

Desain Algoritma Autonomous Deep Learning (ADL) untuk Sistem Kontrol Tangan Prostetis pada Disabilitas

Widhi Winata Sakti ^{a,1,*}, Siti Tsaniyatul Miratis Sulthoniyah ^{b,2}, Donny Setiawan ^{b,3}, Adi Mulyadi ^{b,4}, Khairul Anam ^{b,5}

^{a,1} Universitas PGRI Banyuwangi, Jalan Ikan Tongkol No 22, Kertosari, Banyuwangi, Indonesia

^{b,2} Universitas PGRI Banyuwangi, Jalan Ikan Tongkol No 22, Kertosari, Banyuwangi, Indonesia

^{b,3} Universitas PGRI Banyuwangi, Jalan Ikan Tongkol No 22, Kertosari, Banyuwangi, Indonesia

^{b,4} Universitas PGRI Banyuwangi, Jalan Ikan Tongkol No 22, Kertosari, Banyuwangi, Indonesia

^{b,5} Universitas Jember, Jalan Kalimantan Tegalboto No 37, Sumbersari, Jember, Indonesia

¹ widhiwinatas@unibabwi.ac.id *; ² miratissulthoniyah@gmail.com; ³ donnysetiawanewa@gmail.com; ⁴

adimulyadi@unibabwi.ac.id; ⁵ khairul@unej.ac.id

* Penulis Koresponden

INFO ARTIKEL

Histori Artikel

Pengajuan 2024-09-13

Diperbaiki 2025-12-30

Diterima 2025-01-18

Kata Kunci

Disabilitas

EEG

Klasifikasi

Kontrol

ADL

ABSTRAK

Penyandang Disabilitas (ODD) memiliki risiko kemiskinan yang tinggi di Indonesia dan dianggap tidak produktif. Menurut data Badan Pusat Statistik (BPS), pada tahun 2019-2020, jumlah penyandang disabilitas di Indonesia mencapai lebih dari 28 juta orang. Tangan palsu telah menjadi solusi untuk membantu individu penyandang disabilitas meningkatkan kualitas hidupnya. Pemrosesan sinyal EEG untuk kontrol prostetik masih tergolong baru, dan diperlukan penelitian lebih lanjut untuk mengoptimalkan kinerja algoritma. diusulkan menggunakan desain algoritma Autonomous Deep Learning (ADL). Struktur jaringan dapat dibangun dari awal tanpa adanya pengaturan manual, mengingat kompleksitas jaringan saraf yang sering mengalami overfitting, model tidak efektif dalam klasifikasi. Hasil percobaan dari 6 subjek dengan rata-rata 96% dengan error 10% pada subjek mandiri..

1. Pendahuluan

Di Indonesia, masalah disabilitas merupakan isu penting yang memerlukan perhatian serius dari pemerintah, lembaga swadaya masyarakat, dan sektor swasta [1]. Menurut data Badan Pusat Statistik (BPS) Indonesia, pada tahun 2019-2020, jumlah penyandang disabilitas di Indonesia mencapai lebih dari 28 juta orang atau sekitar 10% dari total populasi [2]. Orang dengan disabilitas (ODD) memiliki risiko kemiskinan yang tinggi di Indonesia dan dianggap tidak produktif [3]. Salah satu tantangan yang dihadapi oleh individu dengan disabilitas adalah keterbatasan dalam menggunakan anggota tubuhnya, seperti tangan[4].

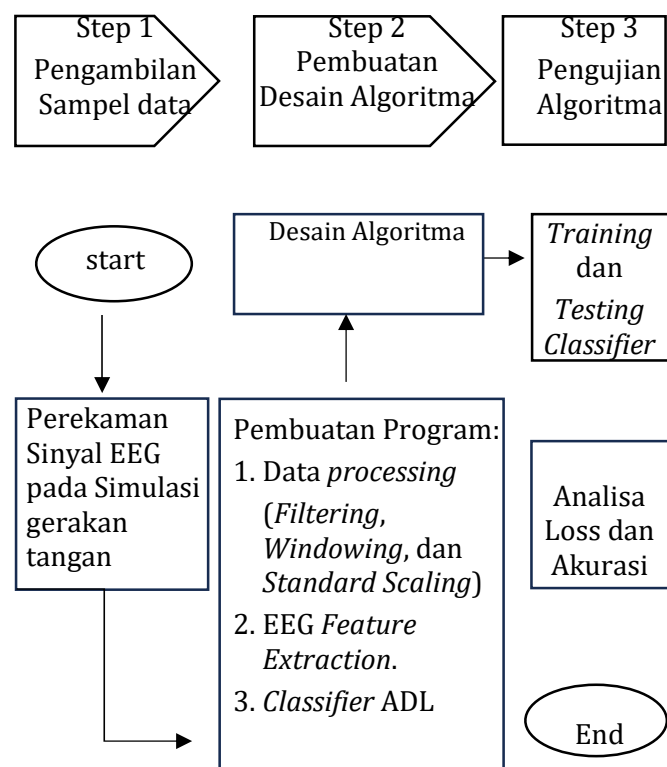
Tangan prostetis telah menjadi solusi untuk membantu individu dengan disabilitas untuk meningkatkan kualitas hidup mereka [5][6]. Penggunaan *Electroencephalogram* (EEG) menggunakan *Brain Computer Interfaces* (BCI) menunjukkan potensi besar dalam mengatasi masalah ini[7][8]. BCI memungkinkan kontrol tangan prostetis menggunakan pikiran dengan memanfaatkan sinyal-sinyal otak yang dihasilkan oleh individu [9].

Pada penelitian sebelumnya telah diusulkan berbagai metode *deep Learning* untuk klasifikasi kontrol tangan prostetis antara lain, menggunakan *convolutional neural networks* (CNN) untuk mengklasifikasikan sinyal dari tiga kelas gerakan tangan dan mendapat keberhasilan akurasi mencapai 70% [10][11]. Penambahan fitur pada model CNN

menggunakan *Filter Bank Common Spatial Pattern* (FBCSP) dapat meningkatkan tingkat akurasi sebesar 74,5% [12]. Konfigurasi *layer* dalam CNN memerlukan pengaturan beberapa parameter, seperti jumlah unit, jenis fungsi aktivasi, dan jumlah neuron di *output layer* [13], konfigurasi ini mengalami kelemahan ketika data mengalami banyak varian [14], hal ini menimbulkan gap penelitian pada algoritma ini. Konfigurasi dari penentuan *deep Learning* yang optimal untuk memproses sinyal EEG masih menjadi subjek penelitian. Urgensi pada penelitian ini adalah pengolahan sinyal EEG untuk kontrol prostetis masih relatif baru, dan masih diperlukan penelitian lebih lanjut untuk mengoptimalkan performa algoritma. Maka diajukan pembaruan menggunakan desain algoritma *Autonomous Deep Learning* (ADL). Desain ini memiliki mekanisme adaptif pada kontruksi jaringan yang dibangun secara otomatis dan diharapkan dapat meningkatkan akurasi pada klasifikasi kontrol tangan prostetis [15][16].

Penelitian ini merancang alat bantu bagi penderita disabilitas yang kehilangan organ tangan. Novelty dari penelitian ini adalah desain algoritma ADL yang memiliki struktur yang fleksibel dan adaptif pada proses penyusunan struktur jaringannya dan tanpa konfigurasi manual seperti penyusunan *layer*, dengan memperhatikan kompleksitas jaringan pada *neural* yang sering mengalami *overfitting* atau model tidak efektif dalam klasifikasi [17].

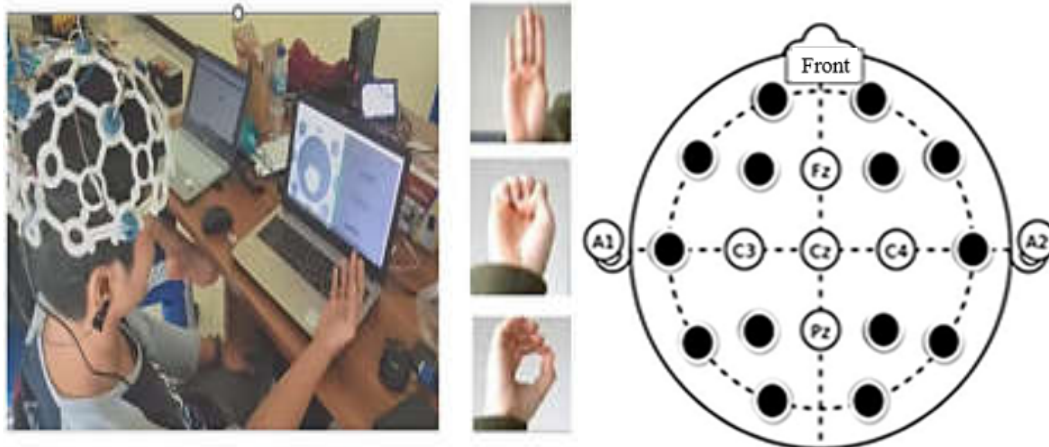
2. Metode penelitian



Gambar 1. Sistem Klasifikasi EEG

Penelitian ini menekankan pada langkah perbaikan dan pembaruan metode. Pada penelitian sebelumnya *Autonomous Deep Learning* (ADL) hanya diujikan pada *Benchmarking* sistem, pada data berbentuk fitur gambar dan data *tabular*, serta belum diujikan kepada data sinyal EEG seperti gambar 1.

1. Tahap Pengambilan Sampel data



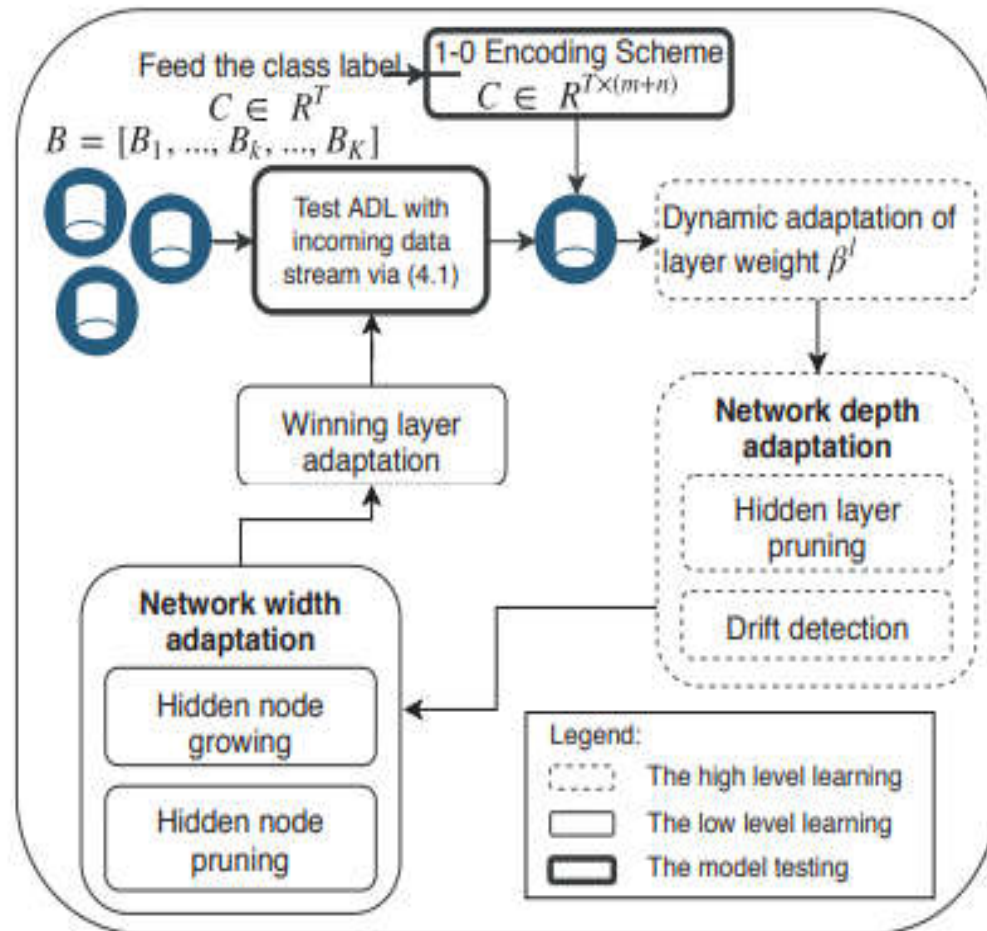
Gambar 2. Peletakan Katode EEG

Pada Gambar 2 Channel yang digunakan pada metode ini sebanyak 5 titik C3, CZ, C4, FZ, dan PZ digunakan sesuai literatur *Motor Imagery Datasheet* untuk perekaman membayangkan aksi motorik [18]. Perekaman sinyal otak dilakukan saat pergerakan tangan membuka, menggenggam dan menjimpit dengan waktu 3 detik. Perekaman akan menghasilkan data Diskrit sinyal EEG terhadap gerakan tangan [19].

2. Tahap Pembuatan Desain Algoritma

Pengolahan sinyal EEG ditekankan pada metode pengolahan fitur data, pada bagian ini fitur divisualisasikan menggunakan *Brain Computer Interface* (BCI) untuk mengetahui pergerakan data pada tampilan sinyal. Teknik pengolahan fitur sinyal mencakup urutan pemrosesan sebagai berikut:

- A. *Notch filter* pada proses ini bertujuan untuk denoising pada sinyal direntang 50 hingga 60 Hz dengan konsep pemilihan kualitas sinyal pada pergerakan rata-rata dari frekuensi sampling digital. Hal ini mengacu pada pemrosesan sinyal yang bertujuan mencegah timbulnya *noise* [20]. *Filter Bandpass*, metode ini mengacu pada literatur yang menyatakan bahwa sinyal EEG untuk kontrol berada pada rentang *Alpha* 7-13 Hz, *Betha* 14-30 Hz, dan *Gamma* 30-100 Hz. Hal ini bertujuan untuk mendapatkan respon yang lebih informatif pada stimulus yang dilakukan selama kegiatan motorik [21].
- B. *Windowing* atau segmentasi data untuk mendapatkan Quantity sinyal informatif pada setiap step pergerakan segmentasi data sinyal dengan mengacu pada literatur untuk mengeneralisasi jumlah varians pada fitur sinyal [22].
- C. *Standart Scaling* pada metode ini data akan diskalakan dari fitur yang mempunyai simpangan yang terlalu tinggi menjadi simpangan sinyal yang lebih rendah, bertujuan untuk menurunkan simpangan error pada model [23].
- D. *Fitur Ekstraksi*, tahap ini akan mengekstrak sinyal EEG 5 fitur utama yaitu *Principal Component Analisis* (PCA) [24].
- E. *Clasifier ADL*



Gambar 3. Clasifier ADL [15]

Pada gambar 3 *Classifier* sistem menggunakan *Autonomous Deep Learning* (ADL). mekanisme *Autonomous Deep Learning* digunakan untuk klasifikasi adaptif dimana ADL membuat jaringan secara mandiri berdasarkan bentuk matrik fitur [25]. ADL mencirikan struktur kedalaman yang berbeda yang dirumuskan sebagai berikut :

$$h^{(l)} = \delta(W^{(l)}h^{(l-1)} + b^{(l)}), h^{(0)} = X$$

Lapisan tersembunyi $h^{(l)}$ memiliki koneksi ke pengklasifikasi unik yang menghasilkan probabilitas $y^{(l)}$, multikelas yaitu L adalah jumlah lapisan tersembunyi. Parameter jaringan dari L-th *hidden layer* dinotasikan dengan $\theta^{(l)}$ dan $W^{(l)} \in R^{R1 \times d}$, $b^{(l)} \in R^{R1}$, $W_s^{(l)} \in R^{Rm \times R1}$, $bs^1 \in R^m$, yaitu Rl dan d adalah urutan hidden node dan urutan hidden layer. ADL memulai proses pembelajarannya dari awal dengan keluaran struktur awal. ADL di sini disimulasikan di bawah prosedur *prequential test-then-train* yaitu data stream pertama kali digunakan untuk proses pengujian dilanjutkan dengan proses pelatihan. Sistem yang diperoleh langsung dari data dan dapat diperbarui secara *rekursif*, berfungsi untuk meningkatkan efisiensi memori dan perhitungan dari algoritma [26].

Adaptasi lebar jaringan dilakukan dalam *low level learning* yang terdiri dari dua mekanisme sebagai berikut, *Hidden node growing*. Mekanisme *Hidden node growing* dikendalikan oleh rumus NS yang mengevaluasi kekuatan generalisasi struktur jaringan diformulasikan yaitu $\pi = 1.3 \exp(-(\text{Bias}(\hat{y}^{(lw)}))^2) + 0.7$ dan mengatur derajat probabilitas dan *Hidden node pruning*. Berasal dari struktur yang memiliki kesamaan jika ada perubahan konsep dalam aliran data. Prinsip mekanisme *hidden node growing*, namun memperhatikan $\text{Var}(\hat{y}^{(lw)})$ dan juga dari $(\text{Bias}(\hat{y}^{(lw)}))^2$ [27].

Adaptasi kedalaman jaringan menggunakan pengurangan struktur kedalaman yang berbeda untuk menangani konsep drift sebagai prosedur pengurangan kompleksitas [28].

Solusi catastrophic forgetting Memiliki struktur fleksibel yang mencakup kedalaman yang berbeda memungkinkan ADL untuk mengatasi masalah melalui dua mekanisme. *Dynamic voting weight adaptation*. Setiap ada voting bobot β secara dinamis disesuaikan factor penurunan yang memainkan peran penting dalam saat beradaptasi dengan konsep *drift*.

Winning layer adaptation digunakan untuk menyesuaikan parameter jaringan dari *Winning layer adaptation*, yaitu $\theta^{(lw)}$ menggunakan label dari *batch* $B_k = (X_k, C_k) \in R^{T \times (n \times m)}$ secara *single-pass*. Hal ini diturunkan menggunakan minimalisasi *loss cross entropy*.

3. Tahap Pengujian Algoritma

Proses pelatihan pada ADL dilakukan dengan mekanisme pembentukan model untuk mendapatkan koefisien atau bobot. Proses ini menggunakan 80% data dari fitur yang dilatih secara acak menggunakan generator data pada *python*. Proses ini berfungsi membentuk model yang menjadi data uji pada data atau realtime system, proses pemakaian model menggunakan rekaman bobot terbaik pada iterasi yang dihasilkan terhadap fitur latih [29].

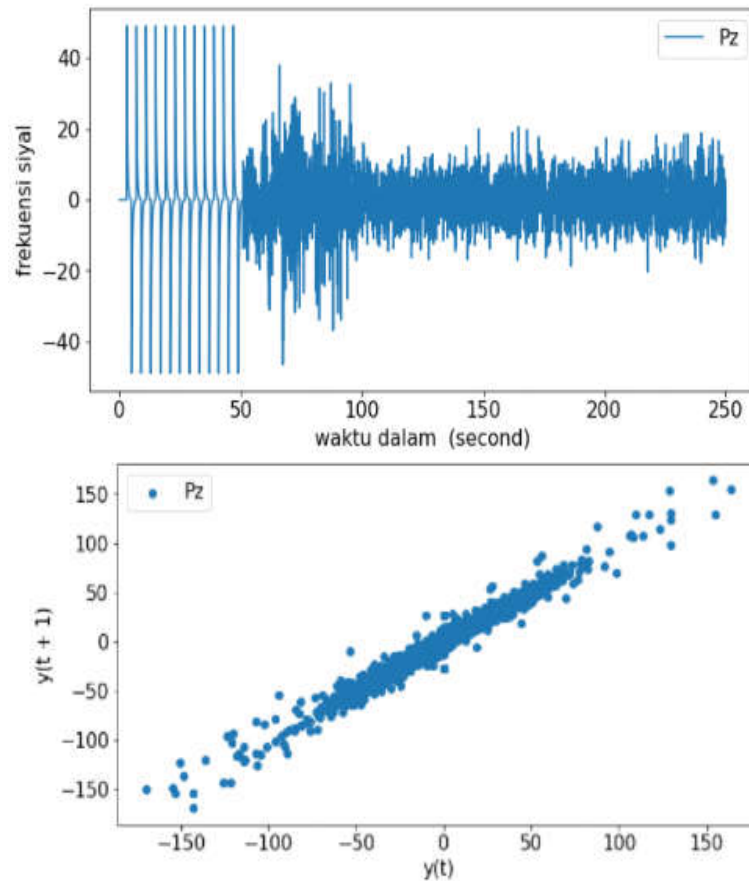
Proses Pengujian bertujuan untuk menguji ADL yang sudah dilatihkan pada tahap sebelumnya. Pengujian menggunakan data baru yang tidak pernah dilatihkan sebelumnya. Proses testing ini menggunakan metode perhitungan akurasi dan loss tiap pertumbuhan serta pengurangan jaringan ADL. Metode ini akan membagi data set yang ada menjadi bagian pelatihan 80% dan pengujian 20%. Pada penelitian ini, perhitungan akurasi untuk menghindari nilai akurasi classifier ADL yang bias [30].

Pada bagian testing system dilakukan pengujian keakuratan prediksi dari sistem yang telah dilatih sebelumnya. Keakuratan dari sistem prediksi pada ADL juga dipengaruhi oleh banyaknya data. Parameter yang digunakan sebagai evaluasi dari sistem yaitu menggunakan Akurasi, dan *loss* [31][32].

3. Hasil dan Analisis

3.1 Hasil Pengambilan Sampel data

Sinyal EEG memiliki domain waktu, sebanyak 21 channel pada datahsheet memiliki bentuk yang merepresentasikan perubahan sebaran data, bentuk RAW (Read Analog Write) pada lagplot mendefinisikan perubahan data setiap jeda waktu nilai $t=1$.

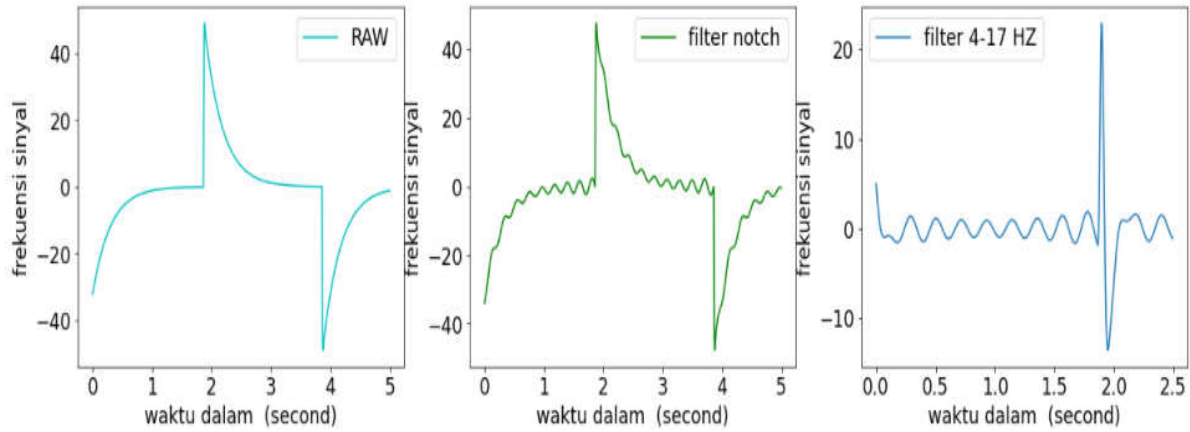


Gambar 4 Bentuk RAW dan Sebaran data Sinyal

Proses visualisasi pada gambar 4, membuktikan bahwa fitur memiliki rentang dan bentuk frekuensi yang berbeda disetiap channel, penggambaran *lag plot* pada sebaran data memudahkan untuk menganalisis kesamaan bentuk pada poin data yang diuji.

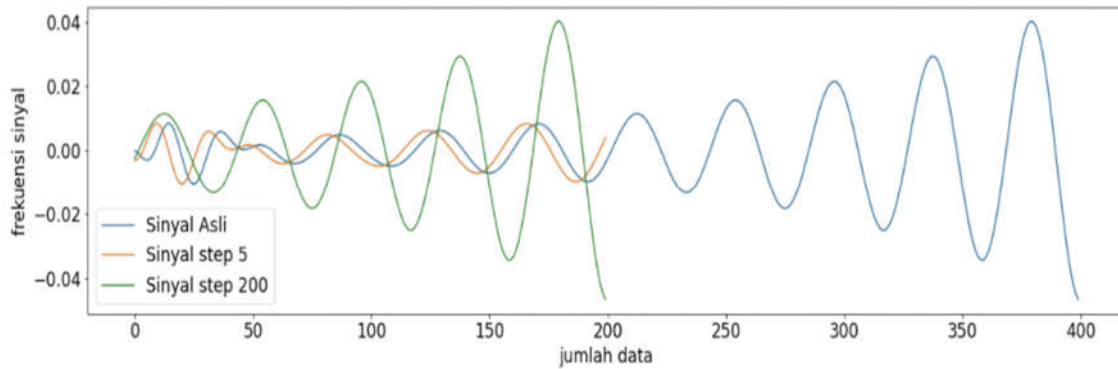
3.2 Hasil Pengolahan Data Fitur

Proses pengolahan sinyal EEG terdiri filtering yang berfungsi untuk mengilangkan noise dan menentukan rentang-rentang pada aktifitas manusia pada motorik maupun sensorik, dilanjutkan *windowing* data yang berfungsi sebagai melipatkan gandan informasi pada domain waktu, filter yang digunakan pada tahap ini terdiri dari *Notch Filter* yang menghilangkan derau pada noise yang diakibatkan dari catudaya, selanjutnya *Bandpass Filter* berguna untuk membatasi rentang berdasarkan aktifitas otak yang cenderung dinamis.



Gambar 5. Proses *Filtering* 4 -17 Hz (*Deltha*)

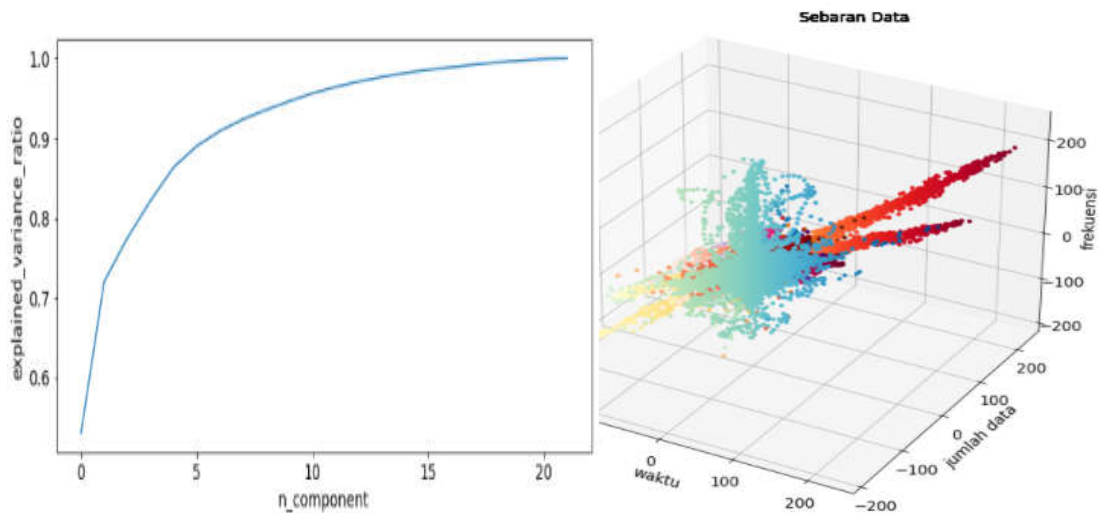
Proses selanjutnya *windowing* dengan panjang data 1 detik sebanyak 200 data akan menambahkan segmentasi lanjutan untuk pengumpulan informasi lebih lanjut tentang aktifitas respon stimulus terhadap domain waktu seperti tampak pada gambar 5



Gambar 6. Proses *Windowing*

Gambar 6 adalah proses *windowing* menjelaskan struktur sinyal filter yang diulang pada jumlah data 200 setiap detik untuk menambahkan proses sensitivitas data, pada deep learning proses ini sebagai sequence data yang mengandung informasi fitur yang merepresentasikan aktivitas stimulus pada domain waktu.

Proses *Dimentional reduction* dimaksudkan untuk menurunkan matrik yang mempunyai varian yang sama, proses ini melibatkan proyeksi orthogonal dari fitur pada sebaran data, dengan menggunakan perumusan matrik *Singulavar Value Decompositon* , matrik dari chanel fitur diektrak dengan proyeksi *Eugene vector* oleh PCA (*Principal Component Analisis*) .

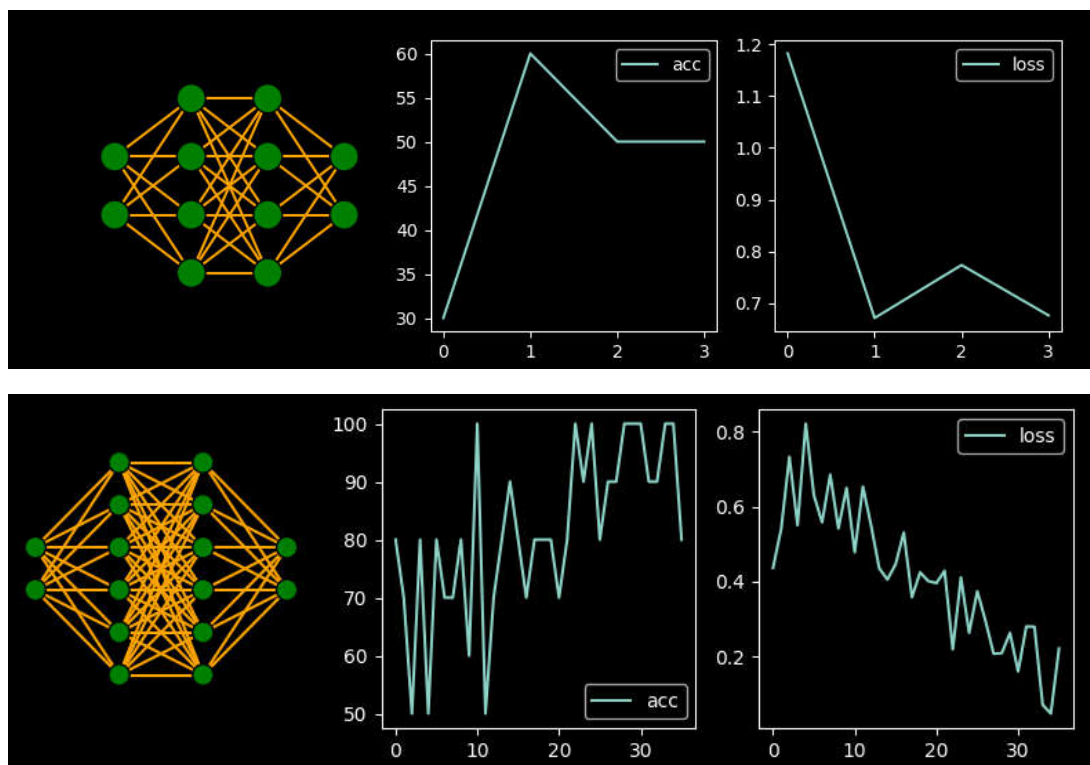


Gambar 7 Transformasi menggunakan PCA

Terlihat pada gambar 7 matrik yang awalnya sangat berkorelasi atau sangat mirip diekstrakt menjadi beberapa komponen yang terpisah pada PCA dengan bentuk sebaran matrik yang terpisah, setelah itu mengumpulkan kembali proyeksinya menjadi satu kesatuan berdasarkan target, dengan ini matrik yang akan diproses akan lebih sedikit tapi lebih informatif dibandingkan matrik sinyal sebelumnya.

3.3 Hasil Permodelan

Proses pelatihan memerlukan monitoring terhadap pengaruh *loss* dan akurasi, tampilan secara berkala menampilkan perubahan *node* dan *size* jaringan yang terjadi pada proses pelatihan, proses ini terjadi pada *backend matplotlib* pada pemrograman *python* dengan *handle* data setelah nilai *output* terbentuk.



Gambar 8. Proses pelatihan model

Pada gambar 8 menampilkan hasil yang diharapkan pada setiap prediksi, angka didapatkan dari nilai yang dirubah dalam bentuk *plot* animasi yang secara berkala menghapus tampilan sebelumnya, proses ini sangat membantu dalam menyimpulkan baik buruknya hasil koleksi data saat dilatih.

3.4 Validasi Permodelan

Tabel 1 Pengujian Subjek menggunakan model ADL

Subjek	Mean ADL	
	Akurasi %	Waktu (detik)
A	96.858310	± 0.002974
B	96.497968	± 0.002859
C	96.431156	± 0.002496
D	96.559212	± 0.002797
F	95.657257	± 0.002698
A	96.858310	± 0.002974

Hasil yang diperoleh pada pengujian sesuai tabel 1 tergolong baik dengan rata-rata prediksi diatas 96% pada pengujian disimpulkan terdapat keberhasilan yang cukup signifikan untuk suatu model. Pengujian pernyataan berpikir aktif dengan teknik rekam diuji langsung secara real time setiap 1 detik atau 200 data, pengujian ini terbagi dalam 3 langkah dengan 100 kali pengulangan data, pengujian dimaksudkan untuk menguji model secara real time dan kemampuan berpikir responden. sangat.

3.3. Penelitian Terdahulu

Pada umumnya Pemodelan *Deep Learning* masih menggunakan pengaturan manual pada jaringan [33]. Hal ini merupakan masalah mendasar pada variabilitas respon individu oleh karena itu diusulkan pembaruan menggunakan *Autonomous Deep Learning (ADL)*. Pemodelan ADL memiliki mekanisme adaptif untuk meningkatkan stabilitas proses pembelajaran akibat inisialisasi jaringan yang buruk. Pada jaringan konstruksi dibangun secara otomatis. Model ADL akan lebih adaptif dan dapat meningkatkan akurasi dalam klasifikasi.

Table 2. Performa Komparasi dengan penelitian terdahulu

Referensi	Tahun	Metode	Akurasi
Tabar and Halici [34]	2016	1D CNN with SAE 3 Class	70.0%
Lawhern et al.[35]	2018	CNN with depth and separable Convolutions 3 Class	69.0%
Schirrmeister et al.[36]	2017	CNN with depth and separable Convolutions 3 Class	72.0%
Sakhavi, et al[8]	2018	Temporal Features with FBCSP dan CNN 3 Class	74,5%
Umar et al [37]	2019	CCNN 3 Class	74,7%
Penelitian ADL	2024	ADL 3 Class	96%

Tabel 2 menjelaskan bahwa penggunaan model ADL dapat meningkatkan akurasi hingga 96% dibandingkan model lainnya. Hal ini menunjukkan bahwa ADL cocok digunakan dalam penggunaan klasifikasi gerakan tangan.

4. Kesimpulan

Model ADL dapat digunakan untuk memprediksi gerakan tangan, hal ini dibuktikan dengan data pengujian yang diambil dari 6 subjek dengan hasil rata-rata akurasi 96% pada subjek mandiri. Pada bagian model ekstraksi fitur berfungsi untuk memperkecil matriks dimensi tanpa kehilangan informasi fitur, matriks tersebut didekomposisi menggunakan PCA dengan rasio perbedaan matriks.

Pengakuan dan Penghargaan

Penelitian ini didanai oleh DIREKTORAT JENDERAL PENDIDIKAN TINGGI, RISET DAN TEKNOLOGI (Dosen Pemula Reguler Penelitian) No. 0667/E5/AL.04/2024. Terima kasih juga kami sampaikan kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) dan Fakultas Teknik (FT) UNIVERSITAS PGRI BANYUWANGI.

References

- [1] J. K. Sains, "Stigma Penyandang Disabilitas dalam Bekerja di Indonesia : Literature Review Stigma of People with Disabilities in Working in Indonesia : Literature Review Jurnal Kolaboratif Sains (JKS)," vol. 7, no. 3, pp. 1076–1086, 2024, doi: 10.56338/jks.v7i3.4669.
- [2] Hastuti, R. K. Dewi, R. P. Pramana, and H. Sadaly, *Kendala Mewujudkan Pembangunan Inklusif Penyandang Disabilitas*. 2020. [Online]. Available: www.smeru.or.id.
- [3] J. E. Prynne *et al.*, "Disability among older people: Analysis of data from disability surveys in six low-and middle-income countries," *Int. J. Environ. Res. Public Health*, vol. 18, no. 13, 2021, doi: 10.3390/ijerph18136962.
- [4] C. A. Calderon-Cordova, C. Ramirez, V. Barros, and G. Punin, "Design and Deployment of Grasp Control System applied to robotic hand prosthesis," *IEEE Lat. Am. Trans.*, vol. 15, no. 2, pp. 181–188, 2017, doi: 10.1109/TLA.2017.7854610.
- [5] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–11, 2018, doi: 10.1002/widm.1254.
- [6] R. Roy, D. Sikdar, M. Mahadevappa, and C. S. Kumar, "A fingertip force prediction model for grasp patterns characterised from the chaotic behaviour of EEG," *Med. Biol. Eng. Comput.*, vol. 56, no. 11, pp. 2095–2107, 2018, doi: 10.1007/s11517-018-1833-0.
- [7] C. Timplalexis, K. Diamantaras, and I. Chouvarda, "Classification of sleep stages for healthy subjects and patients with minor sleep disorders," *Proc. - 2019 IEEE 19th Int. Conf. Bioinforma. Bioeng. BIBE 2019*, no. January 2020, pp. 344–351, 2019, doi: 10.1109/BIBE.2019.00068.
- [8] S. Sakhavi, C. Guan, and S. Yan, "Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018, doi: 10.1109/TNNLS.2018.2789927.
- [9] K. Lee, D. Liu, L. Perroud, R. Chavarriaga, and J. del R. Millán, "A brain-controlled exoskeleton with cascaded event-related desynchronization classifiers," *Rob. Auton. Syst.*, vol. 90, pp. 15–23, 2017, doi: 10.1016/j.robot.2016.10.005.
- [10] Y. Chen, C. Dai, and W. Chen, "Cross-Comparison of EMG-to-Force Methods for Multi-DoF Finger Force Prediction Using One-DoF Training," *IEEE Access*, vol. 8, pp. 13958–13968, 2020, doi: 10.1109/ACCESS.2020.2966007.

- [11] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning Pooling for Convolutional Neural Network," *Neurocomputing*, vol. 224, pp. 96–104, 2017, doi: 10.1016/j.neucom.2016.10.049.
- [12] S. H. Park, D. Lee, and S. G. Lee, "Filter Bank Regularized Common Spatial Pattern Ensemble for Small Sample Motor Imagery Classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 498–505, 2018, doi: 10.1109/TNSRE.2017.2757519.
- [13] G. Bressan, G. Cisotto, and G. R. Müller-putz, "Deep Learning-Based Classification of Fine Hand Movements from Low Frequency EEG," pp. 1–14, 2021.
- [14] X. Yin and X. Liu, "Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, 2018, doi: 10.1109/TIP.2017.2765830.
- [15] K. Anam, S. Bukhori, F. S. Hanggara, and M. Pratama, "Subject-independent Classification on Brain-Computer Interface using Autonomous Deep Learning for finger movement recognition," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 447–450, 2020, doi: 10.1109/EMBC44109.2020.9175718.
- [16] W. W. Sakti, K. Anam, S. B. Utomo, B. Marhaenanto, and S. Nahela, "Artificial Intelligence IoT based EEG Application using Deep Learning for Movement Classification," in *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, Oct. 2021, pp. 192–196. doi: 10.23919/EECSI53397.2021.9624269.
- [17] M. Pratama, C. Za'in, A. Ashfahani, Y. S. Ong, and W. Ding, "Automatic construction of multi-layer perceptron network from streaming examples," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 1171–1180, 2019, doi: 10.1145/lp0678.
- [18] R. Widadi, B. A. Widodo, and D. Zulherman, "Klasifikasi Sinyal EEG pada Sistem BCI Pergerakan Jari Manusia Menggunakan Convolutional Neural Network," *Techno.Com*, vol. 19, no. 4, pp. 459–467, 2020, doi: 10.33633/tc.v19i4.4119.
- [19] N. Shajil, S. Mohan, P. Srinivasan, J. Arivudaiyanambi, and A. Arasappan Murrugesan, "Multiclass Classification of Spatially Filtered Motor Imagery EEG Signals Using Convolutional Neural Network for BCI Based Applications," *J. Med. Biol. Eng.*, no. 0123456789, 2020, doi: 10.1007/s40846-020-00538-3.
- [20] N. Rashid, J. Iqbal, A. Javed, M. I. Tiwana, and U. S. Khan, "Design of Embedded System for Multivariate Classification of Finger and Thumb Movements Using EEG Signals for Control of Upper Limb Prosthesis," *Biomed Res. Int.*, vol. 2018, 2018, doi: 10.1155/2018/2695106.
- [21] Y. Zhang, C. S. Nam, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Temporally constrained sparse group spatial patterns for motor imagery BCI," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3322–3332, 2019, doi: 10.1109/TCYB.2018.2841847.
- [22] T. jian Luo, C. le Zhou, and F. Chao, "Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–18, 2018, doi: 10.1186/s12859-018-2365-1.
- [23] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," *arXiv*, pp. 791–800, 2018.
- [24] M. S. Bascil, "A New Approach on HCI Extracting Conscious Jaw Movements Based on EEG Signals Using Machine Learnings," *J. Med. Syst.*, vol. 42, no. 9, pp. 1–11, 2018, doi: 10.1007/s10916-018-1027-1.
- [25] A. Ashfahani and M. Pratama, "Autonomous deep learning: Continual learning approach

- for dynamic environments," *SIAM Int. Conf. Data Mining, SDM 2019*, pp. 666–674, 2019, doi: 10.1137/1.9781611975673.75.
- [26] P. P. Angelov, X. Gu, and J. C. Principe, "Autonomous learning multimodel systems from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2213–2224, 2018, doi: 10.1109/TFUZZ.2017.2769039.
- [27] A. M. Sinaga *et al.*, "No Covariance structure analysis of health-related indicators in elderly people living at home, focusing on subjective health" *IEEE Trans. Softw. Eng.*, vol. 24, no. 4, pp. 233–244, 2019, doi: 10.1145/1390630.1390641.
- [28] J. Muschelli, "ROC and AUC with a binary predictor: a potentially misleading metric," *arXiv*, 2019, doi: 10.1007/s00357-019-09345-1.
- [29] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level Aggregation for Speaker Recognition in the Wild," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 5791–5795, 2019, doi: 10.1109/ICASSP.2019.8683120.
- [30] M. Pratama, C. Za'in, A. Ashfahani, Y. S. Ong, and W. Ding, "Automatic construction of multi-layer perceptron network from streaming examples," *Int. Conf. Inf. Knowl. Manag. Proc.*, no. 11, pp. 1171–1180, 2019, doi: 10.1145/lp0678.
- [31] W. Abbas and N. A. Khan, "DeepMI: Deep Learning for Multiclass Motor Imagery Classification," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018-July, pp. 219–222, 2018, doi: 10.1109/EMBC.2018.8512271.
- [32] L. Yao, Z. Fang, Y. Xiao, J. Hou, and Z. Fu, "An Intelligent Fault Diagnosis Method for Lithium Battery Systems Based on Grid Search Support Vector Machine," *Energy*, vol. 214, p. 118866, 2021, doi: 10.1016/j.energy.2020.118866.
- [33] R. Lq, W. Zlwk, and L. D. Huv, "Siddique, Fathma, Shadman Sakib, and Md Abu Bakr Siddique. 'Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers.' 2019 5th International Conference on Advanc".
- [34] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of {EEG} motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, p. 16003, Nov. 2016, doi: 10.1088/1741-2560/14/1/016003.
- [35] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, p. 56013, 2018.
- [36] R. T. Schirmermeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017, doi: <https://doi.org/10.1002/hbm.23730>.
- [37] S. Umar, M. Alsulaiman, and G. Muhammad, "Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 542–554, 2019, doi: 10.1016/j.future.2019.06.027.