

Sistem Pengenal Isyarat Tangan Untuk Mengendalikan Gerakan Robot Beroda menggunakan *Convolutional Neural Network*

Habib Astari Adi ^{*1}, Ika Candradewi²

¹Program Studi Elektronika dan Instrumentasi, FMIPA, UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ^{*1}astarihabe@gmail.com, ²ika.candradewi@ugm.ac.id

Abstrak

Saat ini, Interaksi manusia dan komputer umumnya dilakukan dengan menggunakan remote control. Pendekatan ini cenderung tidak praktis pada pengendalian robot beroda karena harus selalu membawa alat perantara selama pengendalian dilakukan. Penerapan pengenalan isyarat tangan menggunakan teknik pengolahan citra digital dan machine learning dalam proses pengendalian robot beroda akan mempermudah pengendalian robot beroda karena tidak lagi membutuhkan alat perantara.

Dalam penelitian ini, citra isyarat tangan diambil menggunakan kamera kemudian akan dilakukan pengenalan isyarat tangan dan di proses menggunakan single board computer untuk dikenali. Hasil pengenalan diteruskan ke arduino leonardo dan motor DC untuk menggerakkan dua belas gerakan robot beroda. Metode yang digunakan dalam penelitian ini yaitu contrast stretching pada preprossesing dan Convolutional Neural Network (CNN) untuk pengenalan isyarat tangan.

Metode ini diujikan dengan variasi cahaya sebesar 26-140 lux, jarak dari tangan ke kamera sejauh 120-200 cm. Sistem pengenalan isyarat tangan menggunakan metode tersebut menghasilkan akurasi sebesar 97,5%, presisi 97,57%, sensitivitas 97,5%, spsivitas 99,77% dan f1 score 97,45%.

Kata kunci— pengenalan isyarat tangan, pengendalian robot, deep learning, convolutional neural network

Abstract

Currently, Human and computer interaction is generally done using a remote control. This approach tends to be impractical for wheeled robot operation because it must always carry an intermediary tool during the operation. The application of hand gesture recognition using digital image processing techniques and machine learning in the control process of wheeled robots will facilitate the control of wheeled robots because control no longer requires an intermediary tool.

In this study, hand image taken using a camera then will be processed using a single board computer to be recognized. The results of recognized are passed on to arduino leonardo and DC motor to control twelve wheeled robot movement. The method used in this study is contrast stretching for preprocessing and Convolutional Neural Network (CNN) for hand recognition.

This method is tested with a variation of bright 26-140 lux, the distance from the face to the camera is 120-200cm. Hand recognition systems using this method resulting accuracy 97,5%, precision 97,57%, sensitivity 97.5%, spesificity 99,77 and f1 score 97.45%.

Keywords— hand gesture recognition, robot operation, deep learning, convolutional neural network

1. PENDAHULUAN

Teknologi robot beroda terus berkembang mengikuti kemajuan zaman, robot banyak digunakan dalam bidang industri, kesehatan maupun militer dengan tujuan untuk mempermudah pekerjaan manusia. Pada umumnya, robot beroda dikendalikan dengan menggunakan *remote control*. Seiring dengan semakin tingginya interaksi manusia dengan robot, semakin kompleks pula tingkat kesulitan manusia untuk mengendalikan robot dengan menggunakan *remote control*. Hal ini dikarenakan apabila semakin banyak interaksi perintah maka kompleksitas bentuk dan tombol pada *remote control* pun akan semakin meningkat. Selain itu penggunaan *remote* akan menyulitkan karena harus dibawa terus menerus selama mengendalikan robot, tentunya hal ini akan menjadi lebih tidak praktis apabila *remote* rusak atau hilang. Agar lebih praktis pengendalian robot dapat dilakukan secara langsung melalui interaksi manusia.

Penelitian tentang pengendali robot atau pengenalan isyarat tangan telah dilakukan oleh beberapa orang sebelumnya. Dilakukan pengenalan terhadap 6 gestur untuk mengendalikan 6 gerakan robot beroda [1]. Selain menggunakan kamera webcam digunakan pula kamera kinect untuk melakukan pengenalan isyarat tangan [2]. Selain penelitian yang langsung diimplementasikan terhadap robot terdapat pula beberapa penelitian pengenalan isyarat tangan menggunakan *deep learnig* salah satunya adalah pengenalan isyarat tangan menggunakan CNN dengan dilakukan preprosesing berupa *contrast stretching* [3]. Selain itu CNN juga digunakan sebagai *classifier* dengan preproseing warna kulit sebagai cirinya[4]. Penggunaan pengenalan isyarat tangan juga diterapkan dalam pengendalian mouse [5]. Saat ini metode CNN menjadi sering diterapkan dalam pengenalan citra karena memiliki performa yang bagus melalui layer lebih dalam[6]. Sehingga diperlukan penelitian lebih lanjut untuk pengendalian robot beroda menggunakan isyarat tangan dengan metode CNN.

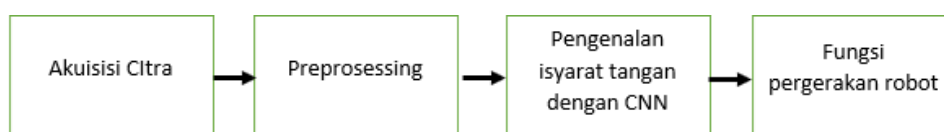
2. METODE PENELITIAN

Dalam penelitian ini, citra isyarat tangan diambil menggunakan kamera kemudian akan dilakukan pengenalan isyarat tangan dan di proses menggunakan single board computer untuk dikenali. Hasil pengenalan diteruskan ke arduino leonardo dan motor DC untuk menggerakkan dua belas gerakan robot beroda.

Sistem pengenalan isyarat tangan yang diimplementasikan pada pengendalian robot beroda yaitu melakukan deteksi wajah lalu dari deteksi tersebut akan menentukan *region of interest (ROI)* yang akan diambil, yaitu disebelah kanan dari citra wajah yang terdeteksi. Selanjutnya *ROI* yang telah diambil dari frame utama akan dilakukan pengenalan isyarat tangan menggunakan CNN. Sistem deteksi wajah menggunakan *Local Binary patern face detection* yang terdapat pada pustaka openCV 3.3.1 dengan bahasa python. *Local Binary patern* digunakan karena pada penelitian [7] didapatkan hasil yang baik dan cepat dalam komputasinya.

2.1 Deskripsi Sistem

Diagram blok pada Gambar 1 merupakan penjelasan cara kerja sistem pengenalan isyarat tangan untuk pengendalian robot beroda. Berawal dari masukan informasi berupa citra yang didapatkan dari kamera *webcam* sampai menjalankan beberapa fungsi pergerakan robot dengan masukannya adalah nilai hasil pengenalan isyarat tangan.



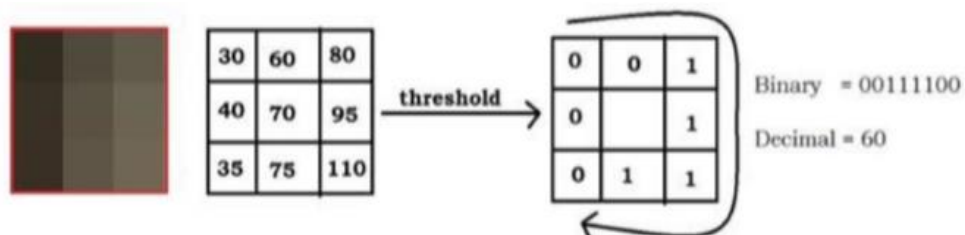
Gambar 1 Diagram Blok Sistem

Kegiatan pertama adalah proses deteksi wajah untuk menentukan dan mendapatkan ROI disebelah kanan wajah dengan cara mengambil citra dengan menggunakan *webcam*, kemudian

citra tersebut dimasukkan kedalam *pre-processing* yang terdiri dari *grayscale* dan *histogram*. *Grayscale* digunakan untuk membuat citra menjadi berwarna hitam dan putih (keabu-abuan), kemudian citra yang sudah diubah menjadi *grayscale* dibawa ke *histogram*, disini citra *grayscale* dilakukan pengaturan untuk meningkatkan kecerahan dalam pengolahan citra. Citra wajah selanjutnya dibandingkan dengan *data set LBP cascade*. Apabila saat dibandingkan sesuai maka citra tersebut terdapat tangan didalamnya. Dari citra wajah yang terdeteksi maka akan diambil crop area yang berada disebelah kanan wajah sejauh 30 pixel. Crop area ini yang nanti menjadi ROI citra yang akan dilakukan pengenalan isyarat tangan sebagai dasar pengendalian robot beroda.

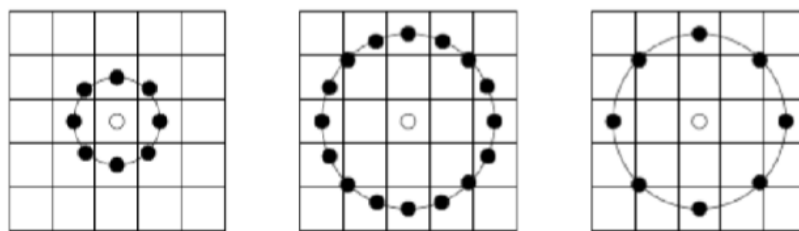
2. 2 Local Binary Pattern

Metode ini menetapkan sebuah label untuk setiap piksel dari citra dengan cara memberikan thresholding atau batasan yaitu 3x3 piksel dengan membandingkan nilai piksel tengah dengan piksel tetangga luarnya yang berhimpitan langsung seperti pada Gambar 2. Hasil yang didapatkan akan menjadi bentuk biner yang kemudian dikonversi menjadi bentuk desimal.



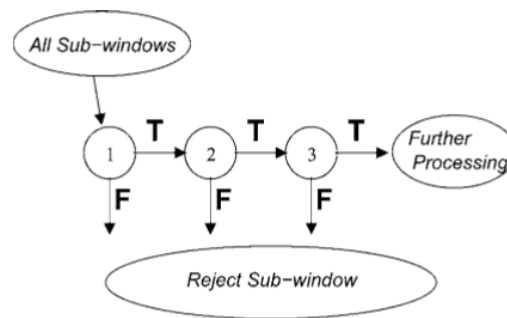
Gambar 2 Operasi Local Binary Pattern [5]

Gambar 3 merupakan notasi (P,R), P berartikan sampling point atau titik sampling pada lingkaran dengan radius R. Pelebaran dari lingkaran tersebut dinamakan uniform pattern. LBP disebut sebagai uniform pattern ketika nilai biner terdapat didalamnya paling banyak 2 bitwise transisi mulai dari 0 ke 1 atau sebaliknya. Semisal binernya berpola 00000000 terdapat 0 transisi, 00001111 terdapat 2 transisi, 01100000 terdapat 2 transisi. Dua transisi tersebut yang dinamakan dengan uniform. Pada komputasi pelabelan dari LBP, untuk setiap uniform pattern diberikan label terpisah atau tersendiri dan untuk semua non-uniform pattern diberikan label yang sama. Ketika menggunakan (8,R) neighborhood terdapat total 256 pola, dengan 58 pola uniform dan terdapat 59 label yang berbeda [7].



Gambar 3 Notasi (P,R) [7]

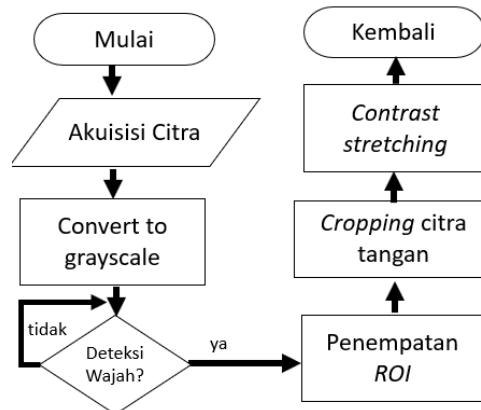
Cascade classifier digunakan untuk meningkatkan kemampuan deteksi dan mengurangi jumlah komputasi. Weak classifier digunakan untuk menolak sub-window dari citra yang dibagi tadi sebelum klasifikasi semakin kompleks yang biasanya mendapatkan tingkat kesalahan yang lebih rendah. Setiap tahapan didalam cascade sudah disusun dan dilatih oleh Adaboost. Hasil positif dari strong classifier diawal akan masuk ketahap kedua, kemudian akan masuk ketahap tiga dan seterusnya. Sedangkan untuk hasil negatif akan langsung ditolak pada tahapan yang diketahui bernilai negatif[8] dan dapat dilihat pada Gambar 4.



Gambar 4 Skematik dari Deteksi Cascade [8]

2.3 Pre-processing

Setelah dipotong dari frame utama data citra kemudian dilakukan *contrast stretching* agar gambar memiliki kontras yang lebih tajam. Selanjutnya untuk mengurangi beban komputasi maka citra akhir preprocessing akan diubah ukurannya (*resize*) menjadi ukuran 96x96 pixel. Dapat dilihat diagram alir proses ini pada Gambar 5.




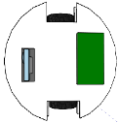

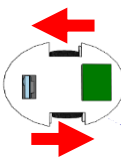

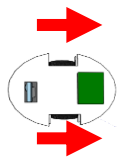

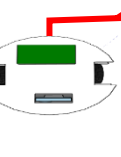

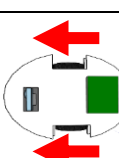
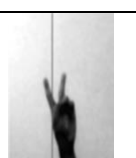
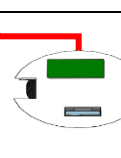

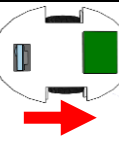

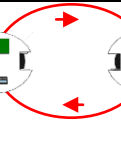

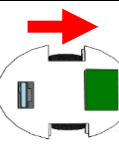

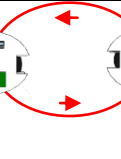

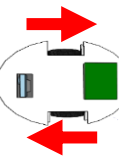

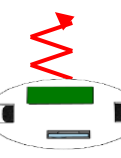
Gambar 5 Diagram Alir Preprocessing

2.4 Convolutional Neural Network

Metode CNN pertama kali diperkenalkan oleh LeCun et al. pada tahun 1998. LeCun et al. membuat model machine learning untuk rekognisi tulisan tangan yang populer dengan istilah MNIST (Modified National Institute of Standards and Technology database). LeNet 5 adalah suatu jaringan berlapis banyak berbasis CNN. Model ini diberi nama LeNet-5 yang merupakan pengembangan dari LeNet-LeNet sebelumnya, dimana LeNet 5 memiliki jumlah parameter bebas atau jumlah lapisan yang lebih banyak dari pendahulunya.

Data yang digunakan sebagai bahan penelitian skripsi ini dikumpulkan dari lima partisipan dengan menggunakan kamera *webcam*. Dua belas isyarat yang digunakan pada penelitian ini adalah isyarat tangan kanan dan menggunakan 10 pola isyarat bilangan yang biasa digunakan sehari-hari dan beberapa penambahan. Isyarat tangan kanan yang telah dikumpulkan menggunakan kamera dengan variasi pencahayaan dan jarak dapat dilihat pada Tabel 1. Citra isyarat tangan untuk setiap kelas terdiri dari 448 citra latih, 112 citra validasi dan 50 citra uji. Dua belas citra isyarat tangan tersebut selanjutnya dilakukan pengenalan citra untuk mengendalikan robot, agar robot dapat memahami perintah yang diberikan manusia secara langsung.

Tabel 1 Pengendalian gerak robot beroda

No	Isyarat Tangan	Gerak	Gerak Robot	No	Isyarat Tangan	Gerak	Gerak Robot
1		Diam		7		Putar kiri	
2		Maju		8		Gerakan menghindar kanan	
3		Mundur		9		Gerakan menghindar kiri	
4		Belok Kanan		10		Gerakan memutar searah jarum jam	
5		Belok Kiri		11		Gerakan memutar berlawanan jarum jam	
6		Putar kanan		12		Gerakan zig-zag	

Setelah proses pre-processing dan *cropping*, ekstraksi ciri dan pengklasifikasian data citra akan dilakukan oleh CNN. Dalam penelitian ini, terdapat dua buah arsitektur CNN, model pertama merupakan pengembangan Lenet 5 yang telah dikembangkan dari penelitian [9] dan model kedua juga merupakan CNN Lenet 5 namun dengan penambahan layer konvolusi, relu dan *maxpooling*. Diagram alir proses CNN tahap *training* dan *testing* ditunjukkan pada Gambar 6.

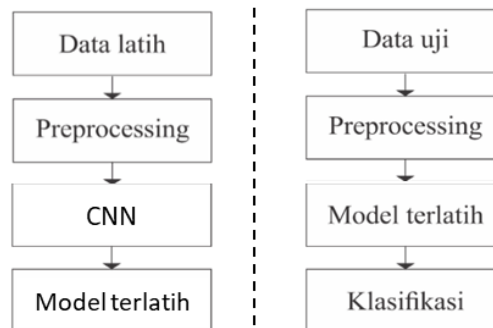
Model akhir pengembangan arsitektur CNN kedua memiliki empat buah lapisan konvolusi dan empat buah lapisan *max pooling*. Lapisan konvolusi ini memiliki lebar filter 3×3 . Sementara lapisan *max pooling* memiliki lebar filter 2×2 . Pada lapisan *fully connected* layer digunakan fungsi aktivasi *rectified linier unit* sejumlah dua buah. Kedua lapisan *fully connected* memiliki 5120 *hidden unit* dan *dropout* sebesar 0,5. Model CNN 1 dan CNN 2 dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2 Pengembangan arsitektur CNN 1 [2]

<i>Layer</i>	<i>Tipe</i>	<i>Neuron</i>	<i>Filter</i>
0	<i>input</i>	$96 \times 96 \times 1$	
1	<i>Conv 1</i>	$94 \times 94 \times 64$	$3 \times 3 \times 64$
2	<i>Max pooling</i>	$47 \times 47 \times 64$	2×2
3	<i>Conv 2</i>	$45 \times 45 \times 64$	$3 \times 3 \times 64$
4	<i>Max pooling</i>	$22 \times 22 \times 64$	2×2
5	<i>Conv 3</i>	$20 \times 20 \times 64$	$3 \times 3 \times 64$
6	<i>Max pooling</i>	$10 \times 10 \times 64$	2×2
7	<i>flatten</i>	6400	
8	<i>Fully connected 1</i>	4096	
9	<i>Fully connected 1</i>	4096	
10	<i>Classifier</i>		

Tabel 3 Pengembangan arsitektur CNN 2

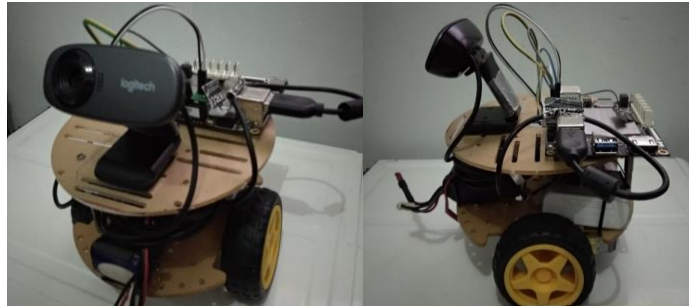
<i>Layer</i>	<i>Tipe</i>	<i>Neuron</i>	<i>Filter</i>
0	<i>input</i>	$96 \times 96 \times 1$	
1	<i>Conv 1</i>	$94 \times 94 \times 64$	$3 \times 3 \times 64$
2	<i>Max pooling</i>	$47 \times 47 \times 64$	2×2
3	<i>Conv 2</i>	$45 \times 45 \times 64$	$3 \times 3 \times 64$
4	<i>Max pooling</i>	$22 \times 22 \times 64$	2×2
5	<i>Conv 3</i>	$20 \times 20 \times 64$	$3 \times 3 \times 64$
6	<i>Max pooling</i>	$10 \times 10 \times 64$	2×2
7	<i>Conv 4</i>	$8 \times 8 \times 32$	$3 \times 3 \times 32$
8	<i>Max pooling</i>	$4 \times 4 \times 32$	2×2
9	<i>flatten</i>	512	
10	<i>Fully connected 1</i>	5120	
11	<i>Fully connected 2</i>	5120	
12	<i>Classifier</i>		



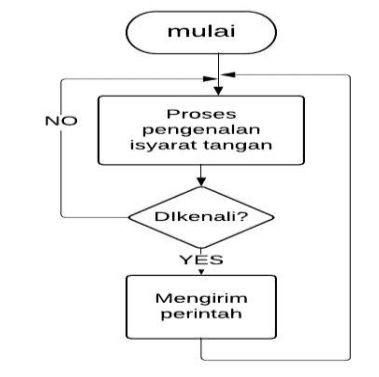
Gambar 6 Blok Diagram (a) Pelatihan dan (b) Pengujian

2.5 Fungsi pergerakan robot

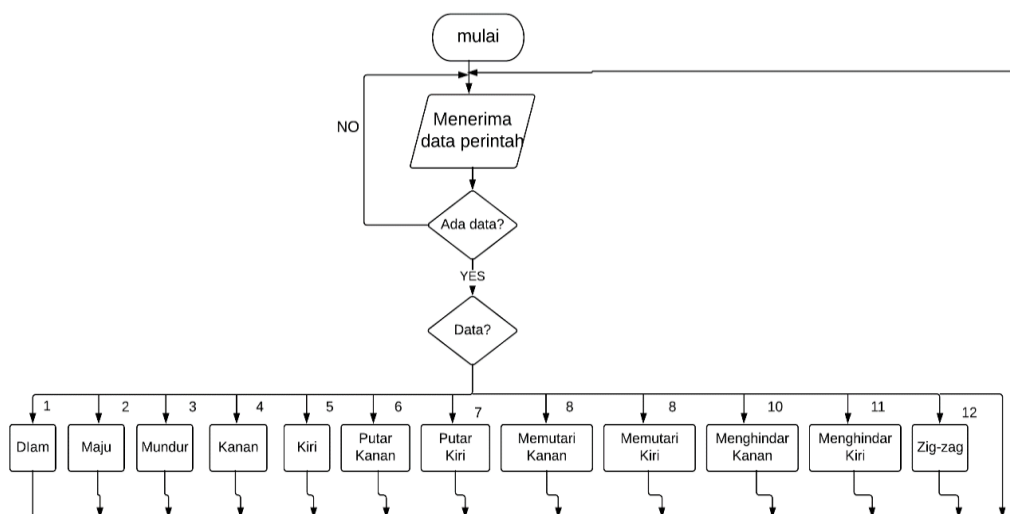
Pengendalian robot beroda dilakukan dengan cara mengarahkan isyarat tangan yang diperintahkan ke kamera robot beroda, setelah isyarat tangan dikenali Lattepanda akan mengirimkan perintah berupa data serial ke dalam mikroprosesor arduino leonardo melalui komunikasi serial. Setelah data diterima robot akan melakukan pergerakan dan menentukan perilaku sesuai program. Analisa proses pengendalian gerakan robot dapat digambarkan menggunakan flowchart ditunjukkan pada Gambar 7, Gambar 8 dan Gambar 9.



Gambar 7 robot beroda dengan pemroses Lattepanda dan Arduino Leonardo



Gambar 8 Flowchart proses pengendalian pada Lattepanda



Gambar 9 Flowchart proses pengendalian pada Arduino Leonardo

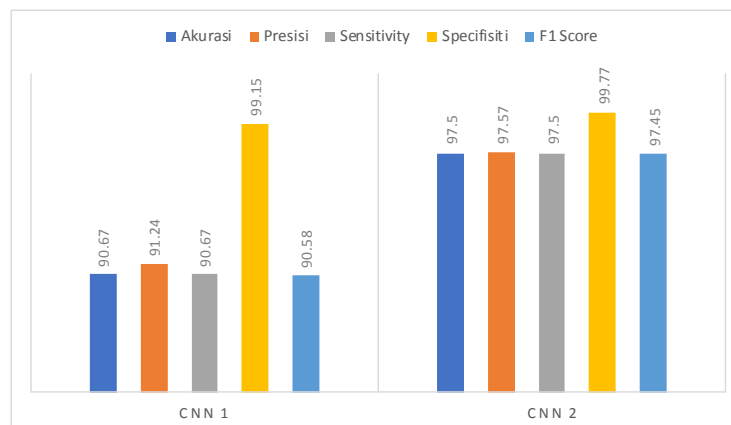
3. HASIL DAN PEMBAHASAN

Dalam penelitian ini, terdapat dua buah arsitektur CNN, model pertama merupakan pengembangan Lenet 5 yang telah dikembangkan dari penelitian [9] dan model kedua juga merupakan CNN Lenet 5 namun dengan penambahan layer konvolusi, aktivasi relu dan *maxpooling*. Setiap model CNN memiliki variasi berupa nilai *learning rate*. Selama pengujian variasi *hyperparameter*, validasi akurasi dilakukan pada data validasi, lalu model terbaik akan digunakan pada data uji. Hasil validasi CNN ditunjukkan pada Tabel 4.

Tabel 4 Hasil pelatihan dan validasi CNN

<i>Learning rate</i>	Akurasi – CNN 1		Akurasi – CNN 2	
	Data latih	Data validasi	Data latih	Data validasi
1×10^{-4}	9,92%	11,22%	7,68%	9,33%
1×10^{-3}	27,12%	43,44%	15,27%	29,24%
1×10^{-2}	99,23%	99,26%	99,46%	99,93%

Terlihat model dengan *learning rate* 1×10^{-2} memiliki akurasi akhir yang paling baik pada kedua model. Model ini lalu dievaluasi pada data uji dan pada model pertama masih terjadi *overfitting* dengan nilai akurasi hanya sebesar 90,67% dan pada model kedua memperoleh akurasi cukup baik dengan nilai sebesar 97,5%. Terlihat dari Gambar 10 model kedua memiliki performa yang lebih baik dibandingkan model pertama, baik dari segi akurasi, presisi, *sensitivity*, *specificity* dan *F1 score*. Berdasarkan performa tadi maka model CNN kedua dipilih sebagai model terbaik dengan akurasi 97,5% untuk dilakukan pengujian untuk mendapatkan kondisi yang optimal dalam intensitas cahaya, latar belakang ruangan dan jarak tangan terhadap *webcam*.



Gambar 10 Performa CNN pada klasifikasi isyarat tangan

Pengujian pertama yang dilakukan adalah pencahayaan *webcam* sehingga system dapat mendeteksi tangan. Dari pengujian yang dilakukan sebanyak 60 kali pada setiap kondisi pencahayaan maka didapat bahwa system dapat mendeteksi tangan pada rentang nilai 26 hingga 140 lux. Pengujian kedua adalah variasi latar belakang ruangan. Dari 4 kondisi yang ada yaitu latar belakang biru dan polos, latar belakang coklat dan polos, latar belakang hijau dan polos, latar belakang dengan perabot, tangan dapat dikenali dengan baik saat latar belakangnya *homogen* dan polos. Sistem kesulitan dalam mendeteksi tangan ketika latar belakang ruangan kompleks dengan perabot karena system masih menganggap perabot sebagai objek tangan. Hasil pengujian pada masing masing ruangan dapat dilihat pada Gambar 11.

Pengujian ketiga yang dilakukan adalah jarak tangan terhadap *webcam* sehingga system dapat mendeteksi tangan. Dari pengujian yang dilakukan sebanyak 60 kali maka didapat bahwa system dapat mendeteksi tangan pada jarak 120 hingga 200 cm. Pada Tabel 5, Tabel 6, dan Tabel 7 dapat dilihat masing masing pengujian untuk parameter tersebut.

Tabel 5 Hasil pengujian pengaruh intensitas cahaya dalam ruangan

No	Intensitas Cahaya (lux)	Keterangan ruang	Jumlah percobaan	Akurasi
1.	10-25	Redup	60	90%
2.	26-80	Sedikit Terang	60	100%
3.	80-140	Terang	60	98,4%
4.	140-200	Sangat Terang	60	93,4%

Tabel 6 Hasil pengujian jarak tangan terhadap webcam

No	Jarak(cm)	Jumlah Percobaan	Akurasi
2.	120 cm	60	100%
3.	160 cm	60	100%
4.	200 cm	60	98,4%

Tabel 7 Hasil pengujian latar belakang/background ruangan

No	Jenis <i>Background</i> Ruangan	Jumlah Percobaan	Akurasi
1.	Hijau dan Polos	60	100%
2.	Biru dan Polos	60	100%
3.	Cokelat dan Polos	60	100%
4.	Ruangan I	60	68,4%



Gambar 11 Pengujian dengan variasi ruangan

Setiap pengenalan isyarat tangan yang didapat tadi diimplementasikan untuk menjalankan fungsi pergerakan robot. Fungsi pergerakan yang digunakan dalam sistem ini yaitu maju, mundur, belok kanan, belok kiri, putar kanan, putar kiri, diam, menghindar kanan, menghindar kiri, memutar searah jarum jam, memutar berlawanan jarum jam dan *zig-zag*. Semua fungsi telah dicoba dan berjalan dengan baik sesuai dengan isyarat tangan yang di kenali dan pada Gambar 12 contoh fungsi yang dijalankan. Menurut penelitian[10], sebuah sistem dikatakan *real-time* jika interaksi antara komputer dan manusia lebih natural dan intuitive tanpa adanya perangkat tambahan dengan minimal frame per *second* nya adalah 10 fps dengan rata-rata 9,36 detik.



Gambar 12 Fungsi pergerakan maju dijalankan

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa *convolutional neural network* dapat digunakan untuk mengenali isyarat tangan. Akurasi penghitungan jari yang didapat dari perhitungan matematika adalah sebesar 97,5%.

Penggunaan deteksi tangan dengan menggunakan LBP *face detection* memudahkan proses inialisasi untuk proses selanjutnya. Dengan adanya proses pendeteksian tangan sebelumnya maka didapatkan *ROI* dimana proses pengenalan isyarat tangan dilakukan. Jarak optimal untuk dapat menghitung jari dengan baik adalah 120-200 cm dengan kecerahan cahaya 26-140 lux. Sistem ini berjalan optimal di ruang yang terang dengan latar homogen dan polos dan kesulitan mengenali pada ruang berlatar belakang kompleks dengan perabot. Dengan menggunakan parameter tersebut dapat dikenali isyarat tangan sehingga fungsi pergerakan robot dapat dijalankan sesuai dengan isyarat tangan tersebut.

5. SARAN

Pada penelitian ini terdapat beberapa hal yang perlu disempurnakan. Saran-saran yang dapat digunakan untuk menyempurnakan penelitian selanjutnya seperti perlunya pelatihan data menggunakan variasi pengguna yang lebih beragam sehingga diketahui tingkat reliabilitas sistem, pada data yang lebih beragam.

DAFTAR PUSTAKA

- [1] J. S. Komputer, F. Ilmu, K. Univesitas, and S. Selatan, "SISTEM PENGENALAN GESTURE TANGAN UNTUK KENDALI GERAK MOBILE ROBOT," pp. 151–156.
- [2] Y. Wang, G. Song, Y. Zhang, and W. Wang, "Wheeled Robot Control Based on Gesture Recognition Using the Kinect Sensor," no. December, 2013.
- [3] A. A. Alani and G. Cosma, "Hand Gesture Recognition Using an Adapted Convolutional Neural Network with Data Augmentation," *2018 4th Int. Conf. Inf. Manag.*, pp. 5–12, 2018.
- [4] S. Hussain and R. Saxena, "Hand Gesture Recognition Using Deep Learning," pp. 48–49, 2017.
- [5] P. Xu, "A Real-time Hand Gesture Recognition and Human-Computer Interaction System," pp. 1–8, 2017.
- [6] P. S. Segmentation *et al.*, "DeepUNet : A Deep Fully Convolutional Network for," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 11, pp. 3954–3962, 2018.
- [7] K. Kadir, M. K. Kamaruddin, H. Nasir, S. I. Safie, and Z. A. K. Bakti, "A comparative study between LBP and Haar-like features for Face Detection using OpenCV," *2014 4th Int. Conf. Eng. Technol. Technopreneuship, ICE2T 2014*, vol. 2014–August, pp. 335–339, 2015 [Online]. Available: <http://ieeexplore.ieee.org/document/7006273/> [Accessed: 29-Aug-2019]
- [8] J. Chang-yeon, "Face Detection using LBP features," Stanford.Edu, pp. 1–4, 2008 [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Face+Detection+using+LBP+features#0> [Accessed: 10-Jul-2019]
- [9] J. L. F. C, E. G. C. A, and L. E. R, "Application of Convolutional Neural Networks for Static Hand Gestures Recognition Under Different Invariant Features," pp. 5–8, 2017.
- [10] S. Qin, X. Zhu, Y. Yang, Y. Jiang, "Real-time Hand Gesture Recognition from Depth Images Using Convex Shape Decomposition Method" Vol. 74 no. , pp. 47-58. January 2014 [Online]. Available: <https://link.springer.com/article/10.1007/s11265-013-0778-7>