

PERANCANGAN PROGRAM PENGENALAN ISYARAT TANGAN DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

Fauzan Akbar^{*)}, Aris Triwiyatno dan Achmad Hidayatno

Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)} E-mail: fauzan19akbar@gmail.com

Abstrak

Perkembangan teknologi dalam bidang pemrosesan sinyal mengalami kemajuan yang pesat. Kemajuan teknologi ini dapat dimanfaatkan untuk menunjang pemberian instruksi secara real-time. Perancangan ini bertujuan untuk melakukan penyederhanaan pemberian instruksi dalam kehidupan sehari-hari. Pengenalan isyarat menggunakan metode Convolutional Neural Network (CNN) dalam proses klasifikasinya. Sistem pengenalan diawali dengan proses pelatihan data latih sebagai referensi. Data latih kemudian akan dibandingkan dengan citra masukan dengan operasi konvolusi. Kemudian citra masukan akan diklasifikasikan berdasarkan data latih yang telah dihimpun sebelumnya. Hasil pengujian data latih sistem mencapai nilai 92% dengan metode random sampling pada software Matlab. Sedangkan pengujian klasifikasi citra masukan melalui webcam mencapai nilai 100%.

Kata Kunci: *Deep Learning, Convolutional Neural Network, AlexNet.*

Abstract

Convolutional Neural Network is one of the Deep Learning Algorithms. CNN itself is developed from Multilayer Perceptron (MLP) method. CNN and MLP are algorithms that focused on processing data in two dimensions form, such as pictures or sounds. CNN is made with the principle of translation invariance. That means CNN is able to recognize objects at various possible positions. There are 150 sign language images that are classified using Alexnet in this system. Alexnet is Krizhevsky's work at developing CNN method as a classifier. CNN architecture developed by Alex has eight feature extraction layers. The layer consists of five convolution layers and three pooling layers. In its classification layer, Alexnet has two fully connected layers, each of them has 4096 neurons. At the end of the layer, there are classifications into 5 categories using softmax activation. The average accuracy of the classification results even reaches 100%.

Keywords: *Deep Learning, Convolutional Neural Network, AlexNet*

1. Pendahuluan

Kemajuan teknologi dalam bidang pemrosesan sinyal data mengalami kemajuan yang pesat dan sangat relevan untuk segera diaplikasikan pada kehidupan sehari-hari. Sinyal data dapat diterima oleh perangkat pintar untuk diproses menjadi instruksi tertentu. Kemungkinan teknologi ini dapat dimanfaatkan untuk penyederhanaan pemberian instruksi-instruksi sederhana dalam kehidupan sehari-hari.

Data yang masif atau biasa disebut *big data* memiliki ciri berukuran besar, variatif, pertumbuhannya yang cepat, dan mungkin tidak terstruktur akan susah diolah dengan pendekatan konvensional [1]. Kelemahan lain pendekatan konvensional adalah keterbatasan dalam mengolah target kelas yang banyak dan hanya dapat mengenali objek yang berada di tengah citra atau tidak dapat mengenali objek dalam citra pada berbagai macam posisi yang mungkin (*translation invariance*). Selain variasi posisi objek, ada

juga kendala lain seperti rotasi objek dan perbedaan ukuran (*scaling*). *Machine Learning* [2] bisa juga mempelajari prinsip *translation invariance*, tetapi memerlukan jauh lebih banyak parameter dibanding *deep learning* yang memang dibuat dengan prinsip *translation invariance*.

Deep Learning adalah cabang ilmu *machine learning* berbasis Jaringan Saraf Tiruan (JST) atau bisa dikatakan sebagai perkembangan dari JST [3]. Pada *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara. *Convolutional Neural Network* (CNN/ConvNet) [4],[5] adalah salah satu algoritma *Deep Learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN dapat belajar langsung dari citra sehingga mengurangi beban dari pemrograman.

Selain itu, metode *machine learning* konvensional hanya mengandalkan CPU dan RAM dalam proses komputasi,

sehingga spesifikasi CPU dan RAM menentukan kecepatan komputasi. Sedangkan metode *Deep Learning* [6], selain menggunakan CPU dan RAM dalam proses komputasi, metode ini juga memanfaatkan kemampuan GPU sehingga proses komputasi data yang besar dapat berlangsung lebih cepat [3].

Sinyal citra merupakan salah satu jenis sinyal yang dapat ditangkap oleh perangkat digital secara detail dan cepat untuk kemudian diproses menjadi instruksi tertentu [7]. Kemajuan inilah yang dapat dimanfaatkan untuk melakukan pengenalan isyarat secara *real-time*. Selanjutnya pengenalan isyarat dapat diinputkan kembali untuk diaktualisasikan dalam bentuk instruksi tertentu pada aktuator. Hal ini merupakan bentuk penyederhanaan instruksi karena pengenalan isyarat tubuh sangat dekat dan alamiah bagi manusia.

Perangkat sehari-hari membutuhkan pengaturan sederhana untuk disesuaikan dengan kondisi yang diinginkan. Instruksi sederhana tersebut dapat dialihfungsikan untuk mengatur sebuah *plant* dengan proses *interfacing* antara identifikasi isyarat dan aktuator.

Penelitian sebelumnya oleh Sarirotul Ilahiyah [8] membahas tentang klasifikasi citra menggunakan metode *Convolutional Neural Network* (CNN) [9],[10]. Pada penelitian ini dilakukan klasifikasi identifikasi jenis tumbuhan. Penelitian ini memiliki kekurangan berupa *input* citra yang dilakukan manual. Penelitian tentang pengaturan kondisi ruang juga dilakukan oleh Salsabila [11] dengan bahasan klasifikasi dengan metode CNN. Pada penelitian ini citra yang diklasifikasikan adalah jenis wayang [12]. Penelitian ini memiliki kekurangan berupa proses pelatihan data yang lama karena kebutuhan data latih yang besar untuk wayang yang memiliki kemiripan satu dengan yang lainnya.

Penelitian lain tentang pengenalan isyarat dilakukan oleh Artha Gilang Saputra [13] dengan judul “Analisis Penerapan Metode *Convex Hull* dan *Convexity Defects* untuk Pengenalan Isyarat Tangan”. Pada penelitian ini dilakukan klasifikasi isyarat menggunakan metode *Hull and Convexity Defects*. Penelitian ini memiliki keandalan dalam proses klasifikasi. Namun penelitian ini tidak menggunakan hasil klasifikasi tersebut sebagai instruksi apapun. Penelitian terkait pengenalan isyarat juga dilakukan oleh Abdul Aziz Al Anshori [14] dengan judul “Perancangan dan Penerapan Pengenalan Pola Tangan pada Sistem *Home Automation* dengan *Haar-Cascade Classifier*”. Pada penelitian ini memiliki kemiripan dengan penelitian penulis. Perbedaan dari penelitian ini hanyalah metode yang digunakan oleh penulis.

Penelitian ini merupakan pengembangan dari penelitian terkait pengolahan sinyal citra dan pengendalian kondisi lingkungan. Pengolahan citra yang diangkat didasari oleh jurnal berjudul “Penerapan *Deep Learning* Menggunakan

Convolutional Neural Network Untuk Klasifikasi Citra Wayang Punakawan” [8], dan sistem pengendalian didasari oleh penelitian sebelumnya dengan judul “Perancangan Sistem Kontrol Suhu Dan Kelembaban Pada *Chamber Uji Isolator* Menggunakan Metode Kontrol PI” dengan memfokuskan pembahasan pada pengendalian suhu dan kelembaban ruangan [11].

2. Metode Penelitian

2.1. Pengumpulan Data Latih

Pengumpulan data citra dilakukan dengan cara mengumpulkan citra secara terpisah sesuai dengan kelasnya. Didapatkan 30 data latih untuk masing-masing kelas. Data latih [15] tersebut kemudian akan disortir dan akan diuji keakuratannya secara program dan secara manual.

Citra yang sudah dikumpulkan sesuai dengan kelasnya kemudian diatur dimensinya. Pengaturan dimensi bertujuan untuk penyesuaian citra dengan kebutuhan pada program. Dimensi yang dikehendaki adalah sebesar 227x227 piksel. Setelah data latih diatur dimensinya, kumpulan data latih dikumpulkan pada direktori tertentu secara terpisah untuk kemudian diakses oleh program saat proses pelatihan data latih. Gambar 1 merupakan data latih dari empat kelas yang akan dijadikan referensi pada sistem.



Gambar 1. Data Latih

2.1.1. Data Latih Isyarat Satu

Isyarat satu yang akan dilatih merupakan bentuk tangan mengepal yang akan digunakan sebagai isyarat masukan. Gambar 2 dan Gambar 3 merupakan ilustrasi isyarat satu yang akan digunakan dalam sistem dan data latih yang digunakan untuk isyarat satu



Gambar 2. Isyarat satu

Kumpulan data latih isyarat satu digunakan sebagai referensi masukan yang diambil melalui *webcam*. Hasil data latih akan menjadi referensi yang akan dibandingkan dengan referensi isyarat lain dan ciri yang paling serupa akan diklasifikasikan sebagai referensi isyarat termirip



Gambar 3. Data latih isyarat satu

2.1.2. Data Latih Isyarat Dua

Isyarat dua yang akan dilatih merupakan bentuk tangan menunjuk satu jari yang akan digunakan sebagai isyarat masukan. Gambar 4 dan Gambar 5 merupakan ilustrasi isyarat dua yang akan digunakan dalam sistem dan kumpulan data latih untuk isyarat dua.



Gambar 4. Isyarat dua



Gambar 5. Data latih isyarat dua

Kumpulan data latih isyarat dua digunakan sebagai referensi masukan yang diambil melalui *webcam*. Hasil data latih akan menjadi referensi yang akan dibandingkan

dengan referensi isyarat lain dan ciri yang paling serupa akan diklasifikasikan sebagai referensi isyarat termirip.

2.1.3. Data Latih Isyarat Tiga

Isyarat tiga yang akan dilatih merupakan bentuk tangan mengacungkan jari telunjuk dan jari kelingking yang akan digunakan sebagai isyarat masukan. Gambar 6 dan Gambar 7 merupakan ilustrasi isyarat tiga yang akan digunakan dalam sistem dan kumpulan data latih yang akan digunakan untuk isyarat tiga.



Gambar 6. Isyarat tiga



Gambar 7. Data latih isyarat tiga

Kumpulan data latih isyarat tiga digunakan sebagai referensi masukan yang diambil melalui *webcam*. Hasil data latih akan menjadi referensi yang akan dibandingkan dengan referensi isyarat lain dan ciri yang paling serupa akan diklasifikasikan sebagai referensi isyarat termirip.

2.1.4. Data Latih Isyarat Empat

Isyarat empat yang akan dilatih merupakan bentuk tangan membuka lima jari yang akan digunakan sebagai isyarat masukan. Gambar 8 dan Gambar 9 merupakan ilustrasi isyarat satu yang akan digunakan dalam sistem dan kumpulan data latih isyarat empat



Gambar 8. Isyarat empat



Gambar 9. Data latih isyarat empat

Kumpulan data latih isyarat empat digunakan sebagai referensi masukan yang diambil melalui *webcam*. Hasil data latih akan menjadi referensi yang akan dibandingkan dengan referensi isyarat lain dan ciri yang paling serupa akan diklasifikasikan sebagai referensi isyarat termirip.

2.2. Pelatihan Data

Pelatihan data latih dilakukan untuk proses ekstraksi ciri dari masing-masing kelas yang akan diklasifikasikan. Pelatihan data melibatkan seluruh gambar yang ada pada direktori. Berikut merupakan *source code* untuk melakukan pelatihan data

```
SumberGambar=imageDatastore('kamera', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

Dapat dilihat pada *source code* di atas, instruksi tersebut adalah pengambilan gambar pada direktori yang sudah ditentukan. Pengambilan gambar dilakukan ke seluruh isi dari direktori dengan keterangan dari gambar sesuai dari nama folder pada direktori yang telah ditentukan. Fungsi “ImageDatastore” merupakan fungsi penentuan letak dari direktori data latih. “IncludeSubfolders” merupakan fungsi pemanggilan folder di dalam folder direktori

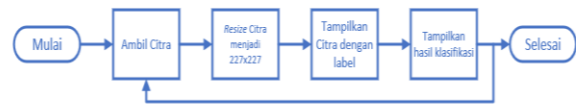
2.3. Pengujian Data

Pengujian data adalah proses penentuan keandalan dari suatu data latih. Pada proses ini dilakukan random sampling dari data latih yang sudah ditentukan. Sistematis pengujian adalah penggunaan 80% citra sebagai data latih dan 20% lainnya digunakan sebagai data uji dari masing-masing kelas yang ada. Perintah “splitEachLabel” merupakan perintah pembagian kelas sesuai masing-masing folder yang terdapat pada direktori. Berikut merupakan proses pelatihan, yang dilakukan pada *software* Matlab..

2.4. Klasifikasi Masukan

Pada tahap ini dilakukan perbandingan antara data latih yang sudah diuji dengan gambar masukan dari *webcam*.

Hasil klasifikasi merupakan perbandingan termirip antara gambar masukan dengan ciri data latih yang telah ditetapkan. Gambar 10 merupakan diagram alir program klasifikasi, diikuti senarai program proses klasifikasi



Gambar 10 Diagram alir proses klasifikasi

```
predictedLabels=classify(Klasifikasi, GambarUji);

while true

gambar=kamera.snapshot;

gambar=imresize(gambar, [227,227]);

label=classify(Klasifikasi, gambar);

image(gambar);

title(char(label));

drawnow;

R=sprintf('%s adalah hasil dari klasifikasi.\n', label);

disp(R)

pause(0.1)
```

Dapat dilihat pada senarai program di atas, proses klasifikasi citra diawali dengan pengambilan gambar pada perangkat *webcam* dengan fungsi “snapshot”. Hasil gambar kemudian diresize dengan perintah “imresize” menjadi ukuran 227x227, kemudian gambar akan diberi label dengan perintah “title(char(Label))” sesuai dengan hasil klasifikasi antara citra masukan dan data latih yang tersedia.

2.5. Komunikasi Data

2.5.1. Inisiasi Mikrokontroler

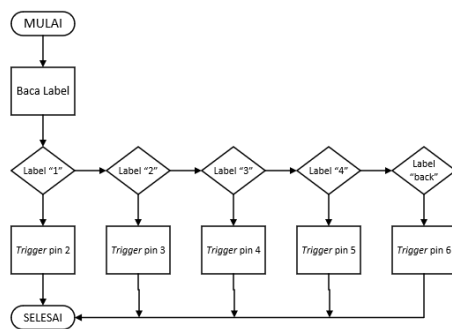
Sebelum melakukan komunikasi data perangkat perlu dikenali dengan cara penginisiasian perangkat yang digunakan. Hal ini bertujuan untuk memberikan akses pada Matlab untuk berkomunikasi dengan mikrokontroler. Berikut merupakan *source code* untuk melakukan inisiasi (pengenalan) mikrokontroler.

```
a= arduino('com5', 'uno');
```

```
[GambarLatih, GambarUji]=splitEachLabel(SumberGambar, 0.8, 'randomize');
```

2.5.2. Komunikasi Digital

Setelah dilakukan inisiasi mikrokontroler, software kini dapat melakukan pengiriman data berupa sinyal digital kepada mikrokontroler. Pengiriman data ini berupa trigger port yang ada pada mikrokontroler. Selanjutnya perintah trigger tersebut dapat kemudian diolah menjadi perintah yang ditentukan selanjutnya. Gambar 11 merupakan diagram alir proses komunikasi digital, diikuti senarai program untuk komunikasi sederhana antara Matlab dan mikrokontroler.



Gambar 11 Diagram alir komunikasi digital

Dapat dilihat pada senarai program diatas, sinyal digital akan dikirimkan sesuai hasil klasifikasi yang ditentukan. Perintah “writeDigitalPin” adalah perintah untuk melakukan *trigger pada pin*. Saat *input* citra menunjukkan isyarat satu (isyarat satu dinotasikan “1”), maka sinyal akan dikirimkan dan men-*trigger* port digital 2. Saat *input* citra menunjukkan isyarat dua (isyarat dua dinotasikan “2”), maka sinyal akan dikirimkan dan men-*trigger* port digital 3. Saat *input* citra menunjukkan isyarat tiga (isyarat tiga dinotasikan “3”), maka sinyal akan dikirimkan dan men-*trigger* port digital 4. Saat *input* citra menunjukkan isyarat empat (isyarat empat dinotasikan “4”), maka sinyal akan dikirimkan dan men-*trigger* port digital 5. Saat *input* citra menunjukkan isyarat background (background dinotasikan dengan “back”), maka sinyal akan dikirimkan ke port digital 6.

```

if label == '1'
    writeDigitalPin(a, 'D2', 1);
    writeDigitalPin(a, 'D3', 0);
    writeDigitalPin(a, 'D4', 0);
    writeDigitalPin(a, 'D5', 0);
    writeDigitalPin(a, 'D6', 0);
    pause(0.5);
  
```

```

elseif label == '2'
    writeDigitalPin(a, 'D2', 0);
    writeDigitalPin(a, 'D3', 1);
    writeDigitalPin(a, 'D4', 0);
    writeDigitalPin(a, 'D5', 0);
    writeDigitalPin(a, 'D6', 0);
    pause(0.5);
  
```

```

elseif label == '3'
    writeDigitalPin(a, 'D2', 0);
    writeDigitalPin(a, 'D3', 0);
    writeDigitalPin(a, 'D4', 1);
    writeDigitalPin(a, 'D5', 0);
    writeDigitalPin(a, 'D6', 0);
    pause(0.5);
  
```

```

elseif label=='4'
    writeDigitalPin(a, 'D2', 0);
    writeDigitalPin(a, 'D3', 0);
    writeDigitalPin(a, 'D4', 0);
    writeDigitalPin(a, 'D5', 1);
    writeDigitalPin(a, 'D6', 0);
    pause(0.5);
  
```

```

elseif label=='back'
    writeDigitalPin(a, 'D2', 0);
    writeDigitalPin(a, 'D3', 0);
    writeDigitalPin(a, 'D4', 0);
    writeDigitalPin(a, 'D5', 0);
    writeDigitalPin(a, 'D6', 1);
    pause(0.5);
  
```

end

3. Hasil dan Pembahasan

3.1. Pengujian Data Latih

Pengujian data adalah proses penentuan keandalan dari suatu data latih. Pada proses ini dilakukan random sampling dari data latih yang sudah ditentukan. Sistematika pengujian adalah penggunaan 80% citra sebagai data latih dan 20% lainnya digunakan sebagai data uji dari masing-masing kelas yang ada. Berikut merupakan hasil uji data latih yang dilakukan pada *software* Matlab. Gambar 12 menunjukkan hasil pengujian data latih pada Matlab

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:16	21.88%	1.9187	0.0010
20	20	00:00:55	100.00%	0.0013	0.0010

accuracy =
0.9200

Gambar 12. Command window hasil pengujian

Dapat dilihat Gambar 12 nilai akurasi citra mencapai nilai 92%. Nilai tersebut didapatkan dari proses random sampling dengan perbandingan 80:20 (80% citra dijadikan data latih. 20% citra dijadikan data uji).

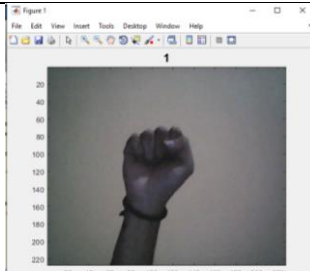
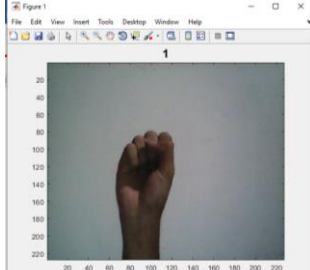
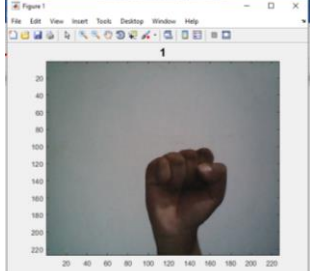
3.2. Hasil Klasifikasi

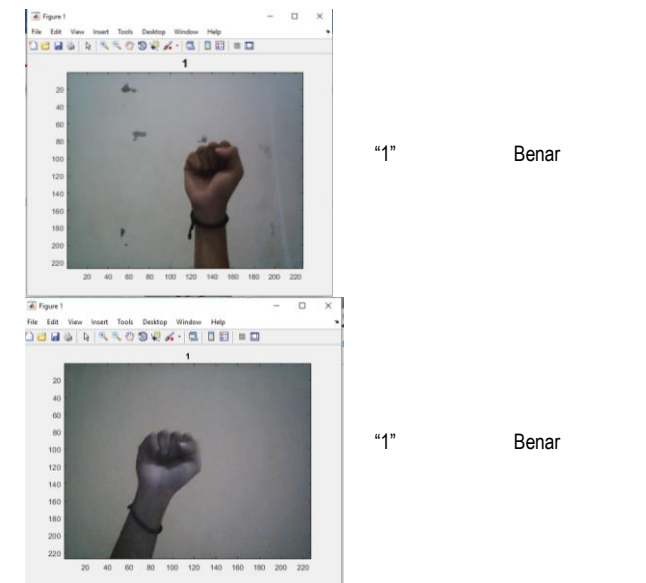
Setelah program diuji, didapatkan hasil klasifikasi citra masukan yang didapat melalui *webcam*. Hasil klasifikasi tersebut akan dikirimkan dalam bentuk sinyal digital ke mikrokontroler untuk kemudian digunakan sebagai perintah pada *plant*. Berikut merupakan hasil klasifikasi

3.2.1. Isyarat Satu

Berikut merupakan hasil klasifikasi citra isyarat satu. Isyarat satu memiliki ciri seperti kepalan tangan. Citra masukan yang memiliki ciri seperti kepalan tangan akan diklasifikasikan sebagai isyarat satu. Tabel 1 merupakan tabel hasil pengujian dari klasifikasi isyarat satu.

Tabel 1. Hasil klasifikasi isyarat satu

Citra	Hasil Klasifikasi	Keterangan
	"1"	Benar
	"1"	Benar
	"1"	Benar

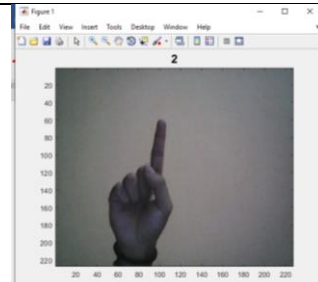
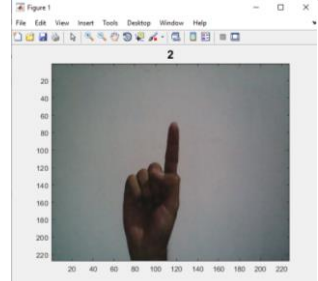


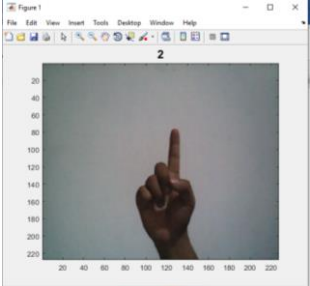
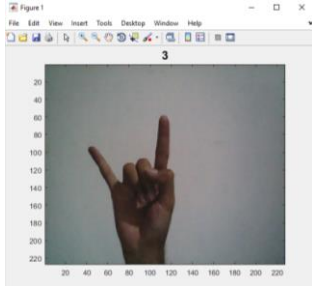
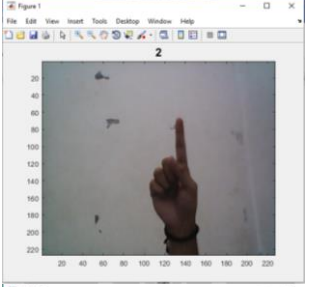
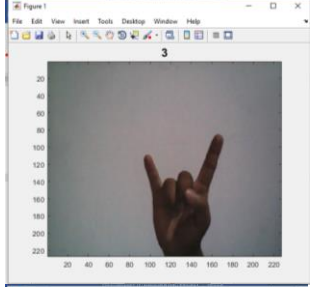
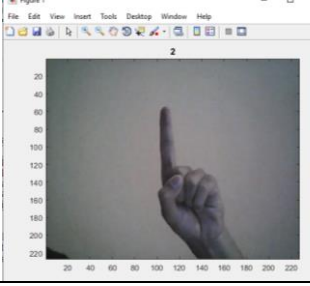
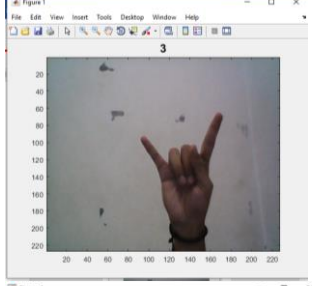
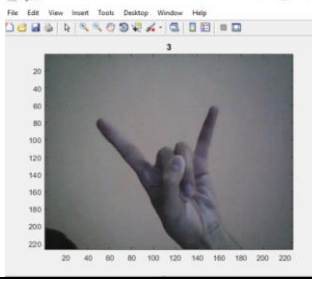
Dapat dilihat dari Tabel 1 bahwa hasil pengujian klasifikasi menunjukkan isyarat dapat diklasifikasikan secara sempurna tanpa kesalahan. Dapat disimpulkan ekstraksi ciri pada isyarat satu berjalan dengan baik dan data latih sudah memiliki keandalan yang baik.

3.2.2. Isyarat Dua

Berikut merupakan hasil klasifikasi citra isyarat dua. Isyarat dua memiliki ciri seperti menunjukkan jari telunjuk. Citra masukan yang memiliki ciri seperti tangan menunjuk akan diklasifikasikan sebagai isyarat dua. Tabel 2 menunjukkan tabel hasil klasifikasi isyarat dua.

Tabel 2. Hasil kalsifikasi isyarat dua

Citra	Hasil Klasifikasi	Keterangan
	"2"	Benar
	"2"	Benar

	"2"	Benar		"3"	Benar
	"2"	Benar		"3"	Benar
	"2"	Benar		"3"	Benar
				"3"	Benar

Dapat dilihat dari Tabel 2 bahwa hasil pengujian klasifikasi menunjukkan isyarat dapat diklasifikasikan secara sempurna tanpa kesalahan. Dapat disimpulkan ekstraksi ciri pada isyarat dua berjalan dengan baik dan data latih sudah memiliki keandalan yang baik.

3.2.3. Isyarat Tiga

Berikut merupakan hasil klasifikasi citra isyarat tiga. Isyarat tiga memiliki ciri seperti tangan kepal yang menunjukkan jari telunjuk dan jari kelingking. Citra masukan yang memiliki ciri seperti kepalan dengan jari telunjuk dan jari kelingking menunjuk akan diklasifikasikan sebagai isyarat tiga. Tabel 3 merupakan tabel hasil klasifikasi isyarat tiga yang diberikan pada webcam.

Tabel 3. Hasil klasifikasi isyarat tiga

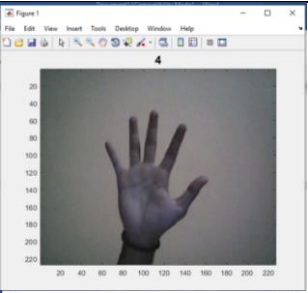
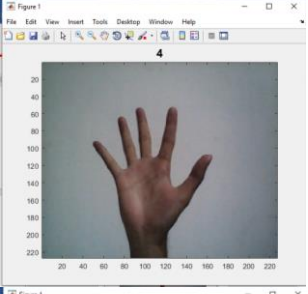
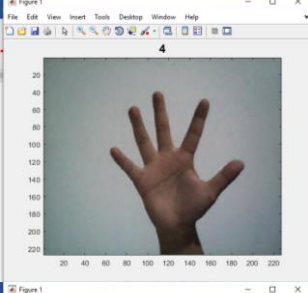
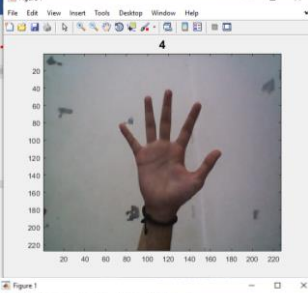
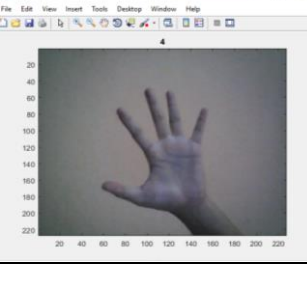
Citra	Hasil Klasifikasi	Keterangan
	"3"	Benar

Dapat dilihat dari Tabel 3 bahwa hasil pengujian klasifikasi menunjukkan isyarat dapat diklasifikasikan secara sempurna tanpa kesalahan. Dapat disimpulkan ekstraksi ciri pada isyarat tiga berjalan dengan baik dan data latih sudah memiliki keandalan yang baik. Pada citra masukan isyarat tiga terkadang mengalami kesalahan keputusan karena memiliki kemiripan dengan isyarat dua dan isyarat empat.

3.2.4. Isyarat Empat

Berikut merupakan hasil klasifikasi citra isyarat empat. Isyarat empat memiliki ciri seperti tangan yang membuka semua jari. Citra masukan yang memiliki ciri seperti tangan membuka akan diklasifikasikan sebagai isyarat empat. Tabel 4 menunjukkan tabel hasil klasifikasi isyarat empat.

Tabel 4. Hasil klasifikasi isyarat empat

Citra	Hasil Klasifikasi	Keterangan
	"4"	Benar
	"4"	Benar
	"4"	Benar
	"4"	Benar
	"4"	Benar

Dapat dilihat dari Tabel 4 bahwa hasil pengujian klasifikasi menunjukkan isyarat dapat diklasifikasikan secara sempurna tanpa kesalahan. Dapat disimpulkan ekstraksi ciri pada isyarat empat berjalan dengan baik dan data latih sudah memiliki keandalan yang baik.

3.3. Komunikasi Data

3.3.1. Inisiasi Mikrokontroler

Sebelum melakukan komunikasi data perangkat perlu dikenali dengan cara penginisiasian perangkat yang digunakan. Hal ini bertujuan untuk memberikan akses pada Matlab untuk berkomunikasi dengan mikrokontroler. Gambar 13 merupakan hasil inisiasi mikrokontroler

```
Arduino Uno detected.
This device is ready for use with MATLAB Support Package for Arduino Hardware.
This device is ready for use with Simulink Support Package for Arduino Hardware.
```



Gambar 13. Hasil inisiasi mikrokontroler

Pada Gambar 13 dapat dilihat keterangan pada *command window* akan muncul saat mikrokontroler diinisiasi. Hal ini menunjukkan jika perangkat mikrokontroler dapat digunakan dan dapat menerima sinyal dari Matlab.





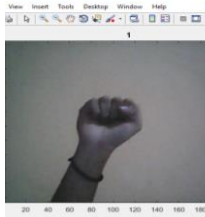
3.3.2. Komunikasi Digital

Setelah dilakukan inisiasi mikrokontroler, *software* kini dapat melakukan pengiriman data berupa sinyal digital kepada mikrokontroler. Pengiriman data ini berupa *trigger port* yang ada pada mikrokontroler. Selanjutnya perintah *trigger* tersebut dapat kemudian diolah menjadi perintah yang ditentukan selanjutnya. Tabel 5 merupakan hasil komunikasi digital pada isyarat satu.

Pada Tabel 5 dapat dilihat hasil klasifikasi pada Matlab dikirimkan ke mikrokontroler dan terbaca sesuai dengan perintah dengan sempurna tanpa kesalahan.

Tabel 5. Hasil komunikasi digital isyarat satu

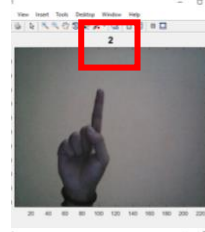


Citra	Klasifikasi	Hasil baca serial monitor	Keterangan
	"1"	Kondisi 1. Kondisi 1. Kondisi 1.	Berhasil
	"1"	Kondisi 1. Kondisi 1. Kondisi 1. Kondisi 1. Kondisi 1. Kondisi 1. Kondisi 1.	Berhasil

	"1"	Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1	Berhasil		"2"	Background Kondisi 2H	Berhasil
	"1"	Kondisi 1 Kondisi 1 Kondisi 1 Kondisi 1	Berhasil		"2"	Kondisi 2H Kondisi 2H	Berhasil
	"1"	Kondisi 1 Kondisi 1	Berhasil				

Perintah ditampilkan pada *serial monitor* di *software* Arduino. Berdasarkan beberapa pengujian perintah dapat disampaikan ke mikrokontroler.

Selanjutnya dilakukan pengujian pada isyarat dua. Tabel 6 menunjukkan hasil komunikasi digital pada isyarat dua.

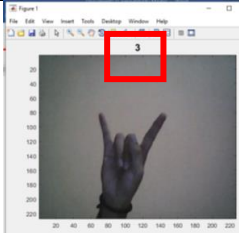
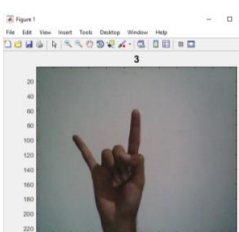
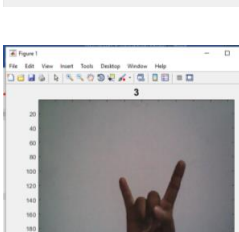
Tabel 6. Hasil komunikasi digital isyarat dua

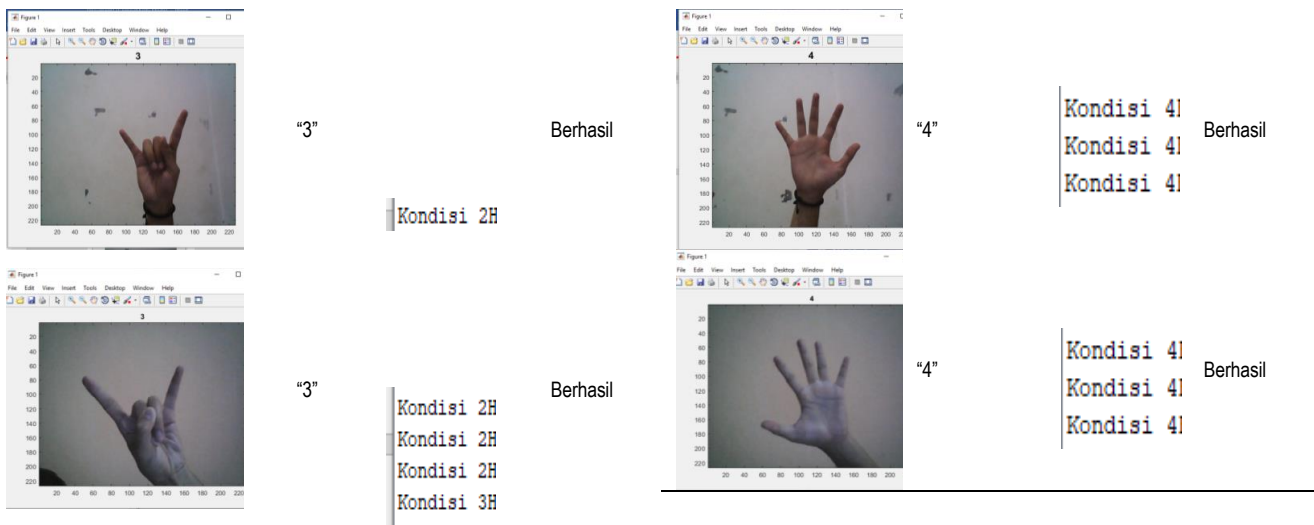
Citra	Klasifikasi	Hasil baca <i>serial monitor</i>	Keterangan
	"2"	Background Kondisi 2H Kondisi 2H	Berhasil
	"2"	Kondisi 2H	Berhasil
	"2"	Kondisi 2H Kondisi 2H	Berhasil

Pada Tabel 6 dapat dilihat hasil klasifikasi pada Matlab dikirimkan ke mikrokontroler dan terbaca sesuai dengan perintah dengan sempurna tanpa kesalahan. Perintah ditampilkan pada *serial monitor* di *software* Arduino. Berdasarkan beberapa pengujian perintah dapat disampaikan ke mikrokontroler secara sempurna.

Selanjutnya dilakukan pengujian pada isyarat tiga. Tabel 7 menunjukkan hasil komunikasi digital pada isyarat tiga.

Tabel 7. Hasil komunikasi digital isyarat tiga

Citra	Klasifikasi	Hasil baca <i>serial monitor</i>	Keterangan
	"3"	Kondisi 2H Kondisi 2H	Berhasil
	"3"	Kondisi 2H Kondisi 2H Kondisi 2H Kondisi 3H	Berhasil
	"3"	Kondisi 2H Kondisi 2H Kondisi 2H Kondisi 3H	Berhasil



Pada Tabel 7 dapat dilihat hasil klasifikasi pada Matlab dikirimkan ke mikrokontroler dan terbaca sesuai dengan perintah dengan sempurna tanpa kesalahan. Perintah ditampilkan pada *serial monitor* di *software* Arduino. Berdasarkan beberapa pengujian perintah dapat disampaikan ke mikrokontroler secara sempurna.

Selanjutnya dilakukan pengujian pada isyarat empat. Tabel 8 menunjukkan hasil komunikasi digital pada isyarat empat.

Tabel 8. Hasil komunikasi digital isyarat empat

Citra	Klasifikasi	Hasil baca serial monitor	Keterangan
	"4"	Kondisi 4I Kondisi 4I Kondisi 4I	Berhasil
	"4"	Kondisi 4I Kondisi 4I Kondisi 4I	Berhasil
	"4"	Kondisi 4I Kondisi 4I Kondisi 4I	Berhasil

Pada Tabel 8 dapat dilihat hasil klasifikasi pada Matlab dikirimkan ke mikrokontroler dan terbaca sesuai dengan perintah dengan sempurna tanpa kesalahan. Perintah ditampilkan pada *serial monitor* di *software* Arduino. Berdasarkan beberapa pengujian, perintah dapat disampaikan ke mikrokontroler secara sempurna.

Dari hasil pengujian yang dilakukan pada setiap kondisi, dapat disimpulkan komunikasi digital berjalan dengan baik tanpa ada *error*. Selanjutnya sinyal digital yang dikirimkan akan menjadi instruksi pengendalian *plant* yang sudah ditetapkan sebelumnya.

4. Kesimpulan

Berdasarkan hasil pengujian data latih yang sudah dirancang, hasil pengujian data latih dapat mencapai angka 92% pada akurasi klasifikasi isyarat. Pada pengujian citra masukan, didapatkan hasil akurasi 100%. Pada pengujian komunikasi digital, didapatkan hasil 100%. Seluruh hasil klasifikasi dapat dikomunikasikan dengan baik ke mikrokontroler. Komunikasi digital yang dilakukan untuk menghubungkan *software* dengan *plant* dapat berjalan secara *real-time*. Metode CNN andal dalam melakukan klasifikasi isyarat karena metode ini didesain untuk proses klasifikasi skala besar. Untuk meningkatkan kemampuan sistem, disarankan melakukan penambahan data latih dan penggunaan *hardware* yang lebih mumpuni untuk menunjang performa yang lebih baik. Pemutakhiran sistem berupa penambahan deteksi

Referensi

- [1]. S. Ruder, *An Overview of Gradient Descent Optimization Algorithms. Learning Computer Science*. Insight Centre for Data Analytics, NUI Galway Aylieen Ltd., Dublin. 2017
- [2]. B. Achmad. *Kecerdasan Buatan*. Diktat Mata Kuliah Kecerdasan Buatan. Universitas Gadjah Mada. Yogyakarta, 2006

- [3]. R. K. S. Putri, *Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network untuk Klasifikasi Gambar*, Tugas Akhir, Jurusan Statistika, Universitas Islam Yogyakarta, Yogyakarta, Indonesia, 2018
- [4]. A. Krizhevsky, I. Sutskever, G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, University of Toronto, Canada. 2012
- [5]. N. Srivastava, G. Hinton, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research 15 1929-1958. 2014
- [6]. R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung. 2004
- [7]. T. Shafira, *Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras*. Tugas Akhir, Program Studi Statistika, Universitas Islam Yogyakarta, Yogyakarta, Indonesia, 2018
- [8]. S. Ilahiyah, Nilogiri, dan Agung, *Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network*. JUSTINDO Vol. 3, No. 2. Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember, Agustus, 2018
- [9]. I. W. Suartika, A. Y. Wijaya, dan R. Soaliman, *Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101*, JURNAL TEKNIK ITS Vol. 5 No.1, 2016
- [10]. E. N. Arrofiqoh, dan Haritanka, *Implementasi Metode Convolutional Neural Network untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi*, Program Studi Teknik Geomatika, Departemen Teknik Geodesi, Universitas Gadjah Mada, 2018
- [11]. Salsabila. *Penerapan Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Wayang Punakawan*, Tugas Akhir, Jurusan Statistika, Universitas Islam Yogyakarta, Yogyakarta, Indonesia, 2018
- [12]. T. Nurhikmat, *Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) pada Citra Wayang Golek*. Tugas Akhir, Program Studi Statistika, Universitas Islam Yogyakarta, Yogyakarta, Indonesia, 2018
- [13]. A. G. Saputra, E. Utami, dan H. Al-Fatta. *Analisis Penerapan Metode Convex Hull dan Convexity Defects untuk Pengenalan Isyarat Tangan*. Magister Teknik Informatika, Universitas Amikom Yogyakarta, 2015
- [14]. A. A. Al Anshori, B. Dirgantoro, dan N. Anbarsanti. *Perancangan dan Penerapan Pengenalan Pola Tangan pada Sistem Home Automation dengan Hierarchical Cascade Classifier*. E-Proceeding of Engineering: Vol.3 No. 3. Prodi Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom. Desember, 2016
- [15]. T. Liu, S. Fang, W. Peng, dan J. Zhang, *Implementation of Training Convolutional Neural Networks*, University of Chinese Academy of Sciences, Beijing, China. 2015