selanjutnya dilakukan proses ekstraksi fitur dengan menyusun model, *GlobalAveragePooling2D* dan *dense* ke dalam sequential. *Dense layer* menggunakan fungsi aktivasi *softmax* yang akan mengklasifikasi gambar ke dalam banyak kelas.

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \quad (1)$$

Keterangan:
x = *input vector*
x_i = fungsi eksponensial untuk *input vector*
x_j = fungsi eksponensial untuk *output vector*
n = jumlah *class*

Model dikompilasi menggunakan *adam optimizer* dengan *learning rate* yang 0.01, 0.001 atau 0.0001. Proses pelatihan model menggunakan epoch 50, 100, dan 150.

$$m_t = \beta_1 * m_{t-1} - 1(1 - \beta_1) * g_t \quad (2)$$

$$v_t = \beta_2 * v_{t-1} - 1(1 - \beta_2) * g_t^2 \quad (3)$$

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (4)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (5)$$

$$\theta_t = \theta_{t-1} - \frac{-\alpha \cdot \widehat{m}_t}{\sqrt{\tilde{v}_t} + \epsilon} \quad (6)$$

Keterangan:
α = *Learning rate*
β_1, β_2 = *Hyperparameter*
g_t = *Gradient*
m_t = Rata-rata bergerak eksponensial dari gradien
v_t = Rata-rata bergerak eksponensial dari gradien kuadrat

Hasil pelatihan kemudian disimpan dalam model tensorflow. Model ini kemudian dikonversi menjadi model tensorflow lite (.tflite). *Tensorflow lite* digunakan karena dapat mengurangi ukuran file dan meningkatkan kecepatan eksekusi tanpa memengaruhi keakuratan sehingga dapat diterapkan pada *android.*

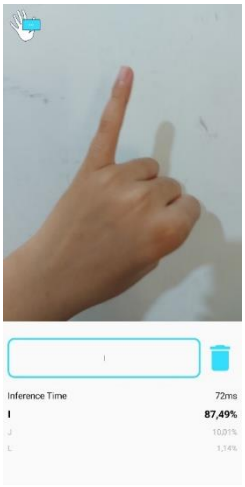
3.5 Implementasi

Aplikasi penerjemah BISINDO dibuat menggunakan *android studio*. Model .tflite yang dihasilkan pada pelatihan sebelumnya disimpan ke dalam sebuah folder bersamaan dengan label.txt pada projek *android*. Untuk dapat megeksekusi model *tensorflow lite* pada projek android, maka harus membuat objek *tensorflow inference interface.* Setelah itu dilakukan pemetaan label berdasarkan output. Apabila tingkat keberhasilan deteksi diatas 80% selama

beberapa kali secara berturut-turut maka akan dikonversi menjadi suara menggunakan *text to speech.*

4. Hasil dan Pembahasan

4.1 Hasil Perancangan



Gambar 5 Halaman Utama Aplikasi

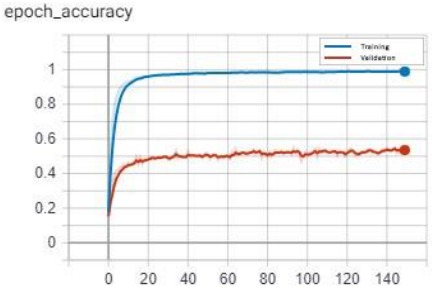
Aplikasi untuk deteksi bahasa isyarat dapat dilihat pada Gambar 10. Pada aplikasi ini citra yang akan digunakan sebagai *input* akan diambil secara otomatis oleh kamera *android*. Persentase hasil deteksi bahasa isyarat dapat dilihat pada bagian bawah halaman. Apabila persentase deteksi diatas 80%, maka huruf/kata akan ditulis pada box area. Pada b*oxarea* dapat diisi oleh beberapa huruf/kata. Selanjutnya huruf/kata tersebut diubah secara otomatis menjadi suara.

Halaman ini juga menyediakan fitur hapus kata pada *box area.* Terdapat 2 pilihan dalam menghapus kata yaitu dengan sekali klik atau dengan menahan klik beberapa saat. Fitur sekali klik akan menghapus setiap huruf dalam area tersebut. Sedangkan fitur klik dengan beberapa saat akan menghapus seluruh kata yang ada dalam *box area.*

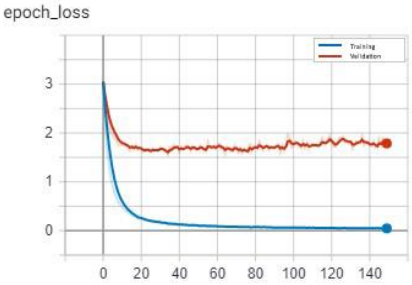
4.2 Pengujian dan Analisis

4.2.1 Confusion Matrix

Pengujian Confusion Matrix dilakukan untuk menguji akurasi pada model yang dilatih. Pelatihan model dilakukan beberapa kali dengan mengatur skenario data dan parameter yang digunakan seperti *input image, learning rate* dan *epoch. Optimizer* yang digunakan adalah *adam optimizer.*

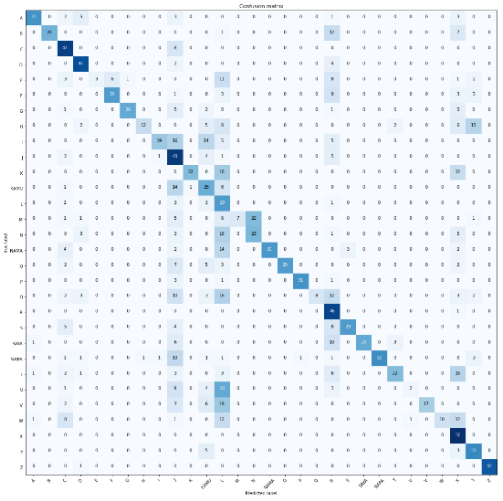


Gambar 6 Grafik *Accuracy*



Gambar 7 Grafik *Loss*

Berdasarkan pengujian yang dilakukan terhadap *input image*, *learning rate* dan *epoch* menunjukkan bahwa akurasi model dipengaruhi oleh ketiga parameter. Akurasi dari data *validation* mencapai 54,8% dengan nilai *loss* 1,713 seperti yang terlihat pada Gambar 11 dan Gambar 12. Adapun skenario data yang digunakan yaitu 4608 *training set*, 1152 *validation set* dan 1440 *test set* dengan parameter *input size 224*, *learning rate* 0.0001 dan *epoch* 150.



Gambar 8 Pengujian Confusion Matrix

Confusion matrix pada Gambar 13 menunjukkan bahwa masih terdapat kesalahan prediksi. Hal ini dapat disebabkan oleh pelatihan model tidak cukup, kurangnya data yang dilatih atau pegaruh dari *BatchNormalization layer* pada *MobilNetV2*. *BatchNormalization* digunakan untuk menormalkan input atau output dari fungsi aktivasi di *hidden layer*. *BatchNormalization* berisi 2 beban yang tidak dapat dilatih selama pelatihan yaitu variabel yang melacak *mean* dan varian.

Selama pelatihan (*BatchNormalization* dengan argumen *training=Ttrue)*, lapisan akan menormalkan inputnya menggunakan mean dan varians dari batch input saat ini. Hal ini dapat menghancurkan apa yang telah dipelajari model. Untuk menghindari hal tersebut dapat dilakukan dengan mengatur *BatchNormalization* dengan menggunakan *taining=False*. *Training=False* akan menormalkan input menggunakan *mean* dan varians dari statistik bergeraknya yang dipelajari selama pelatihan.

4.2.2 Black Box

Pengujian dengan metode *black box* yakni fokus terhadap fungsionalitas perangkat lunak berdasarkan fungsi dan halaman yang tersedia. Pengujian dilakukan dengan cara menjalankan aplikasi berdasarkan skenario yang telah dirancang sebelumnya. Pada pengujian ini diperoleh hasil bahwa setiap fungsi aplikasi dapat berjalan dengan baik dan benar serta sesuai dengan yang diharapkan.

4.2.3 Sistem

Tabel 1 Pengaruh *Background*

Citra	Label	Kemampuan Deteksi
	U	Tidak berhasil
	U	Berhasil
	U	Tidak Berhasil
	U	Tidak berhasil
	U	Tidak berhasil
	U	Tidak berhasil

Pengujian ini dilakukan untuk mengetahui pengaruh *background* terhadap kemampuan dan keakuratan aplikasi dalam mendeteksi isyarat tangan. Hasil dari pengujian pada Tabel 1 diketahui bahwa *background* yang digunakan sangat mempengaruhi tingkat keberhasilan deteksi suatu objek. Semakin kontras warna objek dengan *background,* maka semakin mudah aplikasi dalam mendeteksi objek.

4.2.4 Kuisioner

Pengujian kuisioner dilakukan terhadap 30 responden yang dipilih secara acak dengan latar belakang pekerjaan yang berbeda-beda. Dari 30 responden, terdapat 1 responden yang merupakan pengguna BISINDO dan 29 responden bukan pengguna BISINDO.

Pengujian yang dilakukan kepada pengguna BISINDO, diperoleh hasil bahwa responden sangat setuju dengan pernyataan aplikasi dapat membantu komunikasi antara pengguna bahasa isyarat dan Bahasa Indonesia, setuju dengan pernyataan aplikasi mudah digunakan dan tidak setuju dengan pernyataan aplikasi mudah dipahami dan memiliki tampilan yang menarik.

Berdasarkan hasil yang diperoleh, responden sangat setuju aplikasi ini dapat membantu komunikasi antara pengguna bahasa isyarat dan Bahasa Indonesia. Hal ini membuktikan aplikasi yang dibangun dapat diterima dengan baik.

5. Kesimpulan

Aplikasi penerjemah Bahasa Isyarat Indonesia (BISINDO) menjadi suara secara *realtime* berhasil dibuat dengan menggunakan arsitektur *MobilenetV2*. Akurasi tertinggi model yang dapat diperoleh mencapai 54,8%. Hal ini dikarenakan pada tahap pelatihan, model menggunakan 30 *class* atau 30 isyarat dengan masing-masing memiliki 240 gambar. Isyarat ini tidak hanya terdiri dari isyarat statis namun juga terdapat isyarat dinamis. Selain itu jumlah iterasi yang dapat dilakukan hanya sebanyak 150 iterasi. Iterasi yang lebih tinggi dapat meningkatkan akurasi pada model.

Adapun saran untuk penelitian selanjutnya adalah menambahkan jumlah citra dengan berbagai macam *background* dan menambahkan LSTM (*Long Short Term Memory)* pada model sehingga dapat mendeteksi isyarat yang dinamis dan meningkatkan akurasi pada *class.*

Daftar Pustaka

[1] R. A. Mursita, "Respon Tunatungu Terhadap Penggunaan Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO)," *INKLUSI Journal of Disability Studies,* vol. 2, pp. 221-232, 2015.

[2] T. Handika, R. I. M. Zen, M. D. P. Lestari and I. Sari, "Gesture Recognition For Indonesian Sign Language (BISINDO)," *Journal of Physics: Conference Series,* vol. 1028, no. 1, pp. 1-8, 2018.

[3] M. Luter, K. Frehadtomo, P. R. N. Dayawati and F. N. Prawita, "Sign To Voice Aplikasi Alat Bantu Komunikasi Untuk Tuna Rungu Wicara," 2015.

[4] M. I. Zul, "Feature Extraction For Hand Shape Recognition By Using IP Camera," *In Regional Conference on Computer and Information Engineering,* 2018.

[5] P. Yugopuspito, I. M. Murwantara and J. Sean, "Mobile Sign Language Recognition for Bahasa Indonesia Using Convolutional Neural Network," *Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia,* pp. 84-91, 2018.

[6] G. Gumelar, H. Hafiar and P. Subekti, "Bahasa isyarat indonesia sebagai budaya tuli melalui pemaknaan anggota gerakan untuk kesejahteraan tuna rungu," *INFORMASI : Kajian Ilmu Komunikasi,* vol. 48, no. 1, pp. 65-78, 2018.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "Mobilenetv2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 4510-4520, 2018.

[8] "TensorFlow," [Online]. Available: tensorflow.org. [Accessed 28 September 2020].

[9] I. GoodFellow, Y. Bengio and A. Courville, Deep Learning, MIT press, 2018.