

Evaluasi Model Machine Learning Klasifikasi Gerak Tangan Untuk Sistem Kontrol Prototipe Protesis Tangan

I.M.E. Darmayasa Adiputra¹, Karuna K.S Prasad², Ilham Fauzi³, I.M.P. Arya Winata⁴, I.W. Widhiada⁵

[Submission: 13-02-2023, Accepted: 28-02-2023]

Abstract— Obstacles in the form of loss of function of body parts will cause difficulty in carrying out normal activities. In the application of *electromyography* (EMG) and *electroencephalography* (EEG) sensors that are not good at compensating for various kinds of human physical conditions, *force sensing resistor* (FSR) sensors can be an alternative to EMG and EEG in hand prostheses. In planning the neural network model, the data needed for the actual output is only in the form of hand gestures for post-amputation non-patients. *Long Short Term Memory* (LSTM) is used because it can handle data processing in the long run which is one of the conditions that arise in sequential data processing. The resulting evaluation metrics are in the form of accuracy values in training data with epoch 200 and accuracy in data testing. The first result with no dropout variation shows the accuracy value in training is 0,9449 and the accuracy in testing is 0,961 with the loss value in training is 0,1284 and the loss in testing is 0,0717. The second result with dropout variations shows the accuracy value in training is 0,9699 and accuracy in testing is 0,9688 with a loss value in training is 0,0803 and loss in testing is 0,1061. the metrics accuracy evaluation generated on the dataset has exceeded the value of 0,9. This indicates that the model has run well for the classification of 11 movements.

Keyword — Control System; machine learning; moving arm; protesis arm.

Intisari— Hambatan berupa kehilangan fungsi bagian tubuh akan menyebabkan kesulitan dalam melakukan kegiatan secara normal. Dalam penerapan sensor *electromyography* (EMG) dan *electroencephalography* (EEG) yang kurang baik dalam mengimbangi berbagai macam kondisi fisik manusia, sensor *force sensing resistor* (FSR) dapat menjadi alternatif pengganti EMG dan EEG pada protesis tangan. Dalam perencanaan model neural network, data yang dibutuhkan pada actual output hanya berupa hand gesture pada orang non pasien pasca amputasi. *Long Short Term Memory* (LSTM) digunakan karena dapat menangani proses data dalam jangka panjang yang merupakan salah satu keadaan timbul dalam pengolahan data sekuensial. evaluasi metrics yang dihasilkan berupa nilai accuracy pada data training dengan epoch 200 dan accuracy pada data testing. Hasil pertama dengan tanpa variasi dropout menunjukkan nilai accuracy pada training 0,9449 dan accuracy pada testing 0,961 dengan nilai loss pada training 0,1284 dan loss pada testing 0,0717. Hasil kedua dengan variasi dropout menunjukkan nilai accuracy pada training 0,9699 dan accuracy

pada testing 0,9688 dengan nilai loss pada training 0,0803 dan loss pada testing 0,1061. evaluasi metrics accuracy yang dihasilkan pada dataset telah melampaui nilai 0,9. Hal ini mengindikasikan model telah berjalan dengan baik untuk klasifikasi pada 11 gerakan.

Kata Kunci— Sistem kontrol; machine learning; gerakan tangan; protesis tangan.

I. PENDAHULUAN

Terdapat 15 dari 100 orang di dunia mengalami disabilitas, yakni kondisi individu yang tidak mampu melaksanakan suatu aktivitas secara normal karena kehilangan fungsi tubuh, baik secara psikologi, fisiologi, maupun anatomi [1]. Hambatan berupa kehilangan fungsi bagian tubuh akan menyebabkan kesulitan dalam melakukan kegiatan secara normal. Kegiatan sehari-hari yang membutuhkan banyak gerakan, tentu akan menyulitkan penyandang disabilitas dengan cacat pada bagian tubuh yang memiliki peranan besar dalam mobilitas, seperti anggota gerak tubuh. Hal ini tentu akan berdampak pada terhambatnya kegiatan sosial yang hendak dilakukan oleh penyandang [2]. Terlebih lagi anggota gerak tubuh, khususnya lengan dan tangan, mengambil andil penting dalam sebagian besar aktivitas manusia [3]. Tidak sedikit penyandang disabilitas ini merupakan pasien yang telah melalui proses amputasi lengan dan tangan akibat cedera dari kecelakaan besar. Studi yang dilaksanakan oleh [4] bahkan menunjukkan bahwa salah satu kawasan yang memiliki angka amputasi tinggi berada di Asia Tenggara.

Pada awal pengembangannya, protesis mekanik dikonstruksi untuk mengembalikan fungsi bagian tubuh salah satunya tangan pasca amputasi sebagai pengait serta mengambil barang secara statis. Namun, protesis semacam ini membutuhkan kontrol manual yang lebih banyak dan membuat pengguna merasa relatif kurang nyaman. Kondisi ini menimbulkan tingginya penolakan protesis mekanik di masyarakat [5]. Oleh karena itu, pengembangan aspek fungsional kontrol secara otomatis untuk protesis lengan turut dilakukan. Salah satu pengembangan kontrol otomatis adalah dengan pemanfaatan sensor dengan mengumpulkan data dan informasi yang diperlukan untuk sistem kontrol secara realtime. Mendeteksi dan menginterpretasikan data sensor yang terpasang pada tangan untuk diubah menjadi gerakan yang akurat dan fungsional merupakan tantangan yang kompleks. Proses tersebut melibatkan analisis data yang rumit dan kompleks yang harus dikonversi ke dalam gerakan tangan yang sesuai dengan tujuan yang ingin dicapai. Bahkan dengan teknologi sensor tercanggih, masih sulit untuk mengambil data yang cukup akurat untuk meniru gerakan tangan manusia

^{1,2,3,4,5} Program Studi Teknik Mesin Fakultas Teknik Universitas Udayana, Jln. Raya Kampus Unud Bukit Jimbaran 08361 INDONESIA (telp: 0361-703321; e-mail: esa.darmayasa@gmail.com, iamkaruna09@gmail.com, ilhamg2216@gmail.com, aryaw2001@gmail.com, wynwidhiada@unud.ac.id)



secara tepat. Oleh karena itu, untuk mencapai tujuan yang diinginkan, perlu dilakukan pendekatan yang semirip mungkin dengan gerakan tangan manusia yang sebenarnya.

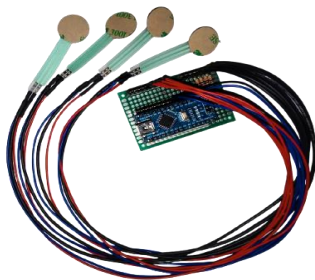
Sebagian besar penelitian memiliki perhatian lebih pada pemanfaatan sensor electromyography (EMG) dan electroencephalography (EEG) [6]. Namun, mekanisme dari kedua sensor ini secara praktikal masih fluktuatif dan tidak dapat merepresentasikan input dengan baik akibat kondisi fisik manusia yang berbeda-beda. Oleh karena itu, penggunaan sensor yang tepat guna serta tersedianya algoritma kerja tertentu sangat diperlukan. Pendekatan-pendekatan ini ditujukan agar gerakan yang dihasilkan oleh prosthesis lengan mampu merepresentasikan kondisi aslinya. Merujuk pada permasalahan-permasalahan tersebut, peningkatan fungsionalitas terhadap aspek sistem kontrol perlu dilaksanakan.

Dalam penerapan sensor EMG dan EEG yang kurang baik dalam mengimbangi berbagai macam kondisi fisik manusia, sensor *force sensing resistor (FSR)* dapat menjadi alternatif pengganti EMG dan EEG pada prosthesis tangan [7]. Berdasarkan prinsip kerjanya, sensor EMG dan EEG berbeda dengan *sensor FSR* yang mengukur perubahan hambatan listrik pada permukaan yang disebabkan oleh tekanan atau gaya. *Sensor FSR* juga sangat fleksibel dan dapat disesuaikan dengan berbagai bentuk dan ukuran, sehingga cocok diterapkan pada prosthesis tangan. Untuk mendukung potensi *sensor FSR* diperlukan juga model *machine learning* dalam pengolahan data dari *sensor FSR* untuk menghasilkan output gerakan tangan yang sesuai. Pada penelitian ini, memanfaatkan *sensor FSR* dan model *machine learning* untuk mencapai fungsionalitas dalam aspek sistem kontrol pada prosthesis tangan.

II. STUDI PUSTAKA

A. Rangkaian Sensor

Diperlukan perangkat keras yang digunakan untuk menangkap sinyal tekanan atau gaya yang kemudian diubah menjadi sebuah data yang dapat diolah model machine learning. Komponen yang digunakan pada kali ini untuk menangkap sinyal berupa tekanan adalah dengan pemanfaatan sensor FSR402 sebanyak 4 buah yang terhubung ke dalam sebuah mikrokontroler Arduino Nano yang menampilkan sinyal pada monitor yang kemudian menjadi data yang dapat diproses oleh machine learning [8]. Perangkat ini sudah disederhanakan dengan tujuan memfokuskan hanya pada fungsi sensor untuk menangkap sinyal tekanan. Semua komponen ini terhubung dalam sebuah circuit approx. berukuran 5x7cm yang ditunjukkan pada Gambar 1.



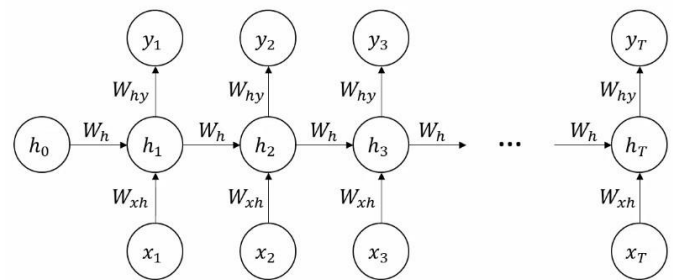
Gambar 1: Rangkaian 4 sensor FSR402

B. Machine Learning Neural Network

Pendekatan dengan algoritma machine learning merupakan salah satu pengembangan tingkat lanjut (advanced) dari sistem kontrol pada prosthesis lengan. Pada konsepnya, machine learning merupakan algoritma yang mampu menghasilkan rule baru berupa model matematis, dibandingkan dengan algoritma konvensional yang mana rule-nya sudah ditetapkan di awal. Pendekatan ini sangat baik diterapkan pada konteks permasalahan yang sulit ditetapkan model matematisnya. Lebih jauh lagi, machine learning telah dikembangkan melalui replikasi cara kerja neuron pada tubuh dengan istilah *neural network* [9], yang mana mampu menghasilkan kompleksitas model matematis yang tinggi namun saat implementasi tidak memerlukan komputasi yang banyak.

Dengan berbagai masalah yang dapat dicakup, pendekatan arsitektur *neural network* turut mengalami perkembangan pada permasalahan yang spesifik. Klasifikasi dan rekognisi pola (classification and pattern recognition) pada serangkaian data (data series) merupakan salah satu pendekatan yang dimuat oleh *neural network* [10], [11]. Penerapan recurrent *neural network* (RNN) dapat dijadikan pendekatan pada permasalahan data series, menimbang kemampuan arsitektur tidak hanya melihat setiap data point sebagai bagian yang terpisah, namun juga meninjau hubungan antar data point-nya [12]. Merujuk dari hal tersebut, kompleksitas respon tubuh dalam memberikan gerakan dapat mengimplementasikan konsep ini. Dengan demikian, fungsionalitas dapat tercapai juga pada aspek sistem kontrol.

Dalam perencanaan model *neural network*, data yang dibutuhkan pada actual output hanya berupa hand gesture pada orang non pasien pasca amputasi. Dengan demikian, pengolahan gambar (computer vision) tangan perlu dilakukan terlebih dahulu, salah satunya dengan menggunakan modul MediaPipe [13]. Arsitektur *neural network* yang digunakan pada proses ini adalah berbasis *Recurrent Neural Network* yang baik dalam mengolah data pattern recognition seperti yang dihasilkan oleh sensor taktil [14]. Arsitektur *Recurrent Neural Network* ditunjukkan pada Gambar 2 berikut.



Gambar 2: Arsitektur *Recurrent Neural Network*

Pertama dari data pada time step pertama. Lalu hitung *hidden state* pada time step pertama dengan menggunakan input, jadi input dikali dengan *weight* antara input dan hidden state W_{xh} . Lalu ditambah dengan hidden state time step sebelumnya, yaitu time step 0, yang di set dengan vektor 0, yang dikali dengan *weight* antar hidden state W_h . Lalu, ditambah dengan bias b_h dan akhirnya diberi activation function f , biasanya dengan hypertangent. Setelah menghitung hidden state, output y pada time step pertama dihitung dari

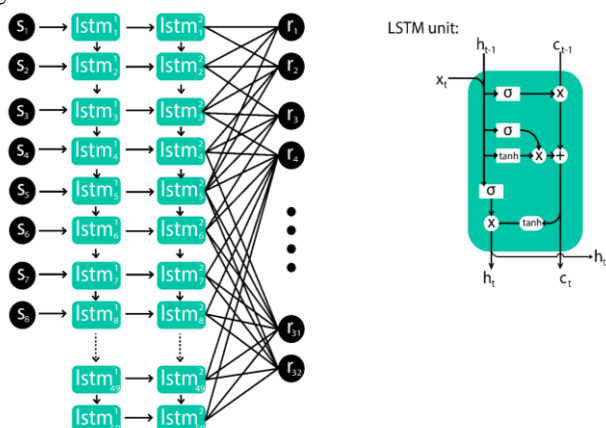
hidden state time step pertama dengan weight antara hidden state dan output W_{hy} . Ditambah bias b_y dan activation function g yang jenisnya tergantung kebutuhan. Untuk hidden state time step sekarang, pakai input time step sekarang dan hidden state sebelumnya. Dan pakai weight W_{xh} dan W_h yang sama dengan sebelumnya, makanya disebut recurrent karena weightnya berulang. Untuk outputnya pakai hidden state sekarang dan pakai weight W_{hy} yang sama juga dan seterusnya sampai pada time step terakhir. Persamaan *hidden state* dan output y dimuat dalam persamaan (1) dan (2) secara berturut-turut.

$$h_T = f(W_{xh} x_T + W_h h_{T-1} + b_h) \dots\dots\dots (1)$$

$$y_T = g(W_{hy} h_T + b_y) \dots\dots\dots (2)$$

Pada RNN, setiap output dipengaruhi oleh time step sebelum-sebelumnya seperti output y_3 di sini mendapatkan informasi dari input sekarang dan juga sebelumnya yaitu x_1, x_2 , dan x_3 . Jadi seolah-olah ada memori lampau yang didapatkan dari *hidden state*. Pada intinya ada semacam memori yang dihintarkan oleh *hidden state* untuk time step berikutnya.

Pada penelitian kali ini menggunakan jenis gate *Long Short Term Memory (LSTM)* yang merupakan jenis dari RNN. LSTM digunakan karena dapat menangani proses data dalam jangka panjang yang merupakan salah satu keadaan timbul dalam pengolahan data sekuensial.



Gambar 3: Arsitektur gate *Long Short Term Memory (LSTM)*

Sama seperti RNN, time step lainnya juga punya diagram yang sama. LSTM memiliki ciri khas selain terdapat *hidden state* (h_t) yang dilanjutkan ke time step berikutnya, terdapat juga yang disebut *cell state* (C_t) yang membawa informasi sebelumnya yang akan diproses pada *hidden state* [15]. *Cell state* (C_t) ini seperti state internal pada LSTM. Pada LSTM terdapat *forget gate* (f_t). Forget gate ini dihitung dari gabungan input sekarang (x_t) dan *hidden state* (h_t) sebelumnya dengan *weight* (W_f) dan *bias* (b_f) yang kemudian masuk ke fungsi *sigmoid* (σ). Fungsi *sigmoid* (σ) ini punya output dengan range 0 hingga 1 yang nantinya *forget gate* (f_t) ini dikalikan dengan *cell state* time step sebelumnya. Jadi, kalau *forget gate*-nya 0, maka *cell state* time step sebelumnya dilupakan. Kalau *forget gate*-nya 1, *cell state* time step sebelumnya lanjut ke time step

sekarang. *Forget gate* (f_t) dirumuskan dalam persamaan (3) berikut.

$$f_t = \sigma(W_f x_1 [h_{t-1}, x_t] + b_f) \dots\dots\dots (3)$$

Setelah *forget gate*, terdapat input gate. Mirip dengan *forget gate*, *input gate* (i_t) terdiri dari input sekarang dan *hidden state* sebelumnya, dengan *weight* dan *bias* yang kemudian masuk ke *sigmoid*. Lalu, inputnya dihitung dengan dengan *weight* (W_i) dan *bias* (b_i). Lalu *input gate* (i_t) ini dikalikan per elemen dengan inputnya kemudian ditambahkan ke cell. Jadi, input gate melambangkan seberapa banyak input yang akan ditambahkan ke *cell state* baru (\tilde{C}_t) yang menggunakan aktivasi tanh untuk menghasilkan vektor nilai. *Input gate* (i_t) dan *cell state* baru (\tilde{C}_t) dirumuskan dalam persamaan (4) dan (5) secara berturut-turut.

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \dots\dots\dots (4)$$

$$\tilde{C}_t = \tanh(W_c [h_{t-1}, x_t] + b_c) \dots\dots\dots (5)$$

Untuk pembaruan cell state dilakukan dengan cara menjumlahkan *cell state* (C_t) dengan *cell state* baru (\tilde{C}_t). Dimana *cell state* (C_t) dikalikan dengan *forget gate* (f_t) dan *cell state* baru (\tilde{C}_t) dikalikan dengan *input gate* (i_t). Berikut *cell state* (C_t) dirumuskan dalam persamaan (6).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots\dots\dots (6)$$

Setelah mendapatkan *cell state* baru (\tilde{C}_t) yang kemudian diproses untuk menjadi *hidden state* pada time step sekarang. *Output gate* (o_t), mirip seperti gate-gate lainnya, terdiri dari gabungan *hidden state* sebelumnya dan input sekarang, dengan *weight* (W_o) dan *bias* (b_o), yang kemudian dimasukkan ke fungsi *sigmoid* (σ). Lalu outputnya sendiri adalah tanh dari *cell state* (C_t) yang sudah dihitung. *Output gate* (o_t) akan mengatur seberapa banyak output yang akan jadi *hidden state* dengan perkalian antar elemen. *Output gate* (o_t) dan *hidden state* (h_t) dirumuskan dalam persamaan (7) dan (8) secara berturut-turut.

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \dots\dots\dots (7)$$

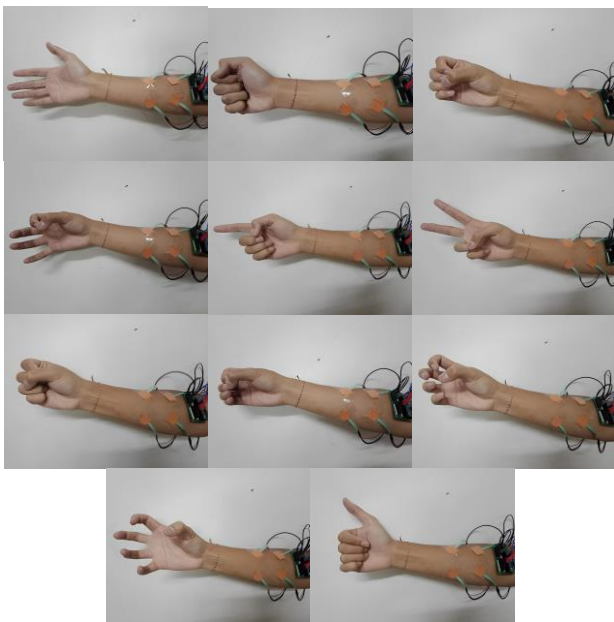
$$h_t = o_t * \tanh(C_t) \dots\dots\dots (8)$$

III. METODOLOGI

A. Pengambilan dataset

Pengumpulan dataset dilakukan dengan mendeteksi secara langsung menggunakan sensor FSR402 sebanyak 4 buah yang terhubung dengan Arduino Nano dan terpasang pada tangan subjek. Sinyal yang direkam kemudian diexport dalam format file .csv. Dataset sensor FSR402 yang didapat sejumlah 4488 datapoints yang selanjutnya dilakukan *splitting* dengan rasio 80:20 berturut-turut untuk *training* dan *testing* dengan tujuan memperoleh data persiapan evaluasi. Beberapa gerakan yang digunakan sebagai label pada model ini berjumlah 11, yaitu ditunjukkan pada Gambar 4.





Gambar 4: Label Klasifikasi Model Neural Network Protesis Lengan

B. Training dan testing dataset

Setelah melakukan *preprocessing data* melalui *splitting*, *training* model dilakukan dengan algoritma RNN-LSTM pada penggunaan data *time-series* sebagaimana dataset pergerakan tangan yang telah dikumpulkan. Adapun parameter yang digunakan dalam proses *training* dan *testing* yaitu *optimizer adam* dan perhitungan *loss function crossentropy*.

Optimizer Adam, juga dikenal sebagai Adaptive Moment Estimation, merupakan salah satu algoritma pembelajaran yang digunakan dalam proses pemilihan parameter model [16]. Adam memadukan teknik gradient descent dan gradient descent with momentum. Algoritma ini memperhitungkan rata-rata moving dari gradien dan variansi gradien untuk setiap parameter, sehingga dapat menyesuaikan tingkat pembaruan untuk setiap parameter berdasarkan karakteristik dari gradien. Hal ini menjadikan Adam lebih efektif dalam mengatasi masalah gradien yang tidak seimbang atau memiliki skala yang berbeda-beda [17]. Adam optimizer efisien secara komputasi, kebutuhan memori rendah, dan mudah diimplementasikan. Penelitian ini menggunakan Adam optimizer untuk mengubah atribut *neural network* untuk meminimalkan loss. Sedangkan *loss function crossentropy* adalah sebuah perhitungan yang sering digunakan dalam pembelajaran mesin dan pembelajaran mesin mendalam untuk menilai kinerja dari sebuah model klasifikasi [18]. *Loss function crossentropy* mengukur seberapa jauh hasil prediksi dari model dari target yang sebenarnya. *Loss function crossentropy* sering digunakan sebagai fungsi loss dalam proses pelatihan model, di mana algoritma optimisasi akan berusaha untuk mengurangi nilai *crossentropy* selama pelatihan. Dengan mengurangi nilai, model akan belajar untuk membuat prediksi yang lebih baik dan akurat.

Penambahan variasi dropout turut dilakukan. cara kerja dropout yaitu dengan memilih acak sebuah neuron kemudian neuron tersebut tidak akan digunakan selama training [19]. Hal tersebut dapat mempercepat proses training itu sendiri karena neuron yang terlewat juga lebih sedikit. Dari cara kerja dapat disimpulkan bahwa dropout dapat digunakan untuk mengurangi overfitting. Dropout dapat digunakan pada

extraction layer maupun di fully connected layer. Sehingga fungsi penambahan dropout dilakukan agar model tidak menghafal dataset sebagaimana keadaan saat deployment sensor FSR402 memberikan sinyal dataset yang berbeda menyesuaikan subjek nantinya.

C. Evaluasi Model Neural Network

Untuk bahan evaluasi model, salah satu metrics simple performance yang dikenal sebagai *metrics accuracy* dapat dihitung dengan cara membagi jumlah rasio prediksi yang benar dengan jumlah rasio prediksi yang salah [20]. Secara konsep, *accuracy* dihasilkan melalui *cross-validation actual* dan *predicted value* yang ditunjukkan pada Tabel 1:

TABEL I
CROSS-VALIDATION NILAI SEBENARNYA (ACTUAL VALUE) DAN NILAI PREDIKSI (PREDICTED VALUE)

		Actual Value	
		True	False
Predicted Value	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Berikut *accuracy* dirumuskan dalam persamaan (9).

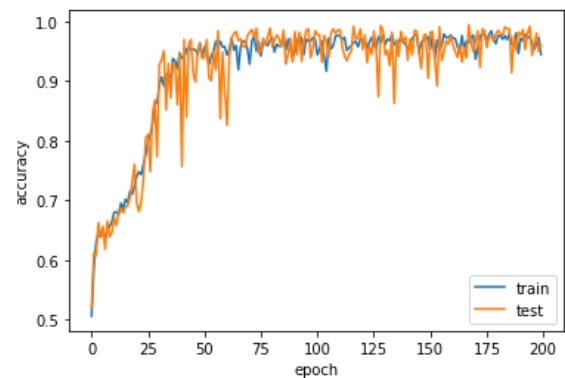
$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \dots\dots\dots(9)$$

IV. HASIL DAN PEMBAHASAN

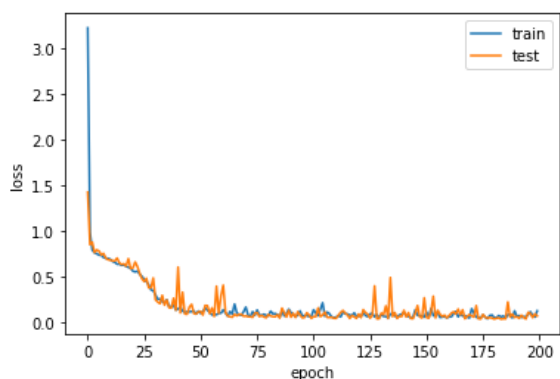
Perancangan model machine learning, training, dan testing dataset dilakukan menggunakan Google Colab yang merupakan editor yang menggunakan bahasa python.

A. Hasil

Hasil *evaluasi metrics* yang dihasilkan berupa nilai *accuracy* pada data *training* dengan epoch 200 dan *accuracy* pada data *testing*. Terdapat dua variasi pada layer model yaitu, tanpa *dropout* dan dengan *dropout* layer. Hasil pertama dengan tanpa variasi *dropout* menunjukkan nilai *accuracy* pada *training* 0,9449 dan *accuracy* pada *testing* 0,961 dengan nilai *loss* pada *training* 0,1284 dan *loss* pada *testing* 0,0717. Gambar 5: merupakan visualisasi Grafik nilai *accuracy* dan *loss* model pada tiap epoch training tanpa variasi *dropout*.



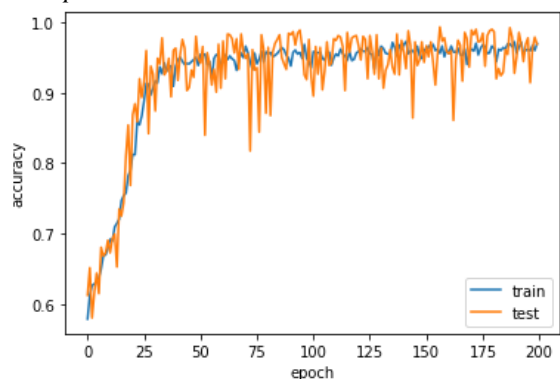
(a)



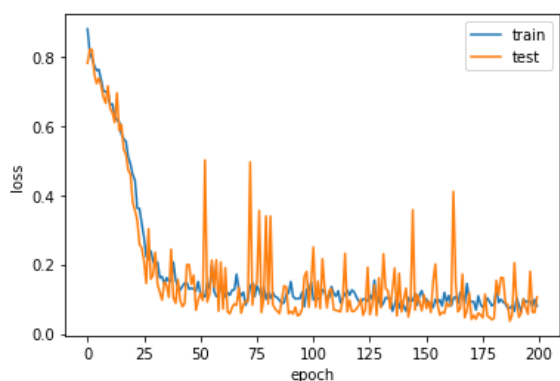
(b)

Gambar 5: Grafik nilai (a) *accuracy* dan (b) *loss* model pada tiap epoch training tanpa variasi *dropout*

Hasil kedua dengan variasi *dropout* menunjukkan nilai *accuracy* pada *training* 0,9699 dan *accuracy* pada *testing* 0,9688 dengan nilai *loss* pada *training* 0,0803 dan *loss* pada *testing* 0,1061. Gambar 6: merupakan visualisasi Grafik nilai *accuracy* dan *loss* model pada tiap epoch training dengan variasi *dropout*.



(a)



(b)

Gambar 6: Grafik nilai (a) *accuracy* dan (b) *loss* model pada tiap epoch training dengan variasi *dropout*

Dari grafik *training* dan *testing* yang didapat, menunjukkan bahwa penambahan variasi *dropout* menunjukkan nilai *accuracy* lebih tinggi daripada tanpa variasi *dropout* dengan selisih nilai

loss yang kecil. Mengingat tujuan penambahan variasi *dropout* dilakukan agar model tidak menghafal dataset sebagaimana keadaan saat deployment sensor FSR402 memberikan sinyal dataset yang berbeda menyesuaikan subjek nantinya. Hal ini mengindikasikan model telah berjalan dengan baik untuk klasifikasi pada 11 gerakan.

V. KESIMPULAN

Dari hasil penelitian yang didapat, menunjukkan bahwa pengembangan model *machine learning neural network* yang telah dilaksanakan ini sangat memungkinkan untuk diimplementasikan. Menghasilkan sistem kontrol pada prostesis tangan dengan memanfaatkan algoritma RNN-LSTM yang mengolah dataset berupa sinyal dari sensor FSR402, kemudian ditunjukkan dari nilai *evaluasi metrics accuracy* yang dihasilkan pada *dataset* telah melampaui nilai 0,9. Hal ini mengindikasikan model telah berjalan dengan baik untuk klasifikasi pada 11 gerakan. Sebagai tahapan pengembangan lanjutan, pengumpulan *dataset* klasifikasi gerakan tangan yang lebih bervariasi perlu dilakukan agar mampu merepresentasikan keadaan sebagaimana saat dideploy langsung pada sistem kontrol prostesis tangan.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya diberikan kepada Universitas Udayana melalui Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) atas pemberian hibah pada kegiatan Merdeka Belajar Kampus Merdeka (MBKM) Riset dengan Nomor Keputusan 921/UN14/HK/2022, serta Mitra MBKM Riset Tim Prostesis Tangan Puspadi Bali atas bantuan dan arahan terhadap pengembangan prostesis tangan dari perspektif praktisi. Sehingga, penelitian ini dapat berjalan dengan semestinya.

REFERENSI

- [1] World Health Organization. (2022). Disability and health. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/disability-and-health>.
- [2] Shin, A. Y., Spinner, R. J., Steinmann, S. P., & Bishop, A. T., "Adult traumatic brachial plexus injuries," *The Journal of the American Academy of Orthopaedic Surgeons*, 13(6), 382–396, 2005. doi: 10.5435/00124635-200510000-00003.
- [3] Vujaklija, I., Farina, D., & Aszmann, O. C., "New developments in prosthetic arm systems," *Orthopedic Research and Reviews*, 8, 31–39, 2016. doi: 10.2147/ORR.S71468.
- [4] Satam, I. A., "Review Studying of the Latest Development of Prosthetic Limbs Technologies," 2021.
- [5] Hutchinson, D. T., "The Quest of Bionic Arms," *Journal of the American Academy of Orthopaedic Surgeons* 22(6):p 346-351, June 2014. doi: 10.5435/JAAOS-22-06-346.
- [6] Naim, A. M., Wickramasinghe, K., De Silva, A., Perera, M. V., Lalitharatne, T. D., & Kappel, S. L., "Low-cost Active Dry-Contact Surface EMG Sensor for Bionic Arms," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 3327–3332, 2020. doi: 10.1109/SMC42975.2020.9283285
- [7] Li, N., Yang, D., Jiang, L. *et al.*, "Combined Use of FSR Sensor Array and SVM Classifier for Finger Motion Recognition Based on Pressure Distribution Map", *J Bionic Eng* 9, 39–47, 2012. doi: 10.1016/S1672-6529(11)60095-4



- [8] Pahlevi, A., Syauby, D., & Rahayudi, B., "Implementasi Gripper Pada End Effector Robot Untuk Memegang Telur Ayam Dengan Sensor FSR (Force Sensitive Resistor)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 2, No. 12, p. 6761-6770, 2018. ISSN 2548-964X.
- [9] C. M. Bishop, "Neural networks and their applications," *Review of Scientific Instruments* 65, 1803, 1994. doi: 10.1063/1.1144830
- [10] Mishra, D., Naik, B., Sahoo, R.M., Nayak, J., "Deep Recurrent Neural Network (Deep-RNN) for Classification of Nonlinear Data," *Advances in Intelligent Systems and Computing*, vol 1120, 2020. doi: 10.1007/978-981-15-2449-3_17
- [11] Lely Meilina, I Nyoman Satya Kumara, I Nyoman Setiawan, "Literature Review Klasifikasi Data Menggunakan Metode Cosine Similarity dan Artificial Neural Network," *Majalah Ilmiah Teknologi Elektro*, v. 20, n. 2, p. 307-314, dec. 2021. ISSN 2503-2372.
- [12] I. F. Rozi, V. N. Wijayaningrum och N. Khozin, "Klasifikasi Teks Laporan Masyarakat Pada Situs Laporan! Menggunakan Recurrent Neural Network," *Jurnal Sistem Informasi*, Vol. 1, No.3, 2020.
- [13] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M., "MediaPipe: A Framework for Building Perception Pipelines," 2019. [Online]. Available: <http://arxiv.org/abs/1906.08172>
- [14] Barron, O., Raison, M., Gaudet, G., & Achiche, S., "Recurrent Neural Network for electromyographic gesture recognition in transhumeral amputees," *Applied Soft Computing Journal*, 96, 106616, 2020. doi: 10.1016/j.asoc.2020.106616
- [15] S. Hochreiter, J. Schmidhuber, "LSTM can solve hard long time lag problems," *Advances in Neural Information Processing Systems* 9, 1996.
- [16] Diederik Kingma and Jimmy Ba., "Adam: A method for stochastic optimization", 2014. doi: 10.48550/arXiv.1412.6980
- [17] Ida Bagus Leo Mahadya Suta, Made Sudarma, I Nyoman Satya Kumara, "Brain Tumor Segmentation Based on Magnetic Resonance Imaging Images Using the U-NET Method," *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 2, p. 151-156, dec. 2020. ISSN 2503-2372.
- [18] de Boer, PT., Kroese, D.P., Mannor, S., "A Tutorial on the Cross-Entropy Method," *Ann Oper Res* 134, 19-67, 2005. doi: 10.1007/s10479-005-5724-z
- [19] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R., "Improving neural networks by preventing co-adaptation of feature detectors." 2012. arXiv preprint arXiv:1207.0580.
- [20] A. Gunawardana, G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research* 10, 2935-2962, 2009.