Jurnal Rekayasa Elektrika Vol. 17, No. 1, Maret 2021, hal. 29-35
ISSN. 1412-4785; e-ISSN. 2252-620X, Terakreditasi RISTEKDIKTI No. 36b/E/KPT/2016
DOI: 10.17529/jre.v17i1.16376

Sistem Navigasi Kursi Roda Elektrik untuk Pasien Penyandang Cacat Fisik Menggunakan Metode Convolutional Neural Network

Sutikno, Khairul Anam, dan Azmi Saleh
Jurusan, Teknik Elektro, Fakultas Teknik, Universitas Jember
Kampus Tegalboto, Jl. Kalimantan No. 37 Krajan Timur, Sumbersari, Jember 68121
e-mail: khairul@unej.ac.id

***Abstrak*—Pasien cacat fisik seperti kehilangan kaki atau mengalami kelumpuhan, akan mengalami kesulitan untuk bergerak dari satu tempat ke tempat yang lain. Mereka membutuhkan seseorang atau perangkat yang dapat membantu mereka. Salah satu yang sering digunakan oleh pasien adalah kursi roda. Tujuan utama penulisan artikel ini adalah merancang sebuah sistem navigasi kursi roda elektrik yang dapat dikendalikan dengan perintah suara menggunakan metode *Convolutional Neural Network* (CNN). CNN digunakan sebagai metode utama untuk mengidentifikasi perintah yang akan ditanam pada mikrokontroler Raspberry Pi. *Data* suara direkam kemudian dikonversi ke gambar *spectrogram* sebelum diumpankan ke CNN. Metode ini terbukti baik dalam pengenalan perintah suara dengan akurasi diatas 90 %. Ada lima perintah suara yang berbeda, yaitu maju, mundur, kiri, kanan, dan berhenti (*stop*). Hasil eksperimen awal menunjukkan bahwa kursi roda elektrik yang dirancang bergerak sesuai dengan perintah yang diberikan.**

Kata kunci: ***pasien, cacat fisik, kursi roda elektrik, convolutional neural network (cnn), raspberry pi***

Abstract—Patients with physical disabilities, such as losing a leg or experiencing paralysis, will have difficulty moving from one place to another. As a result, they need someone or a device that can help them to move. One that is often used by patients is a wheelchair. This study proposes an electric wheelchair navigation system that can be controlled by voice commands using the Convolutional Neural Network (CNN) method. CNN is used as the main method for identifying commands embedded on the Raspberry Pi microcontroller. The recorded voice data is then converted to spectrogram images before being fed to CNN. This method is proven to be better in voice command recognition with an accuracy of above 90%. There are five different voice commands: forward, backward, left, right, and stop. Preliminary experimental results indicate that the electric wheelchair is able to move according to the commands given.

Keywords: ***patient, physical disabilities, electric wheelchair, convolutional neural network (cnn), raspberry pi***

I. PENDAHULUAN

Salah satu yang menjadi tolok ukur kemajuan dari suatu negara dapat dilihat dari perkembangan teknologi yang dihasilkan oleh negara tersebut [1]. Salah satu contoh teknologi yang dapat menggantikan dan membantu dalam mobilitas adalah kursi roda elektrik, terutama bagi para pasien yang mengalami cacat fisik. Menurut survei yang dilakukan oleh *World Health Organization* (WHO), mengatakan bahwa lebih dari 1 miliar penduduk dunia mengalami kecacatan dalam berbagai macam dan jenis yang menyumbang sekitar 15% dari populasi dunia [2]. Sedangkan jumlah disabilitas yang ada di indonesia sekitar 11,5 juta jiwa dimana diantaranya 3 juta adalah penyandang cacat fisik [3].

Pasien dengan gangguan pada anggota tubuh seperti kehilangan kaki, tangan, mengalami kelumpuhan yang disebabkan oleh berbagai faktor seperti kecelakaan, bencana alam, kesehatan, faktor keturunan, dan lain-lain menyebabkan mereka kesulitan dalam bergerak. Oleh sebab itu mereka cenderung membutuhkan alat bantu untuk mempermudah gerak mereka. Alat bantu yang sering digunakan adalah kursi roda biasa yang membutuhkan kekuatan otot untuk bergerak Bagi pasien yang kehilangan kaki, hal ini tentu menjadi masalah baru bagi mereka. Bahkan jika dipaksa mereka harus menggunakan tenaga ekstra, sehingga membuat mereka menjadi lelah [4]. Begitu pula dengan pasien yang mengalami kelumpuhan atau gangguan sistem saraf, mereka akan kesulitan bahkan tidak bisa menggerakkan kursi roda secara normal. Faktor ini terjadi karena kaki dan tangan mereka tidak dapat beroperasi dengan normal untuk memindahkan kursi roda.

Pada penelitian sebelumnya, banyak artikel yang membahas tentang kontrol kursi roda menggunakan perintah suara, seperti pada penelitian [5]-[7]. Penelitian sebelumnya menggunakan *microcontroller* Arduino [5] dan

Received 04 April 2020; Revised 04 October 2020; Accepted 07 October 2020

modul HM 2007 [6] sebagai modul suara, serta *Artificial Neural Network* (ANN) yang beberapa diantaranya dalam bentuk *prototype* [7]. Akan tetapi tingkat akurasi yang dihasilkan dari penelitian sebelumnya masih kurang baik dan belum bisa membedakan perintah pada saat terdapat *noise* di sekitar pengguna, dikarenakan *dataset* yang digunakan sebagai masukan pada sistem tanpa *noise*. Oleh sebab itu, dibutuhkan suatu metode yang lebih akurat untuk mendapat hasil yang lebih baik.

Salah satu poin utama dari penulisan artikel ini adalah penggunaan metode yang masih jarang digunakan untuk navigasi kursi roda elektrik. Salah satu metode yang menjanjikan adalah *Convolutional Neural Network* (CNN). Kontribusi utama dari artikel ini adalah penerapan metode CNN yang tertanam pada Raspberry Pi untuk menafsirkan perintah suara dari pengguna. Kelebihan dari kursi roda ini adalah, selain memiliki tingkat akurasi yang baik, juga dapat mengenali perintah suara dengan tingkat intonasi yang berbeda serta terhadap adanya *noise* .

Kursi roda elektrik ini memiliki peranan penting dalam perkembangan masyarakat masa depan. Berdasarkan kondisi ini, kursi roda elektrik menggunakan teknologi identifikasi pengenalan suara dapat membantu pengguna secara langsung dan memudahkan dalam menggerakkan kursi roda untuk memfasilitasi keterbatasan pasien penyandang cacat fisik. Dalam sistem yang disajikan, sistem pengenalan suara digunakan sebagai antarmuka pengguna dalam mengoperasikan sistem. Pasien yang secara khusus tidak bisa berjalan, diakibatkan oleh cacat atau lumpuh sehingga mereka harus menggunakan kursi roda untuk pekerjaan sehari-hari yang dapat berpindah hanya menggunakan perintah suara mereka [8]. Dengan sistem kontrol kursi roda elektrik ini, para pasien cacat fisik akan menjadi lebih mudah dan mandiri. Sistem kontrol kursi roda elektrik ini menggunakan sistem pengenalan suara untuk memicu dan mengendalikan semua gerakan kursi roda [9].

II. METODE

A. Perekaman dan Pengolahan Data Suara

Proses perekaman suara menggunakan mikrofon/*audio* standar sebagai sinyal *input* suara dengan bantuan *software sound recorder pro* yang dapat diunduhdari *play store*. *Sample rate* diatur menjadi 16 kHz dan *bitrate* 128 kbit/s dengan saluran *audio mono* dan sumber *audio* mikrofon. Sedangkan untuk proses penyimpanan digunakan format *wav* danuntuk pendeteksian tingkat kebisingan pada area sekitar digunakan *sound meter*. Perintah suara terdiri dari lima masukan dan lima perintah reaksi. Perekaman dilakukan dengan intonasi suara yang beragam seperti suara rendah, sedang, dan tinggi dengan tingkat kebisingan (*noise*) yang berbeda seperti pada ruang tertutup dan terbuka. Tingkat kebisingan berkisar antara 35.0 – 85.8 dB. Selain itu terdapat lima perintah suara yang direkam, yaitu maju, mundur, kiri, kanan, dan berhenti (*stop*) [10] dengan durasi perekaman setiap kata adalah 1–2 s.

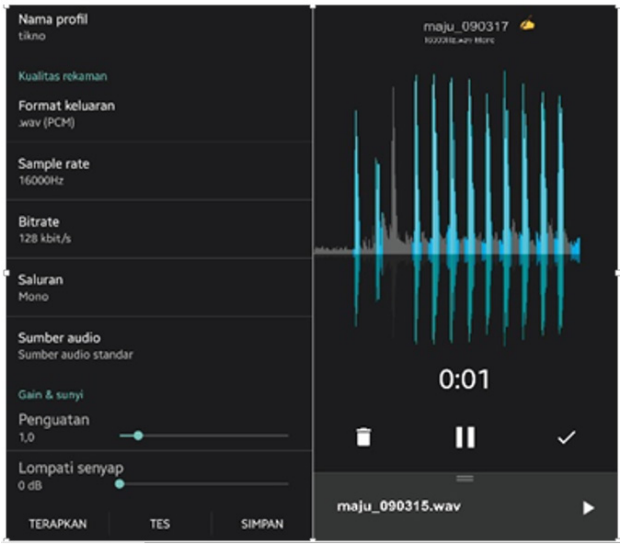
Pengaturan (*setting*) pada aplikasi *sound recorder pro* ditunjukkan pada Gambar 1, Salah satu kelebihan dari aplikasi ini adalah kemampuan perekaman secara terus menerus, sehingga lebih efisien dari segi waktu. Perekaman dilakukan di ruang tenang dan tertutup dengan masing-masing sampel suara sebanyak 120 dan total *data* suara pada ruang tenang sebanyak 600 *data*. Untuk wilayah perkantoran, pusat perbelanjaan, jalan raya dan ruang publik masing-masing memiliki 600 *data* dengan total keseluruhan *data* yang direkam adalah 3.000 *data* suara.

Setiap kata yang direkam akan memiliki ukuran dan durasi yang berbeda, oleh sebab itu untuk mempermudah proses selanjutnya, dibutuhkan proses penyamaan ukuran setiap panjang durasi *data* suara dengan menggunakan aplikasi *sox-sound exchange.* Proses ini disebut dengan *trim* dan *padding*. Pada intinya, *file* suara yang ada tidak dikurangi bahkan *noise*-nya pun tidak dihilangkan , melainkan suara hanya dipotong pada batas ambang tertentu dengan tujuan untuk menghilangkan bagian yang tidak diperlukan. Akan tetapi, pada tahapan ini perlu dilakukan proses *denoising* (menghilangkan *noise*) agar model dapat berkerja pada lingkungan yang terdapat *noise*. Langkah pertama dari proses ini adalah *file* suara di-*trim* ke depan selama 0.1 s, di-*trim* ke belakang selama 0.1 s, *audio file* di-pad selama 2 s, kemudian potong menjadi 1 s. Dengan demikian, semua *data* suara akan memiliki ukuran sama yaitu 32 KB dan durasi sama yakni 1 s. *Data* yang sudah mengalami proses tersebut di atas, selanjutnya disebut sebagai *dataset*.

B. Short-time Fourier Transform (STFT)

Sebelum diumpankan ke CNN, *data* suara harus divisualisikan terlebih dahulu. Gambar 2 menunjukkan hasil plot gelombang suara dari perintah kanan, kiri, maju, mundur, dan berhenti. Gelombang inilah yang nantinya akan diubah terlebih dahulu menjadi gambar *spectrogram*.

Metode pengenalan suara yang paling sering

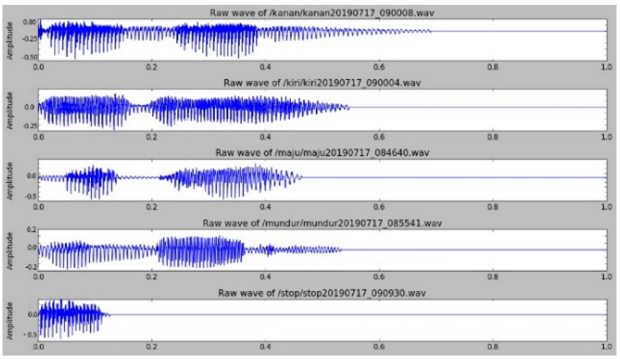


Gambar 1. Pengaturan dan perekaman pada aplikasi *sound recorder pro*

digunakan adalah *Mel Frequency Cepstrum Coefficients* (MFCC). Metode ini merupakan salah satu metode yang sering dipakai dalam bidang teknologi pengenalan suara baik pada *speaker recognition* ataupun *speech recognition*. Pengenalan suara dapat dilakukan dengan cara melakukan *feature extraction* (ekstraksi fitur) yang merupakan proses konversi sinyal suara/*audio* menjadi beberapa parameter [11], serta dengan menggunakan metode *spectrogram*. Metode *spectrogram* merupakan hasil dari representasi sinyal *audio* dalam domain frekuensi. Untuk menghasilkan *spectrogram*, pada penelitian ini digunakan STFT yang berfungsi untuk mengonversi *data* mentah dari *audio window* atau sinyal suara menjadi *spectrogram*. *Spectrogram magnitude* yang dihasilkan kemudian dipetakan ke skala mel, sehingga diperoleh *mel-spectrogram*. Skala frekuensi ini berperan menekan frekuensi rendah di atas frekuensi tinggi, yang mirip dengan kemampuan persepsi telinga manusia [12]. Hasil representasi dari sinyal suara menjadi gambar *spectrogram* dengan menggunakan metode STFT ditunjukkan pada Gambar 3. Selanjutnya, *spectrogram* inilah yang akan menjadi *input* pada CNN.

C. CNN Classifier

CNN adalah salah satu jenis jaringan saraf yang biasa digunakan dalam pengolahan *data* citra [13]. CNN juga dapat digunakan untuk mendeteksi dan mengenali objek berupa gambar. CNN tidak jauh berbeda dari jaringan saraf biasa yang terdiri dari *neuron* yang memiliki bobot, bias dan fungsi aktivasi. CNN termasuk dalam jenis *deep neural network* karena memiliki tingkat jaringan lebih banyak dan secara luas banyak diterapkan pada *data* citra. Metode CNN memiliki dua tahapan, yaitu tahap klasifikasi menggunakan *feedforward* dan tahap pembelajaran menggunakan *backpropagation*. Cara kerja CNN memiliki kesamaan dengan MLP, akan tetapi setiap *neuron* pada CNN disajikan dalam bentuk dua dimensi. Berbeda halnya dengan MLP, dimana setiap *neuron* hanya memilki satu dimensi. Operator konvolusi digunakan untuk mengetahui sinyal *input* yang berupa suara dan beberapa ekstraksi gambar tambahan. Untuk pengklasifikasian CNN, terdapat 5 *input* berupa gambar *spectrogram* yang merupakan hasil representasi dari perintah kanan, kiri, maju, mundur dan berhenti (*stop*). Secara umum arsitektur utama dari CNN terdiri dari *input*, *feature learning layer*, dan *classification*.



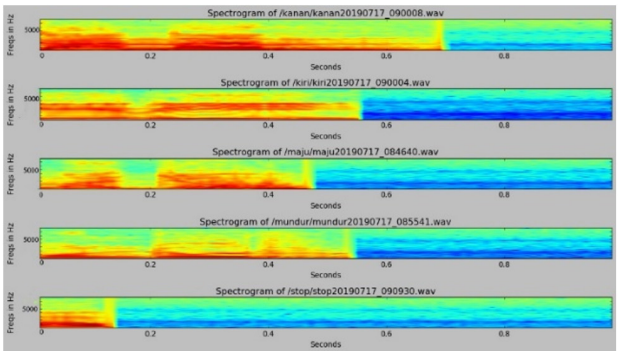
Gambar 2. *Plot* gelombang perintah kanan, kiri, maju, mundur, dan berhenti (*stop*)

Model arsitektur dari metode CNN yang diusulkan pada penelitian ini terdiri dari *input*, 3 lapisan konvolusi, 3 lapisan (*layer*) *pooling*, dan lapisan *fully connected*, seperti yang ditunjukkan pada Gambar 4. Arsitektur tersebut diperoleh setelah melalui beberapa percobaan. Model CNN dibuat menggunakan bahasa pemrograman Python dengan Keras. Adapun ukuran citra masukan dalam CNN dapat dinyatakan sebagai:

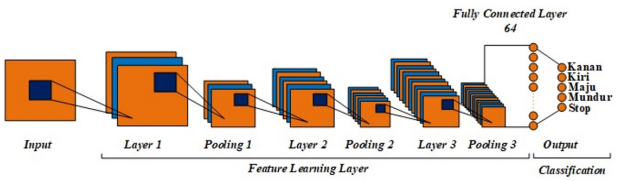
$$w_1 \times h_1 \times d_1, \tag{1}$$

dimana w_1 dan h_1 secara berurutan merupakan lebar dan tinggi citra, sedangkan d_1 adalah jumlah kanal *Red Green Blue* (RGB) yang mana dalam CNN sering disebut *depth* atau kedalaman citra masukan. *Input* berupa gambar *spectrogram* dengan ukuran 177 × 98 piksel yang merupakan *channel* RGB, *batch size* = 32. Selanjutnya dilanjutkan pada konvolusi lapisan (*layer*) pertama dengan ukuran *kernel* 2 × 2 serta *filter* 64 *layer*, dengan aktivasi *ReLU*. Lapisan konvolusi ini merupakan salah satu blok utama dari sebuah CNN, dikarenakan sebagian besar komputasi dilakukan pada blok ini. Sedangkan 2×2 merupakan bidang reseptif untuk setiap *neuron* dan menyatakan bahwa *filter* yang digunakan berukuran 2 × 2. Jumlah parameter pada konvolusi pertama yaitu 320. Bidang reseptif tersebut akan ditelusuri secara tumpang tindih parsial dimana setiap *neuron* akan berbagi bobot koneksi (*weight sharing*). Aktivasi *ReLU* digunakan untuk mengenalkan *non*-linearitas dan meningkatkan representasi dari model. Aktivasi ini mengaplikasikan fungsi f(x) = max(0, x)[14] dan menggunakan *stride* = 1 dengan *zero padding* sebesar:

$$P = \frac{(F-1)}{2}, \tag{2}$$



Gambar 3. *Spectrogram* perintah kanan, kiri, maju, mundur, dan berhenti (stop



Gambar 4. Model arsitektur CNN

dimana P adalah ukuran *padding* dan F adalah ukuran bidang reseptif.

Selesai proses pada konvolusi pertama, masuk pada tahap *pooling layer* 1 yang memiliki ukuran 89 × 49 dan *filter* 64 *layer*. Pada *pooling* pertama menggunakan *max pooling* dengan *kernel* 2 × 2 dan *stride* = 2, tujuannya adalah untuk melakukan reduksi sampel (*down sampling*), serta *data* menjadi lebih kecil, mudah mengontrol *overfitting* dan mudah dikelola. Dengan adanya *max pooling* ini, maka *input* yang awalnya berukuran 177 × 98 piksel menjadi 89 × 49 piksel.

Setelah konvolusi pertama selesai, selanjutnya masuk pada konvolusi kedua yang memiliki ukuran dimensi *input shape* 89 × 49, *filter* 128 *layer*, *kernel* 2 × 2, *stride* = 1, jumlah parameter 32.896, dan aktivasi *ReLU*. Pada *pooling layer* 2 menggunakan *max pooling* dengan *input shape* 45 × 25, *filter* 128 *layer*, *kernel* 2 × 2, dan *stride* = 2. Hasil proses *pooling* 2, akan menghasilkan *input shape* dengan ukuran 45 × 25 piksel.

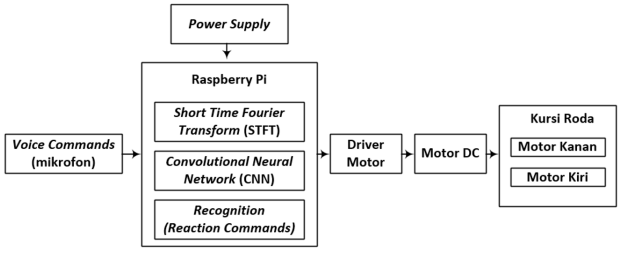
Setelah konvolusi dan *pooling* tahap kedua selesai, masuk pada konvolusi dan *pooling* tahap ketiga. Konvolusi *layer* ketiga menggunakan *input shape* ukuran 45×25 piksel, *filter* 64 *layer*, *kernel* 2 × 2, *stride* = 1, dan aktivasi *ReLU*. Total parameter adalah 32.832 dengan ukuran *input shape* 23 × 13 piksel, *filter* 64 *layer*, dan *stride* = 2. Hasil proses konvolusi pada *pooling layer* 3 berupa *input shape* 23 × 13 piksel.

Setelah proses konvolusi dan *pooling layer* sebanyak 3 kali, selanjutnya akan masuk ke tahap *flatten layer* yang berfungsi merubah hasil dari *pooling* 3 menjadi vektor. Lapisan ini memperoleh *input* dari proses sebelumnya agar dapat menentukan fitur yang paling tepat dengan kelas tertentu. Fungsi dari lapisan ini yaitu menyatukan semua node menjadi satu dimensi [15]. Vektor yang dihasilkan pada *flatten layer* sebanyak 19.136. Hasil dari *flatten layer* ini dimasukkan satu per satu pada *dense* yang berjumlah 64 *layer* dengan jumlah parameter yang dihasilkan yaitu sebanyak 95.685 parameter. Selanjutnya proses *dropout* pada 64 unit *dense* menjadi 5, sehingga akan menghasilkan *dense* sebanyak 5 klasifikasi. Terakhir adalah volume keluaran (*output*) dapat dihitung dengan formula:

$$w_2 \times h_2 \times d_2, \tag{3}$$

dimana:

$$w_2 = \frac{(w_1 - F + 2P)}{S} + 1 \tag{4}$$



Gambar 5. Blok diagram sistem navigasi kursi roda elektrik

$$h_2 = \frac{(h_1 - F + 2P)}{S} + 1 \tag{5}$$

$$d_2 = K, \tag{6}$$

K adalah jumlah *filter*, F ukuran bidang reseptif, S lebar langkah atau *stride*, dan P adalah *zero padding*.

D. Blok Diagram dan Desain Perangkat Keras

Blok diagram sistem secara garis besar dan desain perangkat keras yang diusulkan pada penelitian ini secara berurutan diilustrasikan pada Gambar 5 dan Gambar 6. Melalui ilustrasi ini, diharapkan lebih mempermudah untuk memahami sistem secara keseluruhan. Dalam proses perancangan kursi roda elektrik menggunakan perintah suara, ada beberapa komponen/bahan utama yang digunakan, yaitu kursi roda elektrik, catu daya/power *supply*, modul suara/mikrofon, Raspberry Pi 3, Arduino Uno, motor DC, *driver* motor DC, dan *toolset* lainnya. Raspberry Pi 3 memilki spesifikasi RAM 1 GB, CPU *Quad Core* 1.2GHz *Broadcom* BCM2837 64 bit sehingga hanya cocok digunakan untuk menjalankan model prediksi yang dilatih sebelumnya di komputer/laptop.

III. HASIL DAN PEMBAHASAN

Penelitian ini mengusulkan sistem navigasi kursi roda elektrik dengan perintah suara untuk pasien yang memiliki keterbatasan fisik, Perintah suara akan diklasifikasi menggunakan metode CNN. Perintah suara disimpan dalam bentuk *dataset* yang dikelompokkan menjadi dua, yaitu sebagai *data training* dan *data testing*.

A. Data Training dan Data Testing

Jenis-jenis *data* yang harus tersedia untuk mengetahui tingkat akurasi dan performa CNN yang dibangun adalah *data training* dan *data testing*. *Data training* digunakan untuk melatih algoritma yang dibuat, sedangkan *data testing* digunakan untuk mengetahui performa algoritma

Gambar 6. Desain perangkat keras

yang sudah mengalami pelatihan sebelumnya ketika algoritma tersebut menemukan *data* baru yang belum pernah dikenali sebelumnya. Hal ini sering disebut sebagai generalisasi. Hasil dari pelatihan tersebutdisebut sebagai model. Semakin banyak *data training* yang dijadikan sebagai *data* pelatihan, maka model yang dihasilkan semakin bagus terhadap pengenalan pola perintah suara. Perbandingan antara *data training* dan *data testing* ditunjukkan pada Tabel 1.

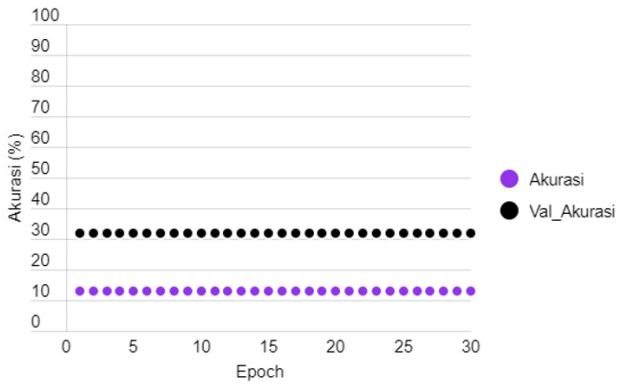
Pada ruang tenang, terdapat 600 *data* suara yang terdiri dari 120 perintah kanan, 120 kiri, 120 maju, 120 mundur dan 120 *stop*. Dari masing-masing *data* 120 tersebut, terbagi dua pertiga sebagai *data training* dan sepertiga *data testing*. Jadi tiap-tiap perintah suara ada 80 *data training* dan 40 *data testing*. Karena ada 5 macam perintah suara, maka total pada tiap lokasi yaitu 80 × 5 = 400 *data training*, dan 40 × 5 = 200 *data testing*. Begitu pula dengan lokasi perkantoran, pusat perbelanjaan, jalan raya, dan ruang publik, jumlah keseluruhan *data training* sebanyak 2.000 dan *data testing* 1.000.

B. Training dan Testing Data

Pelaksanaan *training* dan *testing data* menggunakan *tools google colab* yang sudah menyediakan GPU gratis untuk *user* sehingga mempermudah dan mempercepat proses *training* yang terdiri dari 2.000 *data training* yang akan diolah. Percobaan pertama peneliti mencoba menggunakan *Artificial Neural Network* (ANN) dengan menggunakan keras. Pada ANN terdapat beberapa lapisan, yaitu lapisan pertama berupa lapisan *input* (*input layer*) yang berfungsi menghubungkan jaringan pada sumber *data* dan meneruskan *data* pada jaringan berikutnya.

Tabel 1. Perbandingan *data training* dan *data testing*

No	Perintah	Jumlah	Data training	Data testing
1	Kanan	600	400	200
2	Kiri	600	400	200
3	Maju	600	400	200
4	Mundur	600	400	200
5	Stop	600	400	200
Jumlah		3.000	2.000	1.000



Gambar 7. Akurasi dan validasi akurasi sistem menggunakan ANN

Lapisan kedua yaitu *hidden layer*, sering disebut sebagai lapisan tersembunyi, yang memungkinkan jaringan saraf untuk menghitung fungsi yang tidak dapat dipisahkan secara linier. Lapisan terakhir yaitu *output layer* yang memiliki prinsip kerja seperti *hidden layer*. Pada lapisan ini peneliti menggunakan fungsi *sigmoid* dan *Adam optimizer*. Pada proses awal digunakan *early stopping* yang bertujuan untuk menghentikan proses *training* ketika validasi akurasi menurun. *Input shape* yang digunakan berukuran 177 × 98 dengan *epoch* sebanyak 30.[H1]

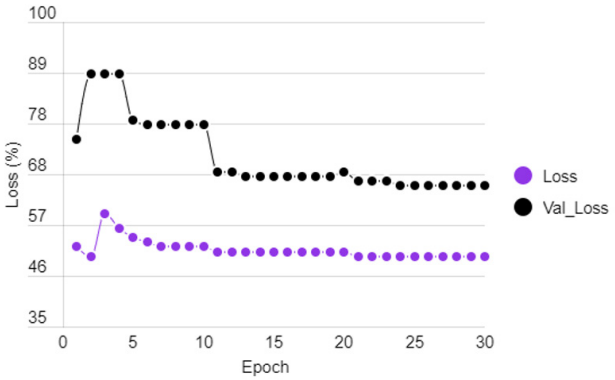
Tabel 2 merupakan hasil pengujian sistem usulan menggunakan metode ANN, terlihat bahwa dari *epoch* pertama hingga *epoch* ke 30 tidak ada perubahan yang signifikan dengan akurasi yang diperoleh rata-rata 13.31 % dan validasi akurasi 32.12 %, *loss* sebesar 50.28 % dan validasi *loss* 65.07 %.

Tingkat akurasi dan *loss* sistem dengan menggunakan metode ANN secara berurutan ditunjukkan pada Gambar 7 dan Gambar 8 . Untuk tingkat akurasi sistem yang dihasilkan menggunakan metode ANN tidak begitu bagus

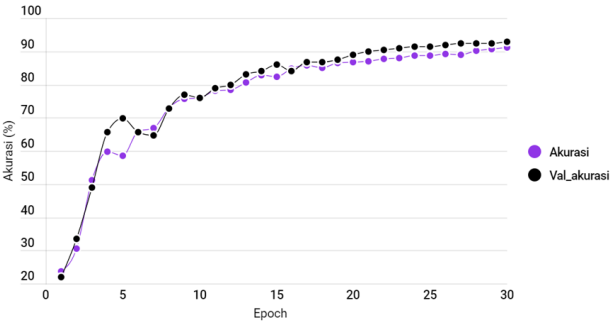
Tabel 2. Data pengujian menggunakan ANN

Epoch	*Loss*	*Val_Loss*	*Accuracy*	*Val_ Accuracy*
1	52.07 %	75.66 %	13.37 %	32.71 %
2	50.30 %	89.99 %	13.07 %	32.17 %
3	59.25 %	89.13 %	13.14 %	32.40 %
4	56.36 %	89.24 %	13.09 %	32.50 %
5	54.65 %	79.51 %	13.91 %	32.54 %
6	53.82 %	78.38 %	13.37 %	32.12 %
7	52.90 %	78.16 %	13.06 %	32.88 %
8	52.65 %	78.12 %	13.14 %	32.67 %
9	52.39 %	78.02 %	13.09 %	32.58 %
10	52.18 %	78.36 %	13.93 %	32.67 %
11	51.75 %	68.89 %	13.37 %	32.58 %
12	51.54 %	68.80 %	13.07 %	32.62 %
13	51.48 %	67.89 %	13.14 %	32.79 %
14	51.14 %	67.88 %	13.09 %	32.54 %
15	51.20 %	67.34 %	13.91 %	32.54 %
16	51.13 %	67.14 %	13.37 %	32.33 %
17	51.38 %	67.40 %	13.27 %	32.29 %
18	51.01 %	67.53 %	13.14 %	32.58 %
19	51.10 %	67.66 %	13.99 %	32.04 %
20	51.16 %	68.81 %	13.22 %	32.79 %
21	50.90 %	66.16 %	13.07 %	32.83 %
22	50.90 %	66.67 %	13.01 %	32.62 %
23	50.90 %	66.07 %	13.14 %	32.23 %
24	50.78 %	65.37 %	13.09 %	32.22 %
25	50.93 %	65.81 %	13.91 %	32.12 %
26	50.81 %	65.75 %	13.37 %	32.12 %
27	50.85 %	65.97 %	13.07 %	32.12 %
28	50.78 %	65.04 %	13.14 %	32.12 %
29	50.70 %	65.40 %	13.09 %	32.12 %
30	50.28 %	65.07 %	13.31 %	32.12 %

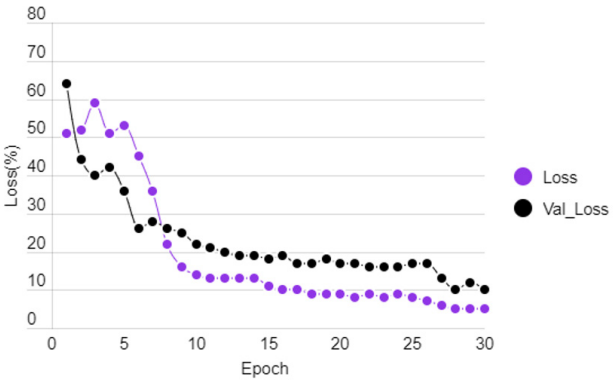
karena pada *fully connected layer* dengan ANN tidak cocok dalam memproses citra gambar. Hal ini diakibatkan oleh jumlah parameter yang sangat besar. Sebagai contoh yaitu pada ANN yang digunakan, dengan *input* berukuran 177 × 98 dengan 128 *neuron* pada lapisan pertama *fully connected* ANN, membutuhkan 177 × 98 × 128 = 2.220.416 dengan total parameter 2.254.085 yang harus dioptimasi pada *layer* pertama. Jika kita menambahkan *layer* maka secara otomatis jumlah parameter tersebut akan bertambah pula, sehingga akan menyebabkan *overfitting*. Dengan demikian dapat disimpulkan bahwa metode ANN tidak cocok untuk digunakan sebagai metode *image recognition*. Solusi untuk mengatasi hal tersebut adalah dengan mengekstrak fitur dan menurunkan dimensi tanpa menghilangkan karakteristik dari *input* yaitu dengan menggunakan metode CNN.



Gambar 8. *Loss* dan validasi *loss* sistem menggunakan ANN



Gambar 9. Akurasi dan validasi akurasi sistem menggunakan CNN



Gambar 10. *Loss* dan validasi *loss* sistem menggunakan CNN

Hasil pengujian menggunakan metode CNN ditunjukkan pada Tabel 3 dimana setiap *epoch* menampilkan *loss*, validasi *loss*, akurasi, dan validasi akurasi dengan jumlah *epoch* yang digunakan sebanyak 30. Pada *epoch* pertama tingkat akurasi yang dihasilkan hanya 23.71 %, pada *epoch* kedua dan seterusnya mengalami peningkatan. Meskipun tidak semua mengalami kenaikan dibandingkan dengan *epoch* sebelumnya, justru terdapat beberapa *epoch* yang lebih kecil. Tingkat akhir akurasi yang di peroleh menggunakan metode CNN ini adalah 91.25 % dan validasi akurasi 92.87 %, seperti ditunjukkan pada Gambar 9.

Pada Gambar 10, diperlihatkan hasil dari nilai *loss* dan validasi *loss* sistem menggunakan metode CNN. Pada awal *training*, *loss* yang dihasilkan sangat besar, yaitu 51.23 % dengan validasi *loss* mencapai 64.55 %. Namun pada *epoch* berikutnya nilai *loss* dan validasi *loss*-nya semakin kecil hingga tercatat *loss* pada *epoch* 30 menjadi 5.65 % dengan validasi *loss* 10.45 %. Tingkat akurasi dan *loss* pada CNN juga dipengaruhi oleh banyaknya *data*

Tabel 3. Data pengujian menggunakan CNN

Epoch	*Loss*	*Val_Loss*	*Accuracy*	*Val_Accuracy*
1	51.23 %	64.55 %	23.71 %	21.98 %
2	52.45 %	44.96 %	30.67 %	33.38 %
3	59.95 %	40.99 %	51.09 %	48.96 %
4	51.60 %	42.96 %	59.87 %	65.62 %
5	53.01 %	36.74 %	58.59 %	69.79 %
6	45.46 %	26.01 %	65.87 %	65.62 %
7	36.70 %	28.33 %	66.99 %	64.62 %
8	22.87 %	26.39 %	73.08 %	72.88 %
9	16.45 %	25.85 %	75.76 %	76.96 %
10	14.21 %	22.11 %	75.88 %	75.96 %
11	13.69 %	21.13 %	78.22 %	78.96 %
12	13.27 %	20.89 %	78.48 %	79.96 %
13	13.66 %	19.50 %	80.65 %	82.96 %
14	13.30 %	18.49 %	82.86 %	83.98 %
15	11.02 %	18.40 %	82.45 %	85.96 %
16	10.07 %	19.65 %	84.87 %	83.97 %
17	10.20 %	17.40 %	85.74 %	86.87 %
18	9.95 %	17.25 %	84.98 %	86.64 %
19	9.22 %	18.27 %	86.48 %	87.44 %
20	9.14 %	17.19 %	86.74 %	88.98 %
21	8.38 %	17.02 %	86.98 %	90.01 %
22	9.79 %	16.15 %	87.65 %	90.45 %
23	8.75 %	16.20 %	87.98 %	90.89 %
24	9.35 %	16.15 %	88.63 %	91.51 %
25	8.28 %	17.26 %	88.57 %	91.43 %
26	7.25 %	17.17 %	89.12 %	91.81 %
27	6.24 %	13.31 %	88.98 %	92.43 %
28	5.22 %	10.56 %	90.16 %	92.36 %
29	5.21 %	12.15 %	90.56 %	92.33 %
30	5.65 %	10.45 %	91.25 %	92.87 %

yang digunakan dalam proses *training* dan juga algoritma yang dibangun. Setelah hasil *training* dan *testing data* dilakukan, tahap selanjutnya adalah menyimpan model yang sudah dibangun. Model tersebut kemudian di-*compile* ke Raspberry Pi 3. Pada tahap ini, prosesnya tidak terlalu rumit dikarenakan proses *training* sudah dilakukan di-*google colab*, sehingga hanya model saja yang di-*upload* para Raspberry Pi 3.

IV. KESIMPULAN

Dalam studi ini, kami mengembangkan sebuah sistem navigasi kursi roda menggunakan metode CNN yang dilengkapi dengan modul suara untuk penyandang cacat fisik. Kursi roda ini secara khusus dirancang untuk mempermudah mereka dalam bergerak tanpa harus menghabiskan banyak energi. Kami menggunakan motor sebagai *driver*, modul mikrofon sebagai alat untuk mengaktifkan suara dan Raspberry Pi 3 sebagai sistem kontrol. Kursi roda dapat bergerak sesuai dengan perintah suara dengan tingkat akurasi diatas 90 %. Tingkat akurasi dipengaruhi oleh jumlah *dataset* yang digunakan, serta algoritma yang dipakai.

REFERENSI

[1] Chaidir, K. Anam, and Rahardi, "Lane tracking pada robot beroda holonomic menggunakan pengolahan citra," *Jurnal ELKOMNIKA*, vol. 8, no. 1, pp. 69–79, Jan. 2020.

[2] C. Joseph, S. Aswin, and J. Sanjeev Prasad, "Voice and gesture controlled wheelchair," in *Proc. 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC)*, August 2019, pp. 29–34.

[3] P. Arie, "Inklusi penyandang disabilitas di Indonesia," *J. Refleks. Huk.*, vol. 1, pp. 1–4, 2017.

[4] R. Chauhan, Y. Jain, H. Agarwal, and A. Patil, "Study of implementation of voice controlled wheelchair," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. Syst. (ICACCS),*, Oct. 2016, pp. 1–4.

[5] K. B. Rai, J. Thakur, and N. Rai, "Voice controlled wheel chair using arduino," *Int. J. Sci. Technol. Manag.*, vol. 4, no. 6, pp. 6–13, June 2015.

[6] D. Aradhana, "Voice controlled smart wheel chair," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 6, pp. 568–570, June 2020.

[7] M. Jabardi, "Voice controlled smart electric-powered wheelchair based on artificial neural network network," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 31–37, June 2017.

[8] H. Sharma, A. Gadhavi, P. Pandya, P. Methaniya, and P. Jadeja, "Voice controlled wheel chair," in *Proc. National Conf. on Emerging Trends, Challenges & Oppor. in Power Sector*, March 2017, pp. 9–13.

[9] J. A. Clark and R.B. Roemer, "Voice controlled wheelchair," *Archives of Physical Medicine and Rehabilitation*, vol. 58, no. 4, pp. 169-175, Apr. 1977 .

[10] M. Ilyas Malik, T. Bashir, M. Omar, and F. Khan, "Voice controlled wheel chair system," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 6, pp. 411–419, June 2017.

[11] Z. Effendi, T. Erlina, and R. Aisuwarya, "Pengenalan suara menggunakan metode MFCC (Mel Frequency Cepstrum Coefficients) dan DTW (Dynamic Time Warping) untuk sistem penguncian pintu," *Semin. Nas. Teknol. Inf. dan Komun. Terap. (SEMANTIK)*, June 2015, pp. 239–243.

[12] P. Yenigalla, A. Kumar, S. Tripathi, C. Singh, S. Kar, and J. Vepa, "Speech emotion recognition using spectrogram & phoneme embedding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Sep. 2018, pp. 3688–3692.

[13] Prabhu. (view Jan. 2021). Understanding of Convolutional Neural Network (CNN)—Deep Learning [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[14] J. Heaton, *Artificial Intelligence for Humans Volume 3 : Deep Learning and Neural Networks*, 1st ed., vol. 3, Heaton Research: Washington DC, 2015.

[15] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," *Entropy*, vol. 19, no. 6, May 2017.