

# ELECTRIC WHEELCHAIR MOVEMENT CONTROL BASED ON FINGER PATTERN RECOGNITION WITH CONVOLUTIONAL-LSTM METHOD

## KONTROL PERGERAKAN KURSI RODA ELEKTRIK BERDASAR PENGENALAN POLA JARI TANGAN DENGAN METODE *CONVOLUTIONAL-LSTM*

Iswahyudi<sup>1\*</sup>, Dudi Irawan<sup>2</sup>, Daffa Hibran Aditia<sup>3</sup>.

<sup>1</sup>Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Jember, Indonesia  
Email: muhammadiiswahyudi@unmuhjember.ac.id

<sup>2</sup>Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Jember, Indonesia  
Email: dudi.irawan@unmuhjember.ac.id

<sup>3</sup>Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Jember, Indonesia  
Email: adytiadaffa@gmail.com

**Received: March 22, 2024   Revised: April 03, 2024   Published: April 03, 2024**

DOI: <https://doi.org/10.24912/tesla.v26i1.29503>

### Abstract

Wheelchairs are a tool for people with disabilities who have difficulty walking to do their daily activities. Users of conventional wheelchairs will quickly get tired if they have to walk long distances. The application of technology has expanded in all fields including biomedical. With the advancement of technology, a variety of wheelchair control interfaces have been developed. In this study, the researchers will develop on the navigation control of the movement of the electric wheelchair based on the identification of the finger posture of the hand. Five finger patterns represent the motion of the wheelchairs to move forward, backwards, right, left, and stop. The research data set was successfully collected as much as 8000 data per finger pattern with a total of 40,000 samples. The proposed methods by the researchers are using deep learning methods such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). The combination of CNN and LSTM was chosen because the CNN method is highly focused on solving current object recognition problems. The combination with the LSTM method is expected to increase the degree of accuracy of the identification of five finger patterns that represent the movement of the wheelchair: forward, backward, right, left, and stop. Based on the results of offline testing, the accuracy score was 96.9% and the average response time was 179 ms. The highest accurate value in the advanced class was 99.7% and the lowest accurately in the stop and right classes was 97.9%. The greatest recall or sensitivity value at the stop class is 99.6% and the least recall value on the left class is 9.7%. The system is capable of predicting as much as  $\pm 6$  frames per second. (fps).

**Keywords:** Disability; Wheelchair; Finger pattern; CNN-LSTM

### Abstrak

Kursi roda merupakan alat bantu bagi penyandang disabilitas yang mengalami kesulitan berjalan untuk melakukan aktivitas sehari-hari. Pengguna kursi roda konvensional akan cepat lelah jika harus berjalan jauh. Penerapan teknologi telah merambah pada semua bidang termasuk dalam bidang biomedis. Seiring dengan perkembangan teknologi, telah dikembangkan berbagai antarmuka kontrol kursi roda. Pada penelitian ini peneliti akan melakukan pengembangan pada kontrol navigasi pergerakan kursi roda elektrik berdasarkan pengenalan pose jari tangan. Lima pola jari tangan mewakili pergerakan kursi roda untuk bergerak maju, mundur, kanan, kiri, dan berhenti. Dataset penelitian yang berhasil dikumpulkan sebanyak 8000 gambar data tiap pola jari tangan dengan total sampel sebanyak 40000 gambar. Metode yang peneliti usulkan menggunakan metode *deep learning* yaitu *Convolutional Neural Network (CNN)* dan *Long Short-Term Memory (LSTM)*. Metode gabungan *CNN* dan *LSTM* dipilih karena metode *CNN* sangat mumpuni untuk menyelesaikan permasalahan pengenalan objek saat ini. Penggabungan dengan metode *LSTM* diharapkan menambah tingkat akurasi dari pengenalan lima pola jari tangan yang mewakili pergerakan kursi roda yaitu

maju, mundur, kanan, kiri, dan berhenti. Berdasarkan hasil pengujian offline didapatkan nilai akurasi sebesar 96,9% dan rata-rata time respon sebesar 179 ms. Nilai presisi paling besar pada kelas maju sebesar 99,7% dan nilai presisi paling kecil pada kelas berhenti dan kanan sebesar 97,9%. Nilai *recall* atau sensitivitas paling besar pada kelas berhenti sebesar 99,6% dan nilai *recall* paling kecil pada kelas kiri sebesar 96,7%. Sistem yang dibuat mampu melakukan prediksi sebanyak  $\pm 6$  frame per second (fps).

**Kata Kunci:** Disabilitas; Kursi Roda; Pola Jari Tangan; CNN-LSTM

## PENDAHULUAN

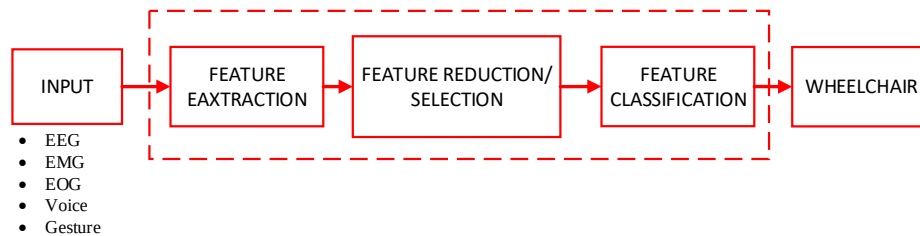
Tunadaksa adalah individu yang memiliki gangguan gerak yang disebabkan oleh kelainan neuro-muskular dan struktur tulang yang bersifat bawaan, sakit atau akibat kecelakaan, termasuk *celebral palsy*, amputasi, polio, dan lumpuh [1]. Hasil survey antar Penduduk SUPAS (2015) sebanyak 3,76 % penduduk Indonesia mengalami kesulitan berjalan [2]. Kursi roda merupakan alat bantu bagi penyandang disabilitas yang mengalami kesulitan berjalan untuk melakukan aktifitas sehari-hari. Kursi roda awalnya digerakkan secara manual dengan menggunakan kekuatan tangan atau dengan bantuan orang lain. Permasalahan penggunaan kursi roda dengan menggunakan penggerak tangan adalah pengguna cepat lelah jika harus berjalan jauh. Gestur atau gerakan tubuh merupakan salah satu jenis komunikasi nonverbal yang paling mudah dibaca [3]. Pengenalan gerakan tangan adalah proses memahami dan mengklasifikasikan gerakan yang bermakna oleh tangan manusia [4]. Penelitian mengenai pengenalan gestur bertujuan untuk menciptakan sebuah sistem yang dapat mengidentifikasi secara spesifik gestur manusia dan dipergunakan untuk menyampaikan informasi atau perangkat pengontrol. Kecerdasan buatan (*Artificial Intelligence*) merupakan bagian bidang ilmu yang dapat menirukan tingkah laku dan fungsi otak manusia, dengan tujuan untuk menggantikan peran operator dengan mesin cerdas [5].

Pengenalan gestur telah diterapkan pada beberapa aplikasi, diantaranya aplikasi pengenalan bahasa isyarat [6], kontrol robot [6], kontrol kendaraan [7] dan sebagainya. Penerapan teknologi telah merambah pada semua bidang termasuk dalam bidang biomedis. Seiring dengan perkembangan teknologi, telah dikembangkan berbagai antarmuka kontrol kursi roda *hands free* berdasarkan gerakan dan pose jari tangan. Beberapa referensi yang menjadi acuan pada penelitian ini yaitu penelitian sebelumnya yang dilakukan oleh [8] telah dilakukan penelitian mengenali pola jari tangan menggunakan metode *Convolutional Neural Network (CNN)*. Pada penelitian tersebut menghasilkan tingkat akurasi 93,6%. Pada penelitian lainnya yaitu kontrol kursi roda menggunakan sensor EMG (Electromyograf) dan sensor Accelerometer menggunakan sistem jaringan syaraf tiruan [9]. Pada penelitian ini hasil yang didapatkan adalah model mampu mendeteksi kontraksi otot dan dapat menentukan arah kursi roda sesuai yang diinginkan. Penelitian lainnya oleh [10] menggunakan sensor Sensor Accelerometer GY-61 yang bisa menangkap derajat kemiringan koordinat x-y-z digunakan sebagai kendali kursi roda. Kursi roda ini bisa juga dikendalikan melalui handphone berbasis Android.

Pada artikel ini peneliti akan melakukan pengembangan pada kontrol navigasi pergerakan kursi roda elektrik berdasarkan pengenalan pose jari tangan. Metode yang peneliti usulkan menggunakan metode *deep learning*. Metode *deep learning* yang peneliti gunakan adalah *Convolutional Neural Network (CNN)* dan *Long Short-Term Memory (LSTM)*. Pola jari tangan sebagai masukan kontrol kursi roda ditangkap menggunakan kamera. Tujuan dilakukan penelitian ini adalah untuk menguji apakah metode *CNN-LSTM* dapat digunakan sebagai pengenalan pose jari tangan menggunakan sensor kamera.

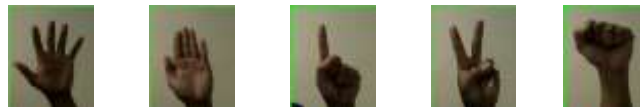
## METODELOGI PENELITIAN

### A. Human Machine Interface (HMI)



Gambar 1 Blok Diagram *Human Machine Interface* Kontrol Kursi Roda [11]

Ada banyak cara untuk mengontrol pergerakan kursi roda ditunjukkan pada gambar 1. *Input* data berupa sinyal atau gambar dapat digunakan sebagai masukan kontrol untuk mengoperasikan pergerakan kursi roda. Pada umumnya penerjemahan suatu sinyal masukan hingga keluaran memiliki kombinasi dua hingga tiga tahap yaitu ekstraksi ciri, pemilihan atau fitur reduksi, dan klasifikasi pola masukan untuk mendekode aktivitas yang diinginkan pengguna. Pada penelitian ini peneliti menggunakan masukan data berupa *gesture* berupa gambar. Gambar *gesture* yang digunakan peneliti adalah lima pose jari tangan untuk mengatur pergerakan kursi roda untuk berhenti, maju, mundur, kanan, dan kiri

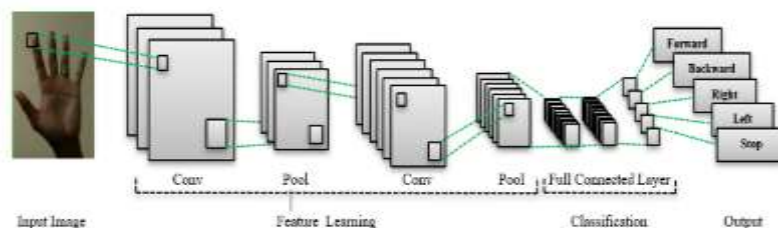


(a) Maju (b) Mundur (c) Kanan (d) Kiri (e) Berhenti

Gambar 2. *Gesture* Lima Pose Jari Tangan

### B. Convolutional Neural Network

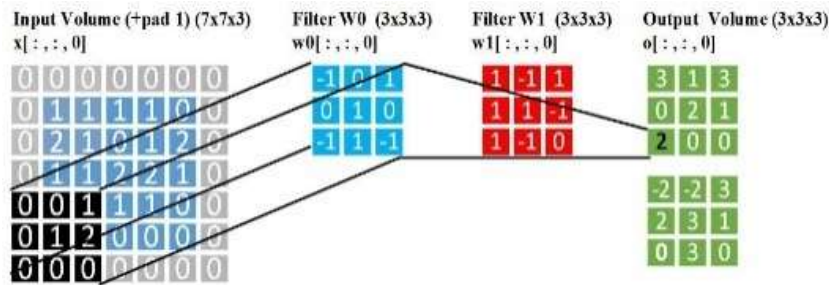
Bentuk paling awal dari CNN adalah model Neocognitron yang diusulkan oleh Kunihiko Fukushima[12]. Ini terdiri dari beberapa lapisan yang secara otomatis belajar sebuah hierarki atau urutan bertingkat dari abstraksi fitur untuk pengenalan pola. Neocognitron terinspirasi dari karya Hubel dan Wiesel [13] pada korteks visual primer yang menunjukkan neuron di otak disusun dalam bentuk lapisan. Lapisan-lapisan ini belajar mengenali visual pola dengan mengekstraksi fitur lokal pertama dan kemudian menggabungkannya untuk mendapatkan tingkat representasi yang lebih tinggi.



Gambar 3. Arsitektur *Convolutional Neural Network*

Pada gambar 3 dapat dijelaskan bahwa CNN terdiri dari 2 bagian utama yaitu fitur *learning* dan *classification*. Pembelajaran fitur terdiri dari lapisan *convolutional* dan *pooling*. Fitur klasifikasi berisi *Fully Connected Layer* (FCL). Lapisan *convolutional* atau filter adalah komponen yang paling penting. Filter digeser keseluruh bagian gambar input, setiap pergeseran akan dilakukan operasi dot sehingga menghasilkan output yang dinamakan *activation map* atau *feature map*. Operasi matematika konvolusi dapat ditulis sebagai berikut:

$$s(t) = (x * t)(t) = \sum_{\alpha=-\infty}^{\infty} x(\alpha) * w(t - \alpha) \quad (1)$$

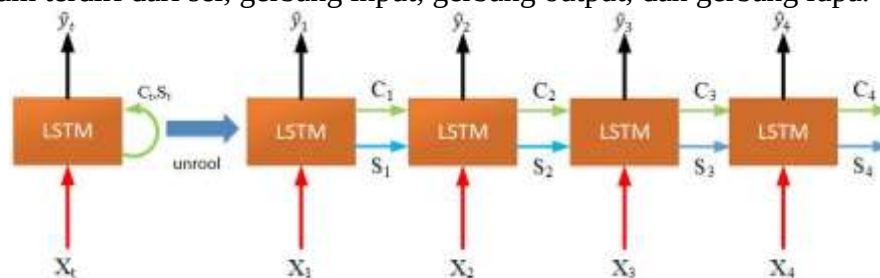


Gambar 4. Proses Konvolusi 2 filter

*Pooling* adalah pengurangan ukuran matriks, biasanya setelah proses konvolusi, terdiri dari filter dengan ukuran dan langkah tertentu yang bergeser ke seluruh area peta fitur secara bergantian. *Fully Connected Layer* (FCL) adalah sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung secara keseluruhan supaya dapat diklasifikasikan.

### C. Long Short Term Memory (LSTM)

*Long Short-Term Memory* (LSTM) adalah variant dari *Recurrent Neural Network* (RNN)[14] yang digunakan dalam bidang *Deep Learning*. Jaringan LSTM berbeda dengan *Multi Layer Perceptron* (MLP). Pada MLP jaringan terdiri dari lapisan neuron. LSTM memiliki koneksi berulang sehingga status dari aktivasi neuron sebelumnya dari langkah waktu sebelumnya digunakan sebagai konteks untuk merumuskan keluaran. Unit LSTM umum terdiri dari sel, gerbang input, gerbang output, dan gerbang lupa.

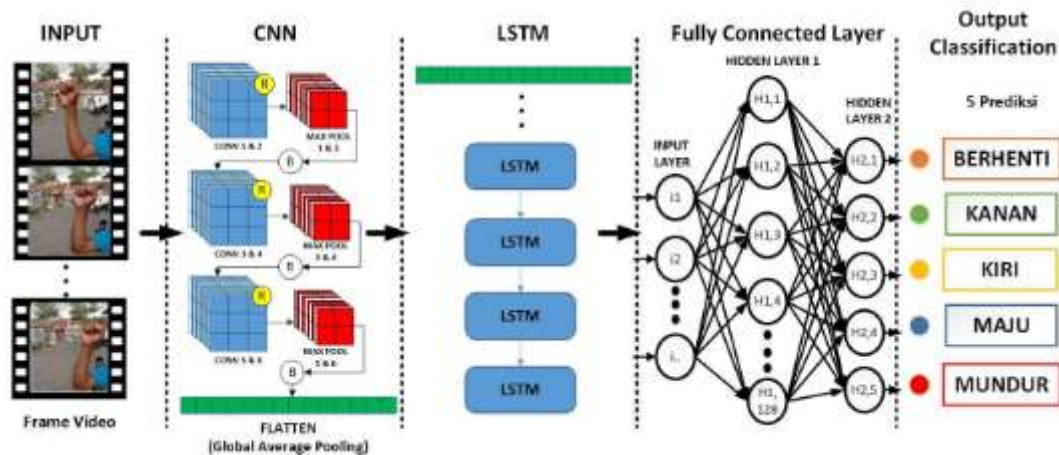


Gambar 5. Cara Kerja Jaringan LSTM

Gambar 5 menjelaskan mengenai gambaran dari cara kerja jaringan LSTM terdiri dari modul LSTM yang dipanggil secara berulang.

### D. Pembuatan model CNN Long Short-Term Memory

KONTROL PERGERAKAN KURSI RODA ELEKTRIK BERDASAR PENGENALAN POLA JARI TANGAN DENGAN METODE CONVOLUTIONAL-LSTM



Gambar 6. Konsep Metode *CNN- LSTM*

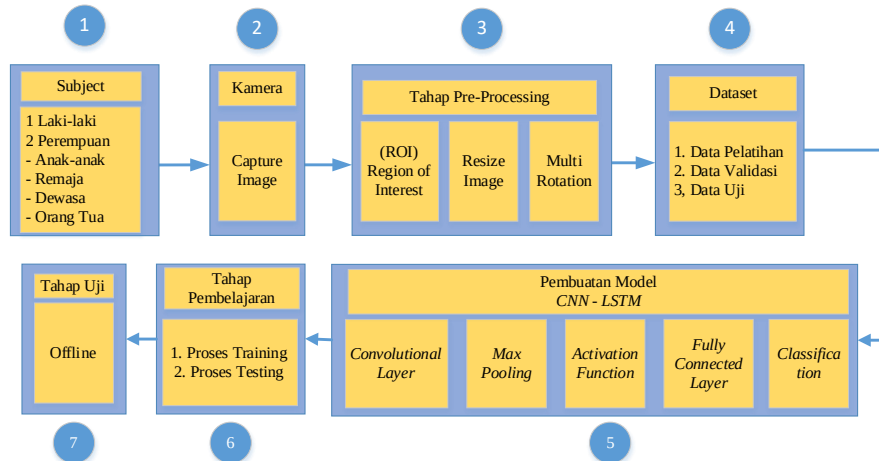
Gambar 6 menjelaskan gambaran konsep metode *CNN-LSTM* yang diusulkan oleh peneliti. Pada bagian input berupa data sequential berupa data video yang tersusun atas *frame-frame*. Input pada penelitian ini berupa video 5 pola jari tangan yang mewakili pergerakan kursi roda elektrik, yaitu berhenti, kanan, kiri, mau, dan mundur. Tiap video pola jari tangan dengan durasi 5 detik. Sebelum input dimasukkan ke *CNN* data video di reprosesing terlebih dahulu. Pada bagian ini tiap video pola jari tangan dibuat sampel sebanyak 8000 frame.

*CNN* berfungsi sebagai feature extraction layer. Pada bagian feature extraction layer terdapat Convolutional layer, pooling dan flatten. *Convolutional layer* berisi sebuah filter yang memiliki nilai ukuran tinggi, lebar dan kedalaman.

*Layer LSTM* menerima input hasil dari proses flatten. *Layer LSTM* berfungsi untuk mengurutkan data hasil fitur ekstraksi dan membuang data yang tidak diinginkan. Hasil dari *layer LSTM* masih berbentuk multi dimensi array. Hasil *layer LSTM* dirubah menjadi satu dimensi atau vektor agar bisa digunakan sebagai input fully connected layer. Bagian akhir adalah output menghasilkan output berupa nilai probabilitas dari ke-5 pola jari tangan.

### E. Tahapan Penelitian

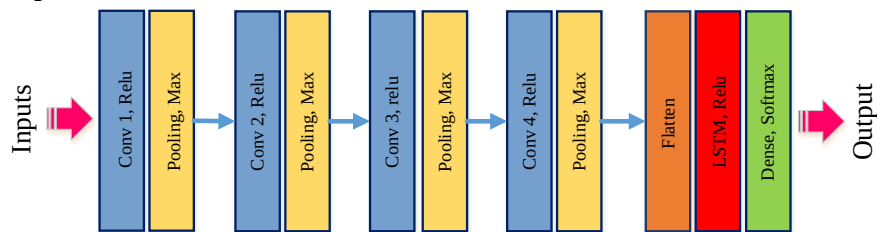
Tahapan penelitian menjelaskan mengenai proses pembuatan sistem dari awal hingga akhir. Untuk memahami cara kerja sistem yang akan di bangun, perlu dibuat tahapan penelitian untuk mempermudah pembaca dalam memahami alur dan proses penelitian yang dilakukan.



Gambar 7. Diagram Tahapan Penelitian

#### F. Pembuatan Model Arsitektur Convolutional-Long Short Term Memory (LSTM)

Pada model Convolutional-LSTM, dimana input gambar melewati lapisan konvolusi dan hasilnya adalah kumpulan yang diratakan ke array 1D. Proses dilakukan ke semua gambar dalam set waktu, hasilnya adalah serangkaian fitur dari waktu ke waktu, dan ini adalah input lapisan LSTM.



Gambar 8. Architecture Convolutional-LSTM

Pada Gambar 8 dapat dilihat arsitektur *Convolutional-LSTM* yang diusulkan. Perbedaan terdapat pada bagian *Fully Connected* terdapat 2 layer yaitu 1 layer LSTM dengan fungsi aktivasi *Relu* dan 1 layer Dense atau *hidden layer* dengan fungsi aktivasi *softmax*. Bagian akhir adalah output menghasilkan output berupa nilai probabilitas dari ke-5 pola jari tangan.

Tabel 1. Tabel Network Parameter Convolutional-LSTM

| Layer                 | Activation  | Filter                          | Output Shape   |
|-----------------------|-------------|---------------------------------|----------------|
| Input Layer           | -           | -                               | 224 × 224      |
| Convolutional Layer 1 | Relu        | Depth: 16 Filter: 6×6 Stride: 2 | 16 × 110 × 110 |
| Pooling Layer 1       | Max Pooling | Filter: 3×3 Stride 1            | 16 × 108 × 108 |
| Convolutional layer 2 | Relu        | Depth: 32 Filter: 3×3 Stride: 1 | 32 × 106 × 106 |
| Pooling Layer 2       | Max Pooling | Filter: 2×2 Stride 1            | 32 × 104 × 104 |
| Convolutional layer 3 | Relu        | Depth: 64 Filter: 3×3 Stride: 1 | 64 × 102 × 102 |
| Pooling Layer 3       | Max Pooling | Filter: 2×2 Stride 1            | 64 × 100 × 100 |
| Convolutional layer 4 | Relu        | Filter: 6×6 Depth: 6 Stride: 2  | 64 × 98 × 98   |
| Pooling Layer 4       | Max Pooling | Filter: 2×2 Stride 2            | 64 × 49 × 49   |

|                |                |   |     |
|----------------|----------------|---|-----|
| <b>Flatten</b> | -              | - |     |
| <b>LSTM</b>    | <i>Relu</i>    | - | 100 |
| <b>Dense</b>   | <i>Softmax</i> | - | 5   |

### G. Confusion Matrix

Confusion matrix adalah salah satu metode untuk mengukur performa dari suatu model klasifikasi adalah dengan mencari nilai akurasi, presisi, dan recall dari suatu model[15]. Untuk mendapatkan nilai akurasi, presisi dan recall kita perlu mendapatkan terlebih dahulu banyaknya data *True positif* (TP), *False Positif* (FP), *True Negatif* (TN), dan *False Negatif* (FN). Untuk mendapatkan akurasi, presisi, dan recall dapat dilihat pada rumus dibawah ini.

$$Akurasi = \sum_{i=1}^1 \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \times 100\% \quad (1)$$

$$Presisi = \frac{\sum_{i=1}^1 TP_i}{\sum_{i=1}^1 (FP_i + TP_i)} \times 100\% \quad (2)$$

$$Recall = \frac{\sum_{i=1}^1 TP_i}{\sum_{i=1}^1 (FN_i + TP_i)} \times 100\% \quad (3)$$

### HASIL DAN DISKUSI

Hasil dari proses *training* dan *validation* pada tahap pelatihan model dengan metode *cross validation* menjadi bahan evaluasi model klasifikasi pada tahap selanjutnya. Hasil pelatihan dengan evaluasi model *cross validation* peneliti tampilkan pada tabel 2.

Tabel 2. Hasil *Cross Validation* Model CNN-LSTM

| No                      | Keterangan        | Jumlah Epoch | Akurasi | Validasi Akurasi | Loss   | Validasi Loss |
|-------------------------|-------------------|--------------|---------|------------------|--------|---------------|
| 1                       | Training Folder 1 | 65           | 0,9766  | 0,9660           | 0,0703 | 0,0846        |
| 2                       | Training Folder 2 | 50           | 0,9631  | 0,9506           | 0,0944 | 0,1356        |
| 3                       | Training Folder 3 | 55           | 0,9691  | 0,9753           | 0,0898 | 0,0691        |
| 4                       | Training Folder 4 | 59           | 0,9833  | 0,9568           | 0,0500 | 0,1167        |
| 5                       | Training Folder 5 | 48           | 0,9813  | 0,9642           | 0,0595 | 0,0951        |
| <b>Jumlah rata-rata</b> |                   |              | 0,9746  | 0,9625           | 0,0720 | 0,1002        |

Berdasarkan pada tabel 2 diatas terdapat perbedaan hasil dari *loss validation* dari kelima model *training folder*. Hasil paling baik pada *training folder* 3 mendapat nilai *loss validation* paling kecil yaitu 0,0691 pada *epoch* ke 55. Dalam konteks pelatihan model dalam *machine learning* dan *deep learning*, jumlah *epoch* merujuk pada jumlah kali keseluruhan dataset dilatih oleh model. Jika kita lihat pada tabel 2, besaran nilai *epoch* pada tiap-tiap *training folder* hasilnya berbeda. Hal ini terjadi karena peneliti menggunakan teknik *early stopping* untuk mendapatkan nilai *loss validation* paling kecil pada tiap-tiap *training folder*. *Early stopping* adalah teknik untuk menghentikan proses pelatihan apabila tidak terjadi perubahan pada nilai *validation loss*. Peneliti menentukan apabila selama lima kali *epoch* tidak ada perubahan pada hasil *validation loss* maka proses *training* akan dihentikan dan menyimpan data hasil *validation loss* yang paling kecil. Teknik *early stopping* akan mempercepat proses *training*. Peneliti mendapatkan hasil



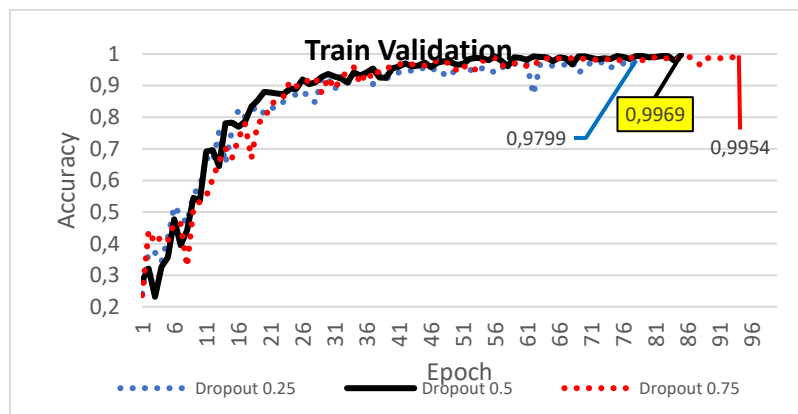
*training* yang lebih cepat daripada menunggu proses *training* berakhir berdasarkan jumlah *epoch* yang ditentukan di awal pelatihan yaitu 100 *epoch*.

#### A. Training Data

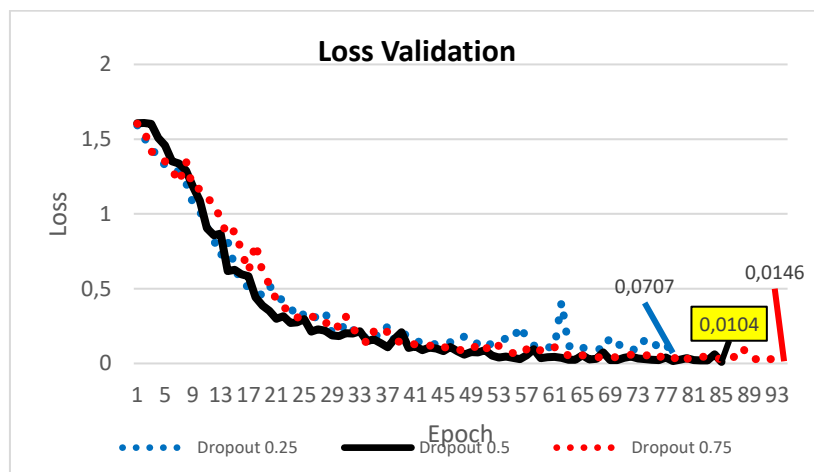
Setelah mendapatkan data hasil dari *cross validation*, tahapan selanjutnya adalah melakukan proses pelatihan dan validasi algoritma model CNN-LSTM menggunakan dataset *training folder* 3. Proses pelatihan dilakukan lebih optimal lagi dengan menambahkan *drop out*.

Tabel 3. Hasil Pelatihan Model CNN 6 Layer dengan *Dropout*

| No | Pembelajaran         | Epoch | Akurasi | Validasi Akurasi | Loss   | Validasi Loss |
|----|----------------------|-------|---------|------------------|--------|---------------|
| 1  | Tanpa <i>Dropout</i> | 55    | 0,9691  | 0,9753           | 0,0898 | 0,0691        |
| 2  | <i>Dropout</i> 0.25  | 77    | 0,9803  | 0,9799           | 0,0541 | 0,0707        |
| 3  | <i>Dropout</i> 0.5   | 84    | 0,9866  | 0,9969           | 0,0392 | 0,0104        |
| 4  | <i>Dropout</i> 0.75  | 93    | 0,9784  | 0,9954           | 0,0679 | 0,0146        |



Gambar 9. Hasil *Train Validasi*



Gambar 10. Hasil *Loss Validasi*

#### B. Testing Data

Tahap selanjutnya setelah melaksanakan proses pelatihan dan validasi adalah melaksanakan proses pengujian model pelatihan yang telah dibuat. Pengujian dilakukan dengan dua cara yaitu pengujian *offline* dan pengujian *online*. Hasil Pengujian secara



offline ditampilkan dalam bentuk *confusion matrix*. *Confusion matrix* adalah salah satu cara dan metode untuk mengukur performa dari suatu model klasifikasi adalah dengan mencari nilai precision, recall, serta nilai akurasi dari suatu model.

### 1. Pengujian Offline

Pengujian model pembelajaran secara *offline* dilakukan untuk mengetahui apakah model pembelajaran yang dibuat mampu melakukan klasifikasi video dengan baik. Data uji yang digunakan telah disiapkan dan belum pernah digunakan pada proses *training* dan *testing*. Jumlah data *offline* yang digunakan adalah 50 file video dengan durasi rata-rata 5 detik.

Tabel 4. Hasil *Confusion Matrix* Pengujian Offline

|          | Berhenti | Kanan | Kiri | Maju | Mundur |
|----------|----------|-------|------|------|--------|
| Berhenti | 897      | 1     | 2    | 0    | 0      |
| Kanan    | 6        | 880   | 14   | 0    | 0      |
| Kiri     | 9        | 18    | 870  | 0    | 3      |
| Maju     | 0        | 0     | 1    | 890  | 9      |
| Mundur   | 4        | 0     | 0    | 3    | 893    |

Berdasarkan tabel 4 maka kita bisa mengetahui tingkat akurasi, presisi, dan *recall* dari sebuah sistem pembelajaran yang diuji.

Tabel 5. Hasil Pengujian Offline

| Kelas    | Presisi | Recall | Time Respon        | Jumlah |
|----------|---------|--------|--------------------|--------|
| Berhenti | 97,9%   | 99,6%  | 180 ms             | 900    |
| Kanan    | 97,9%   | 97,8%  | 180 ms             | 900    |
| Kiri     | 98,1%   | 96,7%  | 178 ms             | 900    |
| Maju     | 99,7%   | 98,9%  | 180 ms             | 900    |
| Mundur   | 98,7%   | 99,2%  | 179 ms             | 900    |
|          | 98,4%   | 98,4%  | $\bar{x} = 179$ ms | 4500   |
|          | 96,9%   |        |                    |        |

Berdasarkan hasil pengujian *offline* didapatkan nilai akurasi sebesar 96,9% dan rata-rata time respon sebesar 179 ms. Nilai presisi paling besar pada kelas maju sebesar 99,7% dan nilai presisi paling kecil pada kelas berhenti dan kanan sebesar 97,9%. Nilai recall atau sensitivitas paling besar pada kelas berhenti sebesar 99,6% dan nilai recall paling kecil pada kelas kiri sebesar 96,7%. Sistem yang dibuat mampu melakukan prediksi sebanyak  $\pm 6$  frame per second (fps)



Gambar 11. Hasil Pengujian Secara Offline

## DAFTAR PUSTAKA

- [1] A. Mais, *Media Pembelajaran Anak Berkebutuhan Khusus*. 2016.
- [2] BPS, *Penduduk Indonesia hasil SUPAS 2015*. Badan Pusat Statistik Indonesia, 2015. [Online]. Available: <http://repositorio.unan.edu.ni/2986/1/5624.pdf>
- [3] T. I. Kusumawati, “Komunikasi Verbal Dan Nonverbal,” *J. Pendidik. dan Konseling*, vol. 6, no. 2, pp. 83–98, 2016.
- [4] A. Sunyoto and A. Harjoko, “Review Teknik, Teknologi, Metodologi dan Implementasi Pengenalan Gestur Tangan Berbasis Visi,” *Semin. Nas. Apl. Teknol. Inf.*, p. H-7, 2014, [Online]. Available: <http://jurnal.uui.ac.id/Snati/article/view/3290>
- [5] W. Dar, *Sistem kendali Cerdas*. Yogyakarta: Graha Ilmu, 2011.
- [6] S. Anwar, S. K. Sinha, S. Vivek, and V. Ashank, “Hand gesture recognition: A survey,” *Lect. Notes Electr. Eng.*, vol. 511, no. 3, pp. 365–371, 2019, doi: 10.1007/978-981-13-0776-8\_33.
- [7] C. a. Pickering, K. J. Burnham, and M. J. Richardson, “A research study of hand gesture recognition technologies and applications for human vehicle interaction,” *3rd Conf. Automot. ...*, pp. 1–15, 2007, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.7688&rep=rep1&type=pdf>
- [8] A. Anam, K., & Saleh, “Intelligent wheelchair control system based on finger pose recognition,” 2020, pp. 257–261. doi: 10.23919/EECSI50503.2020.9251907.
- [9] A. Nur Sasongko, “Kendali Model Kursi Roda dengan Electromyograf dan Accelerometer Menggunakan Metode Jaringan Saraf Tiruan,” *ALINIER J. Artif. Intell. Appl.*, vol. 1, no. 2, pp. 59–68, 2020, doi: 10.36040/alinier.v1i2.2969.
- [10] D. Sukoco, “Accelerometer Gy-61 sebagai Pengendali Kursi Roda untuk Penyandang Disabilitas Berbasis Android,” pp. 259–262, 2019.
- [11] S. Ashok, “High-level hands-free control of wheelchair—a review,” *J. Med. Eng. Technol.*, vol. 41, no. 1, pp. 46–64, 2017, doi: 10.1080/03091902.2016.1210685.
- [12] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.
- [13] D. Hubel and T. Wiesel, “RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT ’ S STRIATE CORTEX By D . H . HUBEL \* AND T . N . WIESEL \* From the Wilmer Institute , The Johns Hopkins Hospital and In the central nervous system the visual pathway from retina to striate cortex provides an,” *J. Physiol.*, vol. 148(3), 57, pp. 574–591, 1959.
- [14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [15] Eko Prasetyo, *Data Mining: Konsep dan Aplikasi menggunakan MATLAB*, 1st ed. Indonesia: CV Andi Offset, 2012.