

SISTEM ISYARAT BAHASA INDONESIA (SIBI) METODE CONVOLUTIONAL NEURAL NETWORK SEQUENTIAL SECARA REAL TIME

Okky Dwi Nurhayati^{*1}, Dania Eridani², Muhammad Hafiz Tsalavin³

^{1,2,3}Universitas Diponegoro, Semarang

Email : ¹okkydwin@gmail.com, ²dania@ce.undip.ac.id, ³hafiztsalavin@gmail.com

^{*}Penulis Korespondensi

(Naskah masuk: 23 Februari 2021, diterima untuk diterbitkan: 20 Agustus 2022)

Abstrak

Bahasa isyarat biasanya dilakukan oleh tuna rungu dan tuna wicara. Sistem Isyarat Bahasa Indonesia (SIBI) merupakan isyarat tangan yang digunakan di Indonesia. Penggunaan isyarat tangan tidak selalu dimengerti oleh manusia normal, sehingga dibutuhkan perangkat tambahan yang dapat mempermudah dalam menerjemahkan suatu isyarat. Perangkat tambahan yang dikembangkan dalam penelitian ini melibatkan teknologi visi komputer *deep learning* sehingga menghasilkan *tools* untuk menerjemahkan bahasa isyarat tangan. Penelitian ini menggunakan data yang berjumlah 29 citra isyarat, yang terdiri dari 26 huruf alfabet dan 3 isyarat tambahan yaitu spasi, hapus, dan tidak terklasifikasi. *Pre-processing* dilakukan dengan mengubah gambar ke dalam bentuk HSV, kemudian dilakukan *cropping* dan *thresholding* untuk memudahkan pengenalan. Metode *Convolutional Neural Network* (CNN) yang digunakan sebagai *feature learning* dan pengklasifikasi isyarat tangan pada sebuah obyek. Pengujian dilakukan pada berbagai variasi cahaya berkisar 10-200 lux, serta jarak tangan ke webcam 50-200 cm. Hasil penelitian dengan metode CNN pada citra isyarat tangan memberikan akurasi sebesar 97,2%, presisi 91,96%, sensitivitas 91,9%, spesifisitas 91,96% dan f1 score 91,9%.

Kata kunci: alfabet, *feature learning*, klasifikasi, CNN, variasi cahaya

A REAL-TIME INDONESIAN LANGUAGE SIGN SYSTEM USING THE CONVOLUTION NEURAL NETWORK METHOD

Abstract

Sign language is usually used by deaf and speech impaired persons. The Sistem Isyarat Bahasa Indonesia (SIBI) is a hand signal language used in Indonesia. The use of hand signals is not always understood by normal humans, such that additional devices are needed to make sign translation easier. The additional device in this study is developed using deep learning and computer vision technology to produce a hand signal translation tool. This study uses 29 sign images for a dataset, consisting of 26 letters of the alphabet and 3 additional signs, namely space, delete, and unclassified. Pre-processing is performed by converting the image into HSV, cropping, and thresholding to make easy recognition. The convolutional neural network (CNN) method is then used as a learning feature and hand signals classifier on an object. The testing phase is performed on various lights ranging from 10-200 lux and the hand distance to the webcam is about 50-200 cm. Experimental results show that the CNN method on the hand signal image could provide an accuracy of 97.2%, precision of 91.96%, sensitivity of 91.9%, specificity of 91.96%, and F1 score of 91.9%, respectively.

Keywords: alphabet, *feature learning*, classification, CNN, various lights

1. PENDAHULUAN

Bahasa isyarat tangan merupakan salah satu cara komunikasi yang dilakukan oleh kaum tunarungu dan tunawicara. Namun tidak setiap orang memahami bahasa isyarat tangan. Keterbatasan inilah yang masih menjadi sebuah masalah bagi tunawicara dan tunarungu dalam berkomunikasi dengan manusia normal sehingga diperlukan sebuah *tools* yang mampu mengubah bahasa isyarat tangan menjadi

sebuah keluaran yang berupa pengenalan *gesture* supaya mudah dipahami oleh manusia normal.

Pengenalan *gesture* merupakan antarmuka yang dapat mengenali gerak isyarat tangan dan mengubah gerakan tangan tersebut sebagai perintah yang dapat dipahami oleh komputer (Haria, dkk, 2017). Pengenalan *gesture* digunakan sebagai penerjemah bahasa isyarat pada tunawicara. Hal ini disebabkan karena banyaknya orang yang tidak mengerti bahasa

isyarat tangan tunawicara. Sehingga, tunawicara menjadi kesulitan dalam berinteraksi di dalam kehidupan masyarakat. Pengenalan *gesture* sebagai penerjemah bahasa isyarat dengan mendeteksi perubahan gerak tangan dapat dikembangkan menggunakan perangkat android (Andrian, M. Y., Purwanto, D., & Mardiyanto, R., 2017). Pengenalan *gesture* atau gerakan tangan dengan mendeteksi perubahan gerak berdasarkan arsitektur *Convolutional neural network* (CNN) telah dikembangkan namun masih dilakukan secara statis (Adithya, V., Rajesh, R., 2020).

Di Indonesia menggunakan Sistem Isyarat Bahasa Indonesia (SIBI) yang merupakan turunan dari *American Sign Language* (ASL). SIBI secara resmi digunakan oleh semua SLB di bawah Kementerian Pendidikan dan Kebudayaan. SIBI memiliki 26 ejaan jari yang menunjukkan 26 huruf alfabet dengan menggunakan satu tangan, 24 tanda berupa gerakan statis, serta 2 tanda berupa gerakan tangan dinamis (UU Mendikbud, 1994).

Penelitian pengenalan bahasa isyarat telah dilakukan di banyak negara, yang sebagian besar menggunakan metode pengolahan citra maupun visi komputer. Identifikasi citra merupakan langkah yang penting dalam pengenalan gerakan tangan. Pengenalan gerakan tangan dengan menggunakan pengolahan citra dan teknik ekstraksi ciri dilakukan untuk mengenali dan mengklasifikasi gerakan tangan dengan tingkat akurasi semaksimal mungkin. Beberapa teknik pra-pengolahan yang dilakukan seperti Histogram Gradient, *Principal Component Analysis*, dan *Local Binary Patterns*. Model dibangun dengan menggunakan deteksi tepi *canny*, dan teknik. Setelah melalui pra-pengolahan kemudian data diklasifikasi dengan beberapa metode klasifikasi seperti *Random Forests*, *Support Vector Machines*, *Naïve Bayes*, *Logistic Regression*, *K-Nearest Neighbours*, *Multilayer Perceptron* untuk mencari hasil yang paling efektif (Sharma, dkk, 2020).

Saat ini *machine learning* dan *deep learning* menjadi salah satu metode yang sangat populer dan memiliki banyak kelebihan untuk dikembangkan. Salah satunya adalah *Convolutional Neural Network* (CNN). *Convolutional Neural Network* digunakan untuk mendeteksi dan mengenali sebuah objek. Salah satu kelebihan CNN dianggap sebagai model terbaik untuk menyelesaikan masalah yang berhubungan dengan deteksi obyek dan pengenalan obyek karena tidak memerlukan komputasi yang besar dalam prosesnya. Hasil yang diperoleh pada penelitian memberikan akurasi sebesar 92,33% dengan pengujian metode *10-fold cross validation*.

Perhitungan *precision* dan *recall* memiliki nilai yang baik, masing-masing sebesar 97,51% dan 94,33% (Alam, I. F., Sarita, M. I., & Sajiah, A. M., 2019). Metode tersebut memiliki kelebihan dalam permasalahan *image recognition* dan klasifikasi pengenalan gestur manusia. *Convolutional Neural Network* digunakan untuk melatih data sehingga

dapat mengenali data isyarat berupa huruf SIBI dan dapat digunakan untuk mendeteksi serta mengenali obyek pada sebuah gambar (Bheda, V., dan D. Radpour, 2017). *Convolutional Neural Network* terdiri dari banyak lapisan pengolahan untuk merepresentasikan data dalam berbagai tingkat. Namun, CNN memiliki jutaan parameter dan banyak lapisan, yang sulit untuk dilatih, dan terkadang diperlukan waktu yang lama untuk menyesuaikan parameter. Dalam penelitian tersebut CNN digunakan untuk mengekstrak fitur gambar, algoritma AdaBoost digunakan untuk mengklasifikasi Softmax menjadi gambar yang dapat dikenali. Model ini menghasilkan peningkatan akurasi 3% dari model CNN *supervised*, dan dapat mengurangi biaya waktu pelatihan (Lee, S. J., Chen, T., Yu, L., & Lai, C. H., 2018).

Penelitian terkait CNN dengan arsitektur LeNet dilakukan oleh (Bakti, M.B.S., & Yuliana, M.P., 2019). Arsitektur CNN LeNet yang diproses dalam 3 iterasi yaitu 25 epoch, 50 epoch dan 100 epoch. Hasil penelitian menunjukkan tingkat akurasi mengalami peningkatan untuk setiap tahap berkisar 67.66%, 89.44% , dan 96.44%. Begitupula dalam prediksi data mengalami peningkatan akurasi berkisar 79.23%, 90.45%, dan 98.89 .

Penelitian terkait CNN lainnya dilakukan oleh (Haria, A., Archanasri S, Nivedhitha A, Shristi P, Jyothi S.N, 2017) menggunakan peralatan *input* seperti *mouse* dan *keyboard* sebagai antarmuka pengguna dan komputer. Penelitian dilakukan dengan merancang sistem pengenalan gerak tangan tanpa marker secara statis maupun dinamis. Sistem yang dibuat dapat menerjemahkan gerak tangan menjadi sebuah aksi seperti membuka situs web, membuka aplikasi VLC Player, dan PowerPoint. Gerak dinamis digunakan untuk mengacak slide dalam presentasi.

Penelitian yang dilakukan oleh (Darmatasia, 2021), 2021) bertujuan melakukan pengenalan alfabet pada Sistem Isyarat Bahasa Indonesia (SIBI). Penelitian diawali dengan pengumpulan dataset alfabet (A, B, C, D, dan E) kemudian pengenalan menggunakan algoritma CNN. Citra masukan yang diberi filter *gradient* dapat meningkatkan akurasi pengenalan sebesar 85%. Sedangkan citra yang tidak difilter memiliki akurasi sebesar 25%. Hasil akurasi tertinggi sebesar 98% diperoleh dengan jumlah iterasi 200 *epoch*. Hasil uji menunjukkan bahwa metode yang digunakan dapat meningkatkan akurasi pengenalan yang memiliki *background* dengan tepi yang lebih tajam.

Penelitian lainnya terkait CNN dilakukan oleh (Adithya V., Rajesh R., 2019). Penelitian dilakukan untuk mengenali gerak tangan pada dua dataset yaitu dataset gerak tangan NUS dan dataset jari tangan A Amerika). Hasil penelitian menunjukkan CNN *Deep Learning* memberikan akurasi pengenalan yang lebih baik daripada metode *Deep Learning Restricted Boltzmann Machine* (RBM).

Perkembangan teknologi *deep learning* untuk menghasilkan piranti bantu dalam menerjemahkan

bahasa isyarat dapat dilakukan dengan melibatkan beberapa teknik pengolahan citra seperti segmentasi, ekstraksi ciri warna, dan pengambilan. Perangkat keras yang digunakan dalam penelitian adalah komputer dan kamera *webcam*. Dalam penelitian tersebut menggunakan metode pembelajaran *supervised learning* dalam menentukan bahasa isyarat tangan berupa 26 huruf alfabet dan 3 kelas tambahan yaitu spasi, hapus dan tidak terklasifikasi (tidak memuat isyarat apapun pada gambar) dengan menggunakan dataset yang sudah dibuat sendiri oleh peneliti (Koh, J.H., & S. H. A. Ali., 2017).

Beberapa penelitian yang terkait dengan *deep learning* diantaranya mengenai pengenalan isyarat tangan menggunakan pengolahan visual digital metode *convex hull*, metode *convexity defects*, maupun pusat massa. Hasil penelitian tersebut menghasilkan sebuah nilai-nilai yang dapat dilatih untuk melakukan pengenalan isyarat pada tahap selanjutnya. Dalam penelitian tersebut, sistem dilatih menggunakan *artificial neural network* pada dua kondisi percobaan. Sistem dapat menangkap citra tangan dari jarak 50 - 65 cm. Penelitian selanjutnya mengenai *deep learning* menggunakan metode *contrast stretching* setelah melalui pra pengolahan dan CNN untuk mengenali isyarat tangan. Metode yang di pilih diimplementasikan dengan berbagai variasi cahaya yang berada pada *range* 26-140 lux, serta jarak tangan ke kamera pada 120-200 cm. Hasil keluaran memberikan nilai akurasi 97,5%, nilai presisi 97,57%, tingkat sensitivitas 97,5%, nilai spesivitas 99,77% dan f1 score 97,45% (Adi, H. A., & Candradewi, I., 2019).

Penelitian lainnya mengenai *Hand Gesture Recognition System* menggunakan webcam untuk men-track gerakan tangan menggunakan metode *Region Of Interest (ROI)* dan algoritma *kernelized correlation filters (KCF)* dengan mengatur ukuran gambar sebesar 100x120 pel. Sistem pengenalan *gesture* tangan yang efektif menggunakan tiga kombinasi komponen, yaitu deteksi tangan, *tracking* tangan dan pembacaan gerakan tangan. Metode lain mengenai *deep learning convolutional neural network (CNN)* dengan arsitektur AlexNet dan VGGNet yang telah di modifikasi memberikan akurasi *pelatihan* 99,90% dan akurasi *pengujian* 95,61% dilakukan oleh (Chung, H.Y., Y. L. Chung, dan W. F. Tsai, 2019).

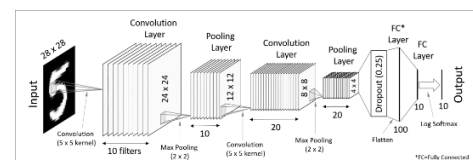
Penelitian lainnya mengenai metode dan aplikasi *deep learning* yang membahas mengenai metode-metode umum *deep learning* serta aplikasinya dalam berbagai ragam pengolahan sinyal dan informasi. Pembahasan secara mendetail telah dibahas dalam sebuah buku mengenai dasar-dasar dari *machine learning* yang terkait dengan teknik-teknik yang dikembangkan seperti *Framework* pembelajaran PAC, *Rademacher Complexity*, teknik VCDimensio, teknik *Support Vector Machines*, metode Kernel, teknik *Boosting*, teknik *On-Line learning*, teknik *MultiClass Classification*, teknik

Ranking, teknik *Regression*, *Maximum Entropy Model*, *Conditional Maximum Entropy Models*, *Algorithmic Stability*, *Dimensionality Reduction*, *Learning Automata* dan *Languages*, *Reinforcement Learning* (Mohri, A., M., Rostamizadeh, A., dan Talwalkar, 2018).

Sistem Isyarat Bahasa Indonesia (SIBI) di turunkan dari bahasa isyarat Amerika (*American Sign Language*) dan diakui secara resmi di Indonesia. SIBI secara resmi digunakan oleh semua SLB dibawah Kementerian Pendidikan dan Kebudayaan (Kemendikbud). Peraturan penggunaan SIBI terdapat di dalam peraturan Menteri Pendidikan Republik Indonesia nomor 0161/U/1994. SIBI memiliki 26 ejaan jari yang menunjukkan 26 alfabet dengan menggunakan satu tangan. Dua puluh empat tanda berupa gerakan statis serta dua tanda berupa gerakan dinamis (J dan Z). Alfabet SIBI tersusun dari kombinasi bentuk jari dan tangan (Pajar, T.Y., D. Purwanto, dan H. Kusuma, 2018).

Penelitian CNN untuk mengklasifikasi berbagai citra foto candi kuno Indonesia. Hasil penelitian menunjukkan nilai akurasi model pelatihan sebesar 98,99% dan pengujian sebesar 85,57% dengan waktu performansi 389.14 detik. Model yang kompleks memerlukan waktu pelatihan lebih lama (Danukusumo, K., Pranowo, & Maslim, M., 2017).

Convolutional neural network merupakan jenis *deep learning* yang di dalam nya terdapat minimal satu layer konvolusi yang di anggap sebagai model terbaik untuk menyelesaikan masalah pada pengenalan dan deteksi obyek. *Convolutional neural network* menggunakan dua tipe lapisan tersembunyi, yaitu: lapisan konvolusi dan lapisan *subsampling*. Setiap *layer* pada CNN menghubungkan *neuron* dari lapisan masukannya dalam bentuk reseptif lokal (potongan persegi) yang di sebut *feature map* atau peta ciri. Melalui bobot bersama (*shared weights*) kepada sebuah peta ciri. Gambar 1 merupakan arsitektur CNN (Traore, B. ., Kamsu-Foguem, & Tangara, F, 2018).



Gambar 1. Arsitektur *Convolutional Neural Network* (Traore, B. ., Kamsu-Foguem, & Tangara, F, 2018)

Lapisan konvolusi merupakan lapisan pertama yang menerima masukan citra pada arsitektur Gambar 1. Di dalam model CNN terdapat beberapa proses antara lain proses konvolusi, proses pooling, proses dropout, proses flatten dan proses klasifikasi. Lapisan konvolusi menggunakan filter untuk mengekstraksi citra input. Terdapat parameter yang diubah untuk memodifikasi ciri setiap lapisan, yaitu;

ukuran filter, stride dan padding (Arrofiqoh & Harintaka, 2018).

Fungsi aktivasi ReLU (*Rectification Linear Unit*) merupakan operasi yang digunakan untuk mengenalkan nonlinearitas dan meningkatkan representasi dari model. Fungsi aktivasi ReLU adalah $f(x) = \max(0, x)$. Nilai keluaran neuron dinyatakan sebagai 0 jika nilai *input* negatif. Tetapi, jika nilai *input* adalah positif, maka nilai *output* neuron adalah nilai input aktivasi (Arrofiqoh & Harintaka, 2018).

Pooling atau subsampling adalah pengurangan ukuran matriks. Dua macam *pooling* yang dapat digunakan yaitu *average pooling* dan *max pooling*. *Fully connected layer* merupakan kumpulan dari proses konvolusi yang mendapatkan input dari proses sebelumnya untuk menentukan fitur yang paling berkorelasi dengan kelas tertentu (Arrofiqoh & Harintaka, 2018). *Pooling layer* CNN bertujuan untuk mengurangi ukuran citra sehingga mudah digantikan dengan *convolution layer*. *Fully Connected Layer* merupakan lapisan dengan semua neuron aktivasi dari lapisan sebelumnya terhubung dengan neuron di lapisan selanjutnya. *Fully Connected Layer* memiliki beberapa lapisan tersembunyi, fungsi aksi, lapisan keluaran dan fungsi *Loss* yang digunakan untuk melakukan klasifikasi (Yanur, 2018).

Fungsi aktivasi softmax digunakan untuk menghasilkan nilai yang diinterpretasi sebagai probabilitas yang belum di normalisasi untuk setiap kelas. Nilai setiap kelas dihitung dengan menggunakan fungsi softmax seperti yang ditunjukkan oleh Persamaan (1) (Arrofiqoh & Harintaka, 2018).

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^D e^{x_{ijt}}} \quad (1)$$

dengan : y_{ijk} = vektor yang berisi nilai antara 0 dan 1; x = vektor yang berisi nilai yang dari lapisan *fully-connected* yang terakhir.

Nilai *precision*, nilai *recall*, dan nilai *accuracy* merupakan parameter yang digunakan untuk menentukan kinerja atau performansi klasifikasi. Sedangkan *confusion matriks* digunakan untuk menghitung faktor-faktor kinerja. Gambar 2 menunjukkan *confusion matriks* yang digunakan dalam penelitian (Tharwat, 2018).

		True/Actual Class	
		Positive (P)	Negative (N)
Predicted Class	True (T)	True Positive (TP)	False Positive (FP)
	False (F)	False Negative (FN)	True Negative (TN)
		P=TP+FN	N=FP+TN

Gambar 2. *Confusion Matrix* (Tharwat, 2018)

Gambar 2 menunjukkan empat kemungkinan keluaran. Diagonal berwarna hijau mewakili prediksi benar sedangkan diagonal berwarna merah muda menunjukkan prediksi yang salah. *True Positive* (TP) yaitu data dimana antara prediksi dengan kenyataan sesuai. *True Negative* (TN) yaitu data diluar kategori uji yang memiliki kecocokan antara prediksi dan kenyataan. *False Positive* (FP) yaitu data diluar kategori uji yang di prediksi sesuai dengan kategori uji. Sedangkan *False Negative* (FN) yaitu data kategori uji yang diprediksi diluar kategori uji. Dari empat kemungkinan keluaran tersebut, dapat dilakukan operasi perhitungan untuk mencari nilai *precision*, nilai *recall*, dan nilai *accuracy*. Presisi di definisikan sebagai jumlah sampel positif benar yang diklasifikasi benar terhadap jumlah sampel yang diprediksi benar.

Nilai *precision* merupakan perbandingan antara nilai TP dengan jumlah data yang diprediksi benar, ditunjukkan oleh Persamaan (2) (Tharwat, 2018).

$$\text{Presisi} = \frac{TP}{FP+TP} \quad (2)$$

Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap jumlah item yang tersedia. Nilai *Recall* / sensitivitas merupakan perbandingan nilai TP dengan jumlah data kategori uji. Perhitungan sensitivitas ditunjukkan oleh Persamaan (3) (Tharwat, 2018).

$$\text{Sensitivitas (Recall)} = \frac{TP}{TP+FN} \quad (3)$$

Akurasi di definisikan sebagai ukuran kedekatan antara hasil pengukuran dengan nilai yang sebenarnya. Nilai *accuracy* merupakan perbandingan total dari semua data klasifikasi dengan aturan yang sudah ditentukan. Perhitungan akurasi ditunjukkan oleh Persamaan (4) (Tharwat, 2018).

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

F1 Score adalah nilai rata-rata harmonik antara nilai presisi dan *recall*. *F1 Score* disebut dengan *F-measure*. *F1 score* merepresentasikan rata-rata harmonik dari presisi dan sensitivitas (*recall*) yang ditunjukkan pada Persamaan (5) (Tharwat, 2018).

$$F1\ Score = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

2. METODE PENELITIAN

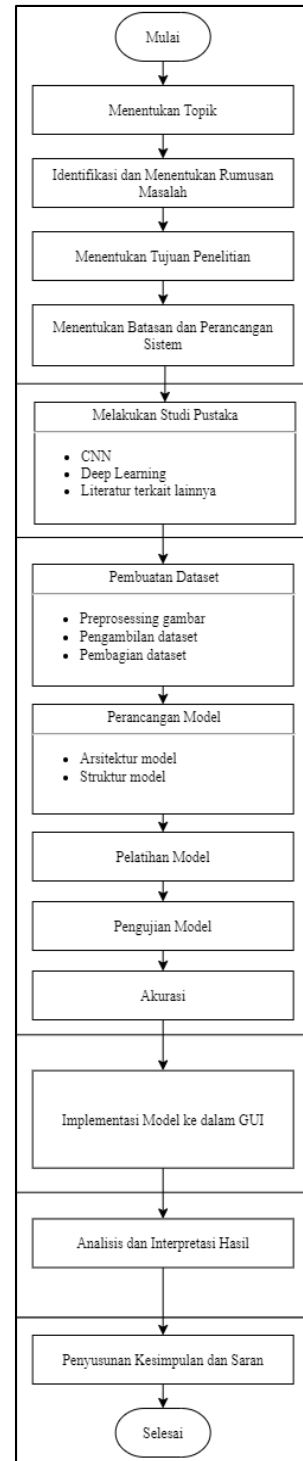
Metodologi penelitian yang dilakukan terdiri dari pengumpulan dataset, pelatihan data, pembentukan model, pengujian data perhitungan akurasi, presisi, sensitivitas, dan F1 score. Data yang digunakan dalam penelitian ini adalah citra Sistem Isyarat Bahasa Indonesia (SIBI) sebanyak 26 huruf alfabet SIBI dan 3 kelas. Data citra di ambil menggunakan *webcam*, kemudian citra dimasukkan kedalam proses *pre-processing*. Himpunan data tersebut memiliki 29 kategori bahasa isyarat. *Tools* yang digunakan dalam penelitian ini adalah *Visual Studio Code* dan *Google-Colab* dengan *Google Chrome* sebagai *web browser*. *Library* yang digunakan untuk pengolahan citra adalah *OpenCV* dan aplikasi *QtDesigner* untuk merancang serta mendesain antarmuka. Bahasa pemrograman yang digunakan yaitu Python dengan *framework* Keras dan Tensorflow. Metodologi penelitian ditunjukkan oleh Gambar 3.

Dalam penelitian ini semua data set mengalami *preprocessing image* untuk mendapatkan gambar area tangan yang akan digunakan untuk proses pelatihan dan pengujian. Tahapan *preprocessing* yang dilakukan mulai dari masukan gambar selanjutnya mengubah gambar ke dalam bentuk HSV dengan mengambil warna sesuai dengan warna kulit atau obyek gestur tangan berbentuk huruf isyarat.

Penelitian dilakukan dengan meletakkan isyarat tangan yang berbentuk huruf kedalam kotak yang merupakan area ROI untuk di *crop* dengan ukuran 200x200 dengan tujuan untuk mengurangi kompleksitas latar belakang dan menyederhanakan pengolahan.

Setelah tahap pra-pengolahan, selanjutnya pemberian label atau nama setiap citra dan pemberian label untuk setiap kelas atau folder. Citra hasil pra-pengolahan langsung di ambil sebanyak 10 gambar setiap detik dengan variasi pengambilan jarak gambar dan kamera *webcam*. Citra yang sudah diambil diberi label untuk menunjukkan objek sesuai dengan kategori masing-masing. Pemberian label kelas atau folder sesuai dengan masukan gestur yang di *capture*. Citra yang telah di *capture* kemudian di simpan dalam ekstensi png.

Langkah selanjutnya adalah pembagian dataset. Proses evaluasi model didalam *deep learning*, dibagi menjadi tiga himpunan data yaitu *train*, *validation*, dan *test*. Data *train* berfungsi untuk melatih model. Selanjutnya, model dievaluasi menggunakan data *validation*. Kemudian model diuji dengan menggunakan data *test*. Data yang terbagi menjadi tiga himpunan berfungsi untuk meningkatkan kemampuan latih dan mengurangi adanya kemungkinan *overfitting*.



Gambar 3. Langkah Penelitian

Gambar 4 menunjukkan hasil keluaran dari setiap tahap pra-pengolahan gambar.



Gambar 4. (a) Citra Masukan (b) Citra HSV (c) Citra HSV Tangan (d) Citra Grayscale (e) Citra Biner

Langkah selanjutnya adalah pembagian dataset. Proses evaluasi model didalam *deep learning*, dibagi menjadi tiga himpunan data yaitu *train*, *validation*, dan *test*. Data *train* berfungsi untuk melatih model. Selanjutnya, model dievaluasi menggunakan data *validation*. Kemudian model diuji dengan menggunakan data *test*. Data yang terbagi menjadi tiga himpunan berfungsi untuk meningkatkan kemampuan latih dan mengurangi adanya kemungkinan *overfitting*. Dalam penelitian ini menggunakan data citra sebanyak 29.000 gambar dengan ukuran 100x100 pel, masing-masing kelas memiliki 1000 gambar. Dari 29.000 gambar tersebut, 19.720 gambar digunakan untuk *pelatihan* data, 3.480 gambar digunakan untuk *validation* data dan 5.800 gambar digunakan sebagai *pengujian* data. Sehingga, rasio data yang digunakan untuk *pelatihan*, *validation* dan *test* data berturut-turut 0.68:0.12:0.20 atau dengan perbandingan data latih dan data uji sebesar 80% : 20%. Data *validasi* diambil 15% dari data latih. Perancangan struktur model CNN ditunjukkan oleh Tabel 1 berikut.

Tabel 1. Struktur Model CNN

Lapisan	Konfigurasi Lapisan
Konvolusi	32 filter, 3x3 kernel dan ReLU
Max-Pooling	2x2 kernel
Konvolusi	32 filter, 3x3 kernel dan ReLU
Max-Pooling	2x2 kernel
Konvolusi	64 filter, 3x3 kernel dan ReLU
Max-Pooling	2x2 kernel
Flatten	6400 Neuron
Fully connected	256 Neuron
Dropout	50%
Fully connected	29 Neuron
Output layer	Softmax 29 kelas

Setelah model CNN dibangun, tahap selanjutnya adalah proses *pelatihan* dengan data latih dan data uji. Dalam proses pelatihan menggunakan bobot (*weight*) model yang dapat disimpan dan dapat di *load* kembali untuk melakukan proses implementasi model untuk melakukan prediksi. Terdapat 2 parameter dalam proses pelatihan yaitu jumlah iterasi dan *batch size*. Iterasi adalah proses seluruh dataset melewati seluruh lapisan CNN sampai kembali ke awal dalam satu putaran. Jumlah iterasi adalah jumlah putaran proses pelatihan atau seluruh dataset melewati seluruh lapisan CNN dengan data latih dan data uji. Sedangkan *batch size* adalah jumlah iterasi yang disebar ke lapisan CNN. Pada penelitian ini menggunakan 5 variasi kondisi ; yaitu: 5, 10, 15, 20, dan 25 iterasi dengan nilai *batch size* model yaitu sebesar 10.

Pada penelitian sebelumnya, model yang telah dibangun memiliki variasi nilai *iterasi* dengan skema pembagian himpunan data sebesar 80% data latih dengan 20% data pengujian. Nilai akurasi tertinggi terhadap hasil pelatihan dipilih sebagai model terbaik dan akan dilakukan pengujian untuk mendapatkan kondisi optimal. Pengujian pertama dilakukan terhadap pencahayaan *webcam* dalam ruangan. Tabel

2 menunjukkan pembagian range intensitas cahaya dalam penelitian (Adi, H. A., & Candradewi, I, 2019).

Tabel 2. Tabel Range Intensitas Cahaya

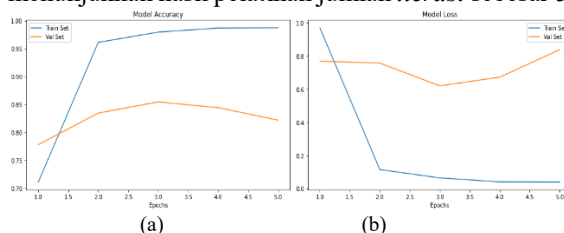
No	Intensitas Cahaya (lux)	Keterangan Ruang
1	10-25	Redup
2	26-80	Sedikit Terang
3	80-140	Terang
4	140-200	Sangat Terang

Pengujian kedua dilakukan perubahan jarak tangan terhadap *webcam* hingga model dapat mendeteksi dan memprediksi isyarat tangan. Jarak tangan terhadap *webcam* yang ambil dalam penelitian ini adalah 50 cm hingga 200 cm. Dari pengujian tersebut dapat diambil kesimpulan terkait kinerja model dalam mendapatkan kondisi optimal untuk memprediksi isyarat tangan menggunakan parameter tersebut sehingga mendapatkan keluaran berupa huruf isyarat bahasa Indonesia. Untuk mempermudah dalam proses pengujian dan proses pembuatan dataset, dibuatlah rancangan antarmuka atau *Graphic User Interface* (GUI). GUI yang dibuat didesain menggunakan software *QTDesigner* dengan library *PyQT5* sebagai *toolkit* GUI untuk bahasa Python.

3. HASIL DAN PEMBAHASAN

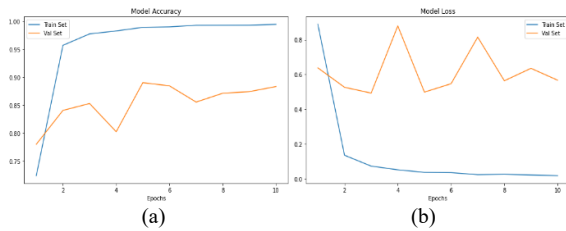
Penelitian ini menggunakan 29 citra isyarat yang terdiri dari 26 huruf alfabet dan 3 isyarat tambahan yaitu spasi, hapus, dan tidak terklasifikasi. Langkah utama penelitian dilakukan dengan melatih data pelatihan (data latih). Melatih data pelatihan memiliki tujuan untuk membentuk model yang digunakan dalam pengujian. Sebelum dilakukan pelatihan, dataset dibagi menjadi dua kelompok yaitu data latih dan data uji. Dalam penelitian ini, data citra yang dibagi menjadi 80% untuk data latih dan 20% untuk data uji. Proses pelatihan dan pengujian data dilakukan sebanyak 5 tahap, mulai dari pelatihan sebanyak 5, 10, 15, 20, dan 25 iterasi. Hal tersebut dilakukan dengan tujuan untuk melihat nilai akurasi pada setiap tahap. Model terbaik memiliki akurasi paling tinggi dan akan dipilih dalam tahap uji.

Pengujian dilakukan dengan mengatur jarak tangan terhadap *webcam* sehingga dapat mendeteksi serta memprediksi isyarat jari-jari tangan. Jarak tangan terhadap *webcam* yang diambil dalam penelitian ini adalah 50 cm hingga 200 cm. Dari pengujian dapat diambil kesimpulan terkait kinerja model untuk mendapatkan kondisi optimal dalam memprediksi isyarat tangan. Penelitian ini menggunakan nilai *batch size* sebesar 10. Gambar 5 (a) dan 5 (b) menunjukkan hasil pelatihan jumlah *iterasi* sebesar 5.



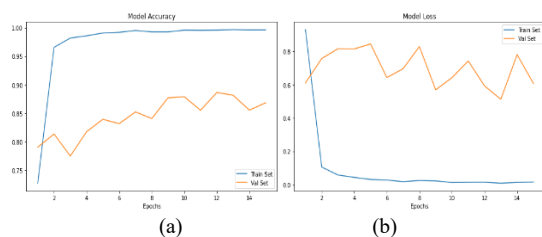
Gambar 5.(a) Grafik Model Akurasi, (b) Grafik Model Loss iterasi 5

Gambar 6 (a) dan 6 (b) menunjukkan hasil pelatihan jumlah *iterasi* sebesar 10.



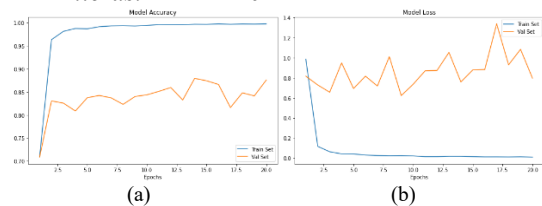
Gambar 6. (a) Grafik Model Akurasi, (b) Grafik Model Loss Iterasi 10

Gambar 7 (a) dan 7 (b) menunjukkan hasil pelatihan jumlah *iterasi* sebesar 15.



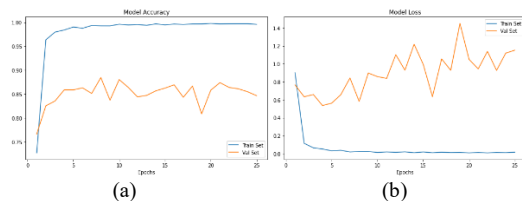
Gambar 7. (a) Grafik Model Akurasi, (b) Grafik Model Loss dengan Iterasi =15

Gambar 8. (a) dan 8 (b) menunjukkan hasil pelatihan jumlah *iterasi* sebesar 20.



Gambar 8. (a) Grafik Model Akurasi, (b) Grafik Model Loss dengan Iterasi =20

Gambar 9 (a) dan 9 (b) menunjukkan hasil pelatihan jumlah *iterasi* sebesar 25.



Gambar 9. (a) Grafik Model Akurasi, (b) Grafik Model Loss dengan Iterasi =25

Pada proses pelatihan menggunakan *iterasi* sebesar 5, 10, 15, 20, dan 25. Nilai akurasi dan nilai loss iterasi terakhir pada iterasi 15 yaitu 0.9964 dan 0.0157. Hasil ini menunjukkan model cukup tinggi dalam memprediksi menggunakan data pelatihan. Selanjutnya nilai validasi akurasi dan validasi *loss* diperoleh sebesar 0,8684 dan 0,6087. Hal ini menunjukkan model dengan *iterasi* 15 mengalami peningkatan dari sisi akurasi dibandingkan dengan model dengan *iterasi* 5 dan 10. Sedangkan pada proses pelatihan menggunakan *iterasi* 20, nilai

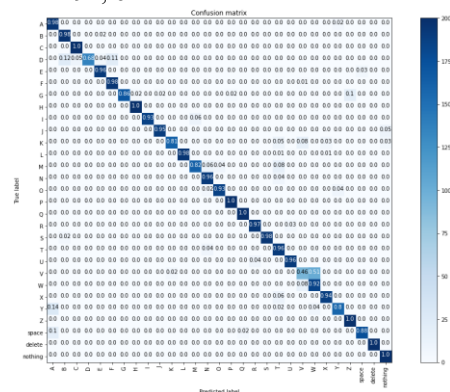
akurasi dan nilai loss iterasi terakhir yaitu 0.9959 dan 0.0165. Nilai validasi akurasi dan validasi *loss* diperoleh 1,1526 dan 0,8466. Jumlah *iterasi* berpengaruh terhadap tingkat akurasi. Jumlah *iterasi* sebesar 25 memiliki akurasi yang lebih rendah daripada iterasi 20. Dari percobaan tersebut, akurasi tertinggi diperoleh 0,92 pada iterasi 20. Tabel 3 menunjukkan hasil perbandingan jumlah *iterasi* dari hasil pelatihan dan pengujian.

Tabel 3. Hasil Uji Percobaan Berdasarkan Jumlah Iterasi

No	Jumlah Iterasi	Tahap Pelatihan		Tahap Test	
		Akurasi	Loss	Akurasi	Skor F1
1	5	0.9875	0.0388	0.87	0.87
2	10	0.9947	0.0181	0.91	0.91
3	15	0.9964	0.0157	0.89	0.89
4	20	0.9973	0.0088	0.92	0.92
5	25	0.9959	0.0165	0.90	0.89

Dari percobaan terhadap parameter jumlah *iterasi* yang telah dilakukan pada model, dapat diketahui bahwa jumlah *iterasi* berpengaruh terhadap tingkat akurasi. Dapat dilihat bahwa model dengan *iterasi* 20 memiliki akurasi yang terbaik diantara *iterasi* 5, 10, 15 dan 25 dengan memiliki akurasi sebesar 0,92. Model terbaik dengan iterasi 20 disimpan dalam file dengan ekstensi h5 yang berisi arsitektur model dan bobot dari proses *pelatihan* dan *pengujian* yang kemudian menjadi template dari aplikasi berbasis *desktop* untuk dapat melakukan klasifikasi bahasa isyarat SIBI. Model dengan iterasi 20 memiliki peningkatan akurasi dibandingkan iterasi 5, 10, 15, dan 25 sehingga dipilih sebagai model terbaik. Confusion matrix untuk iterasi 20 ditunjukkan pada Gambar 10.

Hasil akurasi *test* pada percobaan tersebut diperoleh dengan nilai sebesar 0,92 (92%) dan Skor F1 sebesar 0,91. Hasil menggunakan data test ini lebih baik dari pada model sebelumnya yang menggunakan *iterasi* 5, 10 dan 15. Untuk nilai sensitivitas terbaik diperoleh dengan nilai 1,00 pada huruf C, H, P, Q dan X. Sedangkan untuk nilai presisi terbaik diperoleh dengan nilai 1,00 pada huruf G dan I. Dengan hasil percobaan tersebut, model dapat memprediksi data uji dengan baik. Hal ini dapat ditunjukkan dengan nilai akurasi yang tinggi dan angka skor 91, 6 %.

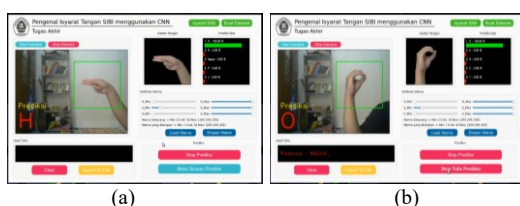


Gambar 10. Confusion Matrix untuk Iterasi =20

Didalam tahap ini, antarmuka atau GUI berfungsi memprediksi data atau gambar diluar direktori data *pelatihan*, data validasi dan data *pengujian* dengan model yang telah disimpan dalam format .h5. Implementasi dari rancangan antarmuka ditunjukkan pada Gambar 11 (a) dan (b) serta Gambar 12 (a) dan (b).



Gambar 11. (a) Kamera Webcam Aktif, (b) Hasil Kalibrasi Warna



Gambar 12. (a) Hasil Proses Prediksi ; (b) Hasil prediksi

Setelah mendapatkan model dengan akurasi yang paling baik yaitu akurasi 92% pada model dengan *iterasi* sebanyak 20, selanjutnya melakukan pengujian untuk mengetahui pengaruh dari perubahan intensitas cahaya dan jarak antara tangan dengan *webcam* di dalam ruangan berlatar belakang polos dengan nilai kalibrasi yang sama. Pengukuran tingkat kecerahan yang terdapat pada ruangan menggunakan sebuah alat bernama Lux Meter. Pada pengujian terdapat 16 kondisi pengujian dengan intensitas cahaya yang paling redup yaitu 10 lux sampai cahaya yang paling terang yaitu 200 lux, sedangkan jarak paling dekat dengan *webcam* yaitu 50 cm dan yang paling jauh 200 cm. Setiap kondisi percobaan dilakukan percobaan sebanyak 5 kali dan di ukur untuk mendapatkan nilai akurasi. Akurasi tersebut untuk mengetahui keakuratan model setelah diimplementasikan dalam aplikasi. Akurasi diukur menggunakan persamaan untuk menghitung presentase akurasi. Pengujian yang dilakukan untuk menguji model, apakah model masih mampu memprediksi dengan benar walaupun dengan kondisi pencahayaan dan jarak yang berubah-ubah. Hasil uji percobaan berdasarkan jarak dan intensitas cahaya ditunjukkan oleh Tabel 4.

Tabel 4. Hasil Percobaan Berdasar Jarak dan Intensitas Cahaya

No	Jarak (cm)	(lux)	Ruang	Jumlah	Akurasi
1	50	10-25	Redup	5	49,6%
2	50	26-80	Sedikit Terang	5	76,5%
3	50	81-140	Terang	5	82%
4	50	141-200	Sangat Terang	5	66,1%
5	100	10-25	Redup	5	96,5%
6	100	26-80	Sedikit Terang	5	100%
7	100	81-140	Terang	5	95,8%
8	100	141-200	Sangat Terang	5	97,2%
9	150	10-25	Redup	5	86,8%

10	150	26-80	Sedikit Terang	5	95,1%
11	150	81-140	Terang	5	90,3%
12	150	141-200	Sangat Terang	5	92,3%
13	200	10-25	Redup	5	58,6%
14	200	26-80	Sedikit Terang	5	75,8%
15	200	81-140	Terang	5	68,2%
16	200	141-200	Sangat Terang	5	82%

Pada Tabel 4 menunjukkan pengujian dengan kondisi cahaya redup dan jarak tangan 50 cm, diperoleh hasil akurasi yang rendah. Hal ini disebabkan masih banyak huruf isyarat yang tidak terprediksi dengan benar. Sedangkan pada percobaan dengan kondisi cahaya terang menunjukkan adanya peningkatan akurasi. Pada jarak yang sama, hasil ini lebih baik dari pada kondisi sebelumnya dengan pencahayaan lebih terang. Tabel 5 menunjukkan hasil persentase dari keberhasilan model dalam memprediksi dengan jarak tangan terhadap *webcam* 150 cm dan intensitas cahaya berkisar antara 26 sampai 80 lux adalah 95,8%. Hasil pada kondisi jarak 150 cm memiliki pencahayaan yang lebih baik dibanding dengan kondisi pencahayaan yang redup. Apabila pada kondisi pencahayaan redup mendapatkan hasil akurasi rata-rata diatas 80% maka pada kondisi pencahayaan lebih terang mendapatkan nilai akurasi rata-rata diatas 90%, hal ini membuktikan bahwa meningkatnya pencahayaan akan meningkatkan kualitas citra yang diambil sehingga proses prediksi meningkat. Berbagai perubahan intensitas cahaya dan jarak antara tangan dengan *webcam* yang telah dilakukan dapat ditunjukkan bahwa kondisi paling optimal adalah kondisi pencahayaan 26 sampai 80 lux dengan jarak kamera 100 cm memberikan hasil akurasi tertinggi 97,2%. Perbandingan metode pada hasil penelitian dengan hasil penelitian sebelumnya ditunjukkan oleh Tabel 5. Pada penelitian yang dilakukan oleh Nurhayati, dkk menggunakan citra *gesture* tangan sebagai masukan sedangkan pada penelitian yang dilakukan Adi dan Candra menggunakan metode CNN untuk mengendalikan robot beroda.

Tabel 5. Perbandingan Hasil Metode Penelitian

PERBANDINGAN	ODN, dkk (Tangan)	ADI & CANDRA (Robot)
AKURASI	97,2 %	97,5 %
PRESISI	91,96 %	97,57 %
SENSITIVITAS	91,9 %	97,5 %
SPESIVISITAS	91,96 %	99,7 %
F1 SCORE	91,9 %	97,45

Tabel 5 menunjukkan dengan menggunakan parameter uji yang sama memberikan tingkat akurasi, presisi, sensitivitas, spesivisitas, serta nilai F1 score yang berbeda. Hal ini dipengaruhi oleh beberapa faktor diantaranya; jarak obyek terhadap kamera pada kisar 50-200 cm, intensitas cahaya yang berkisar 10-200 lux.

Perbandingan hasil dari penelitian-penelitian diantaranya Adithya V., Rajesh R. (2019). Adi, H. A., & Candradewi, I. (2019), Chung, H.Y., Y. L.

Chung, dan W. F. Tsai, (2019), Danukusumo, K., Pranowo, & Maslim, M., (2017), Adithya, V., Rajesh, R., (2020), Haria, dkk, (2017), Sharma, dkk, (2020), Alam, I. F., Sarita, M. I., & Sajiah, A. M., (2019), Bheda, V., dan D. Radpour, (2017), (Lee, S. J., Chen, T., Yu, L., & Lai, C. H., 2018), dll dengan penelitian ini, belum pernah dilakukan penelitian mengenai CNN *Sequential* yang digunakan untuk mengenali jari-jari tangan secara *real time*.

4. KESIMPULAN

Analisis terhadap hasil penelitian sistem isyarat bahasa Indonesia menggunakan metode CNN *sequential* menghasilkan kesimpulan sebagai berikut:

1. Sistem yang dibangun berupa tools dapat digunakan untuk mendeteksi huruf isyarat tangan SIBI menggunakan metode *Convolutional Neural Network*.
2. Jumlah *iterasi* dapat memengaruhi nilai akurasi model. Jumlah *iterasi* paling optimal dalam percobaan ini yaitu sebesar 20 *iterasi* dengan hasil akurasi *test* sebesar 0,92 menggunakan perbandingan rasio data *pelatihan* dan data *test* 80:20.
3. Nilai akurasi ditentukan oleh posisi, gerak, jarak berkisar 50-200 cm, arah objek terhadap kamera, serta intensitas cahaya yang berkisar 10-200 lux.
4. Jarak paling optimal untuk dapat mengidentifikasi dan memprediksi dengan baik adalah jarak 100 cm dengan kecerahan cahaya 26 sampai 80 lux dengan hasil persentase keberhasilan 100%.

DAFTAR PUSTAKA

- ADI, H. A., & CANDRADEWI, I. 2019. Sistem Pengenal Isyarat Tangan Untuk Mengendalikan Gerakan Robot Beroda menggunakan Convolutional Neural Network. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 9(2), pp 193-202.
- ADITHYA V. & RAJESH R. 2019. A Deep Convolutional Neural Network Approach for Static Hand. *Third International Conference on Computing and Network Communications (CoCoNet'19)*, *Procedia Computer Science*, pp. 2353-2361
- ALAM I. F., SARITA, M. I., & SAJIAH, A. M. 2019. Implementasi Deep Learning dengan Metode Convolutional Neural Network untuk Identifikasi Objek secara Real Time Berbasis Android, *SemanTIK*, 5(2), pp. 237-244
- ANDRIAN, M. Y., PURWANTO, D., & MARDIYANTO, R. 2017. Penerjemahan Bahasa Isyarat Indonesia Menggunakan Kamera pada Telepon Genggam Android. *Jurnal Teknik ITS*, 6(1), pp. 180-183.
- BAKTI, M.B.S., & YULIANA, M.P. 2019. Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network. *Seminar Nasional Inovasi Teknologi*, pp. 11-16.
- BHEDA, V., & D. RADPOUR. 2017. *Using Deep Convolutional Networks for Gesture Recognition in American Sign Language*. Retrieved from <https://www.researchgate.net/publication/320487023>.
- CHUNG, H.Y., Y. L. CHUNG, & W. F. TSAI. 2019. An efficient hand gesture recognition system based on deep CNN. *Prosiding of IEEE International Conference on Industrial Technology*, pp. 853-858.
- DANUKUSUMO, K., PRANOWO, & MASLIM, M. 2017. Indonesia ancient temple classification using convolutional neural network, pp. 50-54.
- DARMATASIA. 2021. Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Gradient-Convolutional Neural Network. *Jurnal Instek*, pp. 56-65.
- DENG., L & D. Yu. 2013. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4), pp. 197-387.
- HARIA, A., ARCHANASRI S, NIVEDHITHA A, SHRISTI P, & JYOTHI S.N. 2017. Hand Gesture Recognition for Human Computer Interaction. *7th International Conference on Advances in Computing & Communications, ICACC-2017*, *Procedia Computer Science*, pp. 367-374.
- HURLBUT, H. M. 2014. *The Signed Languages of Indonesia : An Enigma*. Retrieved 2020, from <https://www.sil.org/resources/publications/entry/58160>.
- KOH, J.H., & S. H. A. ALI. 2017. ASL fingerspelling recognition system for interactive learning and education purpose. *Journal of Telecommunication Electronic and Computer Engineering*, 9 (3-7), pp. 43-47.
- LEE, S. J., CHEN, T., YU, L., & LAI, C. H. 2018. Image Classification Based on the Boost Convolutional Neural Network. *IEEE Access*, 6(14), pp. 12755-12768.
- MENDIKBUD. 1994. *Pembakuan Sistem Isyarat Bahasa Indonesia Bagi Kaum Tuna Rungu . no. 0161/U/1994*. Retrieved 2020, from <https://peraturan.bkpm.go.id/jdih/front/index/114/20>.
- MOHRI, A., M., ROSTAMIZADEH, A., & TALWALKAR. 2018. *Foundations of machine learning*. MIT Press.
- PAJAR, T.Y., D. PURWANTO, & H. KUSUMA. 2018. Pengenalan Bahasa Isyarat Tangan Menggunakan Depth Image. *Jurnal Teknik ITS*, 7(1).

- THARWAT, A. 2018. Classification assessment methods. *Applied Computing and Informatics*, pp. 1-13.
- TRAORE, B., KAMSU-FOGUEM, & TANGARA, F. 2018. Deep convolution neural network for image recognition. *Ecological Information*, 48, pp. 257 – 268.
- YANUR, A. 2018. <https://machinelearning.mipa.ugm.ac.id/2018/06/25/fullyconnected-layer-cnn-dan-Implementasinya/>
[Di akses 2 Januari 2020]