

PENGAPLIKASIAN *DEEP LEARNING* DENGAN DESAIN ARSITEKTUR JST UNTUK TUGAS REGRESI DAN KLASIFIKASI

Serka Dadang Lesmana¹⁾ Sertu Antonius Nenet²⁾ dan Letda Inf Kasiyanto³⁾

¹⁾Jurusan Elkasista, Prodi D4 Elektronika Sistem Senjata Poltekad Kodiklatat
E - mail : lesmanadadang19@gmail.com¹,

ARTICLE TITLE (TEMPLATE FOR MANUSCRIPT FOR JURNAL TEKNOLOGI INDONESIA)

Abstract: Technological developments have driven the application of Deep Learning, particularly Artificial Neural Networks (ANN), in solving regression and classification tasks. ANNs consist of interconnected artificial neurons that are effective in classification, prediction, and pattern recognition. This study aims to build and analyze the impact of variations in Artificial Neural Network (ANN) architecture on prediction accuracy, using a quantitative experimental method with a controlled randomized design. The BostonHousing.csv dataset was used for regression, and the Iris.csv dataset for classification. Regression evaluation uses MSE, R^2 , and MAE; while classification uses accuracy and confusion matrix. The best regression results were obtained from the 4-hidden-layer architecture (512, 256, 128, 64), with ReLU, sigmoid, tanh, and ReLU activation functions, a learning rate of 0.001, achieving an R^2 of 89.2% and an MAE of 1.94. For classification, the best architecture (128, 64, 32, 16) with a softmax output yielded an accuracy of 99.8% and a model accuracy of 100%.

Keywords – deep learning, artificial neural network, regression, classification, ANN architecture

Abstrak: Perkembangan teknologi mendorong penerapan Deep Learning, khususnya Artificial Neural Network (ANN), dalam menyelesaikan tugas regresi dan klasifikasi. ANN terdiri dari neuron buatan yang saling terhubung dan efektif dalam klasifikasi, prediksi, serta pengenalan pola. Penelitian ini bertujuan membangun dan menganalisis pengaruh variasi arsitektur Jaringan Syaraf Tiruan (JST) terhadap akurasi prediksi, menggunakan metode kuantitatif eksperimental dengan desain acak terkontrol. Dataset BostonHousing.csv digunakan untuk regresi dan Iris.csv untuk klasifikasi. Evaluasi regresi menggunakan MSE, R^2 , dan MAE; sedangkan klasifikasi menggunakan akurasi dan confusion matrix. Hasil terbaik regresi diperoleh dari arsitektur 4 hidden layer (512, 256, 128, 64), fungsi aktivasi relu, sigmoid, tanh, relu, learning rate 0.001, dengan R^2 89,2% dan MAE 1,94. Untuk klasifikasi, arsitektur terbaik (128, 64, 32, 16) dengan softmax output menghasilkan akurasi 99,8% dan akurasi model 100%.

Kata Kunci – deep learning, jaringan saraf tiruan, regresi, klasifikasi, arsitektur JST

PENDAHULUAN

Era digitalisasi menunjukan perkembangan teknologi yang semakin pesat dengan penerapan Deep Learning yang saat ini populer di kalangan akademisi dan profesional, yang menjadi pendekatan utama dalam menyelesaikan berbagai permasalahan seperti tugas regresi dan klasifikasi. Deep Learning merupakan salah satu cabang dari Machine Learning (ML) dengan memanfaatkan Deep Neural Network untuk menyelesaikan berbagai permasalahan dalam domain Machine Learning (ML) (Hayati & Hadiprajitno, 2021).

Politeknik Angkatan Darat (Poltekad) pendidikan tinggi vokasi yang bertugas untuk memberikan inovasi teknologi dengan bertujuan mendukung tugas OMSP dan OMP khususnya dapat menerapkan dan mengawaki alutsista yang sudah berbasis teknologi (Elkasista et al., 2022). Pada penelitian praktikum ini, bagian dari mata kuliah Praktek Jaringan Saraf Tiruan yang sangat berguna untuk identifikasi. Sebagai dasar dalam melakukan penelitian praktikum, akan membahas beberapa bagian seperti dasar-dasar Neural Network, yang merupakan elemen kunci dalam Deep Learning dan mempelajari lebih dalam bagaimana Neural Network bekerja dan dilatih untuk menyelesaikan berbagai tugas dengan efisien. Neural Network bekerja dengan meniru cara kerja otak manusia dalam mengenali pola dan membuat

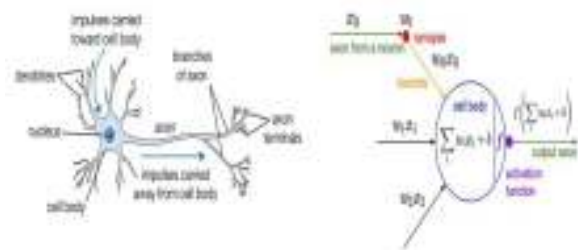
keputusan berdasarkan data yang diberikan (Syahputro et al., 2020). Deep Learning yang digunakan, termasuk:

1. Computer Vision – Pengenalan wajah, deteksi objek, segmentasi gambar, dan analisis citra medis. Contohnya, sistem pengenalan wajah pada smartphone atau pendeteksian kanker dari hasil pencitraan medis.
2. Natural Language Processing (NLP) – Penerjemahan bahasa otomatis, chatbot, analisis sentimen, dan pengenalan suara. Misalnya, Google Translate dan asisten virtual seperti Siri atau Alexa.

Neural Network

Neural network atau dalam Bahasa Indonesia dikenal dengan istilah Jaringan Syaraf Tiruan (JST) adalah model komputasi yang meniru cara kerja neuron dalam otak manusia. Di dalam otak, neuron-neuron saling terhubung membentuk jaringan kompleks yang memungkinkan aliran informasi secara simultan. Setiap neuron menerima sinyal dari neuron lain, memprosesnya, dan meneruskan sinyal tersebut ke neuron berikutnya berdasarkan kekuatan hubungan (atau bobot) antar neuronelka. Neural network terdiri dari lapisan-lapisan neuron buatan yang berfungsi untuk mengenali pola dan menyelesaikan berbagai tugas, seperti klasifikasi gambar, pemrosesan bahasa alami, dan prediksi data. Model matematis

dari neuron buatan biasanya melibatkan bobot (weights), bias, serta fungsi aktivasi yang menentukan keluaran dari neuron berdasarkan masukan yang diterima. Pada gambar 1.1. menggambarkan bagaimana sebuah neuron buatan bekerja, termasuk bagaimana sinyal diproses melalui fungsi aktivasi sebelum diteruskan ke neuron lain dalam jaringan (Nwadiugwu, 2020).



Gambar 1.1. Ilustrasi Bagaimana Artificial Neural Network Meniru Jaringan Otak Manusia

Setiap neuron dalam Neural Network menerima sejumlah input, di mana masing-masing input dikalikan dengan bobot (weight) tertentu. Selanjutnya, neuron menjumlahkan hasil perkalian ini (weighted sum) dan menambahkan bias untuk menyesuaikan nilai akhir. Setelah proses penjumlahan, hasilnya akan diteruskan ke fungsi aktivasi (activation function), yang berperan dalam menentukan apakah neuron akan "aktif" dan seberapa besar sinyal yang akan diteruskan ke lapisan berikutnya. Fungsi aktivasi memiliki peranan penting untuk memperkenalkan non-linearitas ke dalam jaringan, agar model dapat

menangani permasalahan yang kompleks. Output yang dihasilkan dari fungsi aktivasi inilah yang menjadi keluaran dari neuron sebelum dikirim ke neuron lainnya dalam jaringan (Akbar Nugraha et al., 2022).

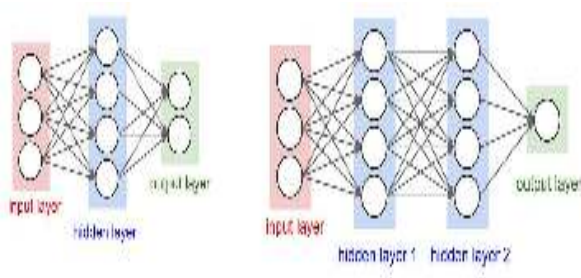
1.2. Arsitektur Neural Network

Artificial Neural Network (ANN) adalah salah satu model dalam kecerdasan buatan yang terinspirasi dari cara kerja otak manusia. ANN terdiri dari kumpulan neuron buatan yang saling terhubung dan bekerja sama untuk memproses informasi serta menyelesaikan berbagai tugas komputasi, seperti klasifikasi, prediksi, dan pengenalan pola (Raup et al., 2022). Secara umum, arsitektur ANN terdiri dari tiga jenis lapisan utama:

1. Input Layer – Lapisan ini berfungsi sebagai pintu masuk bagi data yang akan diproses oleh jaringan. Setiap neuron dalam lapisan ini mewakili fitur atau variabel dalam data.
2. Hidden Layers – Lapisan tersembunyi merupakan inti dari ANN yang bertugas mengekstraksi pola dan hubungan dari data. Setiap neuron dalam lapisan ini menerapkan operasi matematis dengan bobot (weights), bias, dan fungsi aktivasi untuk menentukan outputnya. Jaringan dengan lebih dari satu hidden layer disebut Deep Neural Network (DNN).
3. Output Layer – Lapisan ini menghasilkan hasil akhir dari model, baik

dalam bentuk klasifikasi (contoh: menentukan apakah gambar adalah kucing atau anjing) maupun regresi (contoh: memprediksi harga rumah).

ANN bekerja dengan mengoptimalkan bobot dan bias melalui proses pelatihan menggunakan algoritma backpropagation dan optimisasi berbasis gradient descent. Dengan arsitektur yang tepat, ANN dapat digunakan dalam berbagai aplikasi, seperti pengenalan wajah, deteksi objek, pemrosesan bahasa alami, dan sistem rekomendasi (Blessinda, 2024). Dalam implementasinya dalam dunia militer, metode backpropagation berfungsi untuk mengenali bobot atau presentase jenis suara tembakan SS2. Sensor mikrofon menerima suara tembakan yang nantinya dengan penggunaan raspberry pi 4 akan menghasilkan output yang mempresentasikan suara senjata ringan SS2- V1 pada layar lcd dan speaker (Gunawan et al., 2021).



Gambar 1.2. Arsitektur Dasar Artificial Neural Network

Gambar 1.2. dikenal sebagai Multi-Layer Perceptron (MLP) atau Fully-Connected

Layer, di mana setiap neuron di satu lapisan terhubung dengan semua neuron di lapisan berikutnya, berikut penjelasannya:

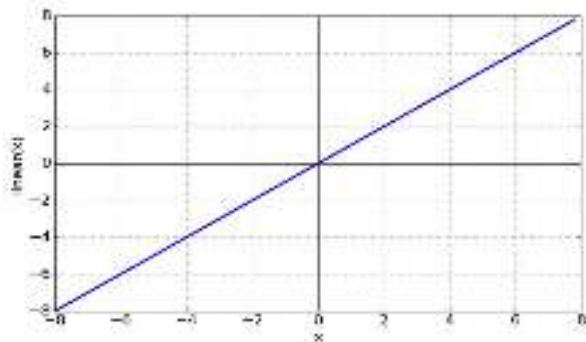
1. Input Layer terdiri dari 3 neuron, yang masing-masing merepresentasikan fitur dari data masukan.
2. Hidden Layer memiliki 4 neuron, yang bertugas melakukan pemrosesan dan ekstraksi fitur menggunakan bobot (weight) dan fungsi aktivasi.
3. Output Layer terdiri dari 2 neuron, yang memberikan hasil akhir berdasarkan klasifikasi atau regresi yang dilakukan oleh jaringan.

1.3. Activation Function

Activation function berperan penting dalam menentukan apakah sebuah neuron akan aktif atau tidak, berdasarkan nilai weighted sum dari input yang diterimanya. Tanpa activation function, neural network hanya akan melakukan operasi linear sederhana, yang membatasi kemampuannya dalam menyelesaikan masalah kompleks. Dengan demikian, fungsi aktivasi sangat penting untuk memperkenalkan non-linearitas ke dalam model, memungkinkan jaringan untuk menangani berbagai pola dan hubungan yang lebih rumit dalam data. Secara umum, activation function dapat dikategorikan menjadi dua jenis yaitu activation linear dan non-linear (Efrian & Latifa, 2022).

1.3.1. Activation Function Linear

Activation function linear menghasilkan output yang proporsional terhadap input, sehingga tidak memperkenalkan non-linearitas. Akibatnya, jaringan dengan fungsi aktivasi linear tidak dapat menyelesaikan masalah yang membutuhkan pemetaan kompleks. Penggunaan fungsi aktivasi linear dalam jaringan saraf berlapis-lapis (multi-layer neural network) memiliki keterbatasan. Jika semua neuron dalam jaringan hanya menggunakan fungsi linear, maka seluruh jaringan akan tetap bekerja sebagai fungsi linear, tidak peduli seberapa banyak lapisan yang digunakan (Setiawan, 2011).

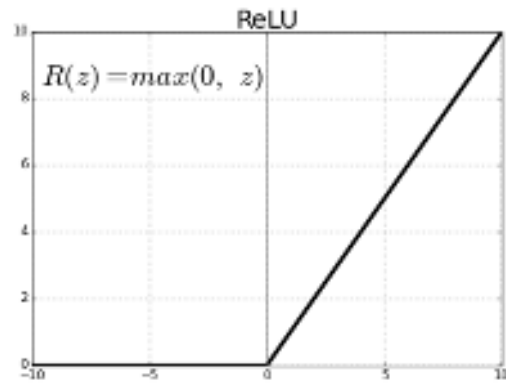


Gambar 1.3.1 Activation Function Linear

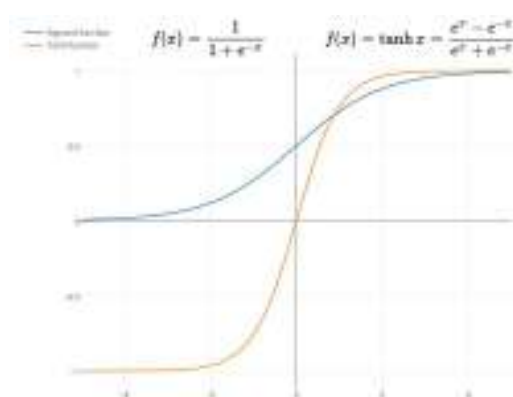
1.3.2. Activation Function Non-Linear

Activation function non-linear berfungsi untuk mempelajari hubungan yang lebih kompleks dalam data dengan memperkenalkan non-linearitas. Contoh fungsi aktivasi non-linear yang umum digunakan meliputi ReLU (Rectified Linear Unit), Sigmoid, dan Tanh. Penggunaan activation function yang tepat sangat

bergantung pada jenis masalah yang ingin diselesaikan dan karakteristik data yang digunakan dalam pelatihan model (Schmidt-Hieber, 2020).



Gambar 1.3.2. Activation Function Sigmoid dan Tanh



Gambar 1.3.3. Activation Function Relu

Secara default, activation function pada sebuah neuron dapat dianggap sebagai fungsi linear. Jika sebuah neuron hanya menggunakan fungsi linear, maka outputnya akan sama dengan weighted sum dari input ditambah dengan bias, tanpa adanya transformasi tambahan., karena lapisan tambahan tidak menambah kemampuan pemetaan non-linear. Dengan demikian, agar neural network dapat

menangani masalah yang lebih rumit, diperlukan fungsi aktivasi non-linear, seperti ReLU, Sigmoid, atau Tanh, untuk memperkenalkan kompleksitas dalam pemrosesan data (Pratama et al., 2023).

METODE PENELITIAN

Penelitian praktikum ini menggunakan studi kuantitatif eksperimental dengan desain acak terkontrol (Ismail et al., n.d.). Arsitektur JST diuji pada dua dataset berbeda: regresi dengan dataset BostonHousing.csv. (data produksi) dan klasifikasi dengan dataset Iris.csv. (data citra). Evaluasi dilakukan menggunakan metrik *Mean Squared Error* (MSE) atau R^2 score, nilai error (MAE) untuk regresi dan akurasi untuk klasifikasi menggunakan metrik *confusion* yang masing-masing berisi implementasi model Neural Network untuk menyelesaikan kedua tugas tersebut (Andriani, 2021).

Pada penelitian praktikum ini, akan mempelajari bagaimana Deep Learning diterapkan dalam dua jenis tugas utama, yaitu regresi dengan dataset BostonHousing.csv dan klasifikasi dengan dataset Iris.csv. Dengan menggunakan aplikasi platform google colab untuk menjalankan program dengan dua notebook berformat .ipynb yang masing-masing berisi implementasi model Neural Network untuk menyelesaikan kedua tugas tersebut, melakukan Langkah-Langkah praktikum sebagai berikut (Aminullah, 2021):

1. Mengetahui Struktur Notebook:
 - a. Notebook pertama berisi model yang digunakan untuk tugas regresi, di mana model bertujuan untuk memprediksi nilai kontinu berdasarkan data input.
 - b. Notebook kedua berisi model untuk tugas klasifikasi, di mana model bertujuan untuk mengkategorikan data ke dalam beberapa kelas berdasarkan pola yang dipelajari.
2. Memahami Parameter yang Dapat Diubah. Dalam kedua notebook ini, terdapat beberapa parameter yang dapat diubah untuk meningkatkan performa model. Beberapa parameter utama yang akan dieksplorasi meliputi:
 - a. Jumlah Neuron dan Hidden Layers – Mengatur jumlah lapisan tersembunyi dan jumlah neuron dalam setiap lapisan untuk melihat bagaimana arsitektur jaringan mempengaruhi performa model.
 - b. Learning Rate – Mengontrol seberapa cepat model belajar selama proses training. Nilai yang terlalu besar dapat menyebabkan model tidak konvergen, sedangkan nilai yang terlalu kecil bisa membuat proses pelatihan sangat lambat.
 - c. Epochs – Menentukan berapa kali seluruh dataset akan diproses selama pelatihan model.

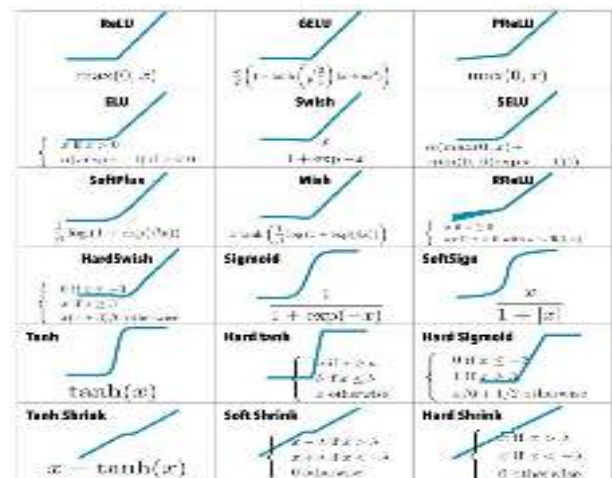
- d. Activation Function – Menggunakan fungsi aktivasi yang berbeda (seperti ReLU, Sigmoid, atau Tanh) untuk memahami dampaknya terhadap kinerja model.
3. Eksperimen dan Evaluasi
 - a. Setelah mengubah parameter tertentu, kita akan melatih ulang model dan menganalisis hasilnya menggunakan metrik evaluasi yang sesuai:
 - 1) Untuk regresi, metrik yang digunakan bisa berupa Mean Squared Error (MSE) atau R² score.
 - 2) Untuk klasifikasi, kita dapat mengevaluasi model dengan akurasi dan confusion matrix.
 - b. Tujuan utama dari eksperimen ini adalah mencari kombinasi parameter terbaik yang dapat menghasilkan performa optimal untuk setiap tugas.
4. Membandingkan Hasil
 - a. Setelah melakukan beberapa kali eksperimen dengan parameter yang berbeda, akan membandingkan hasil model sebelum dan sesudah tuning parameter.
 - b. Analisis ini akan membantu memahami bagaimana parameter tertentu mempengaruhi kinerja model serta bagaimana cara meningkatkan akurasi dan efisiensi model Deep Learning.

5. Rumus matematis, fungsi linear ini dapat ditulis sebagai:

$$y = \sum(w_i \cdot x_i) + b$$

di mana:

- x_i adalah input,
- w_i adalah bobot (weight),
- b adalah bias, dan
- y adalah output dari neuron.



Gambar 2 Variasi Activation Function

III. HASIL DAN DISKUSI

3.1 Hasil Pengamatan Praktikum Tugas Regresi

Hasil pengamatan praktikum tugas regresi ini untuk mengetahui model regresi dalam memprediksi harga rumah dengan analisis data dari dataset bostonhousing.csv, yang berisi informasi mengenai berbagai karakteristik perumahan di wilayah Boston. Proses yang dilakukan eksplorasi data, scaling data dan mencari arsitektur model terbaik dengan menambah jumlah hidden layer, mengubah jumlah neuron, activation function, dan learning rate. Hasil model

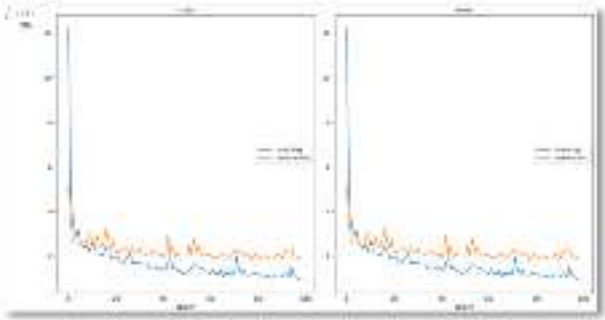
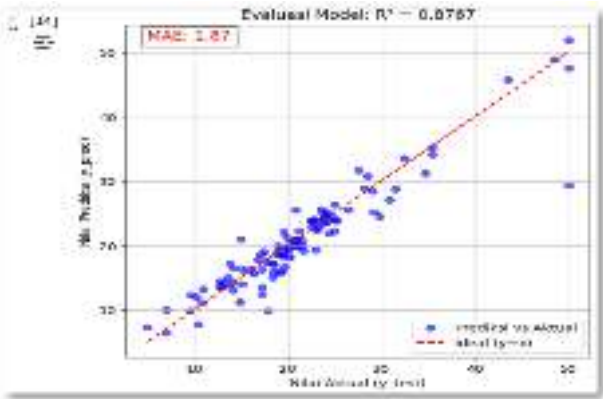
dianalisis menggunakan metrik evaluasi model seperti nilai koefisien determinasi (R^2), Mean Squared Error (MSE) atau Mean Absolute Error (MAE), dan uji signifikansi koefisien (Elmuna, 2023).

3.1.1 Menambah Hidden Layer

Mengubah desain arsitektur dari ANN dengan menambah hidden layer sampai 6 step-by-step untuk melihat fenomena dari setiap perubahan dengan jumlah neuron yang tetap (256).

Tabel 3.1.1. Data Pengamatan Jumlah Hidden Layer

Output Hidden Layer			
No	Jumlah Hidden Layer	Akurasi (R^2) Koefesien Determinasi	Error (MAE) % Mean Absolute Error
1	1	0.8533 / 85.33 %	1.87
2	2	0.8524 / 85.24 %	2.11
3	3	0.8502 / 85.02 %	2.13
4	4	0.8767 / 87.67 %	1.87
5	5	0.8753 / 87.53 %	1.96
6	6	0.8657 / 86.02 %	1.90



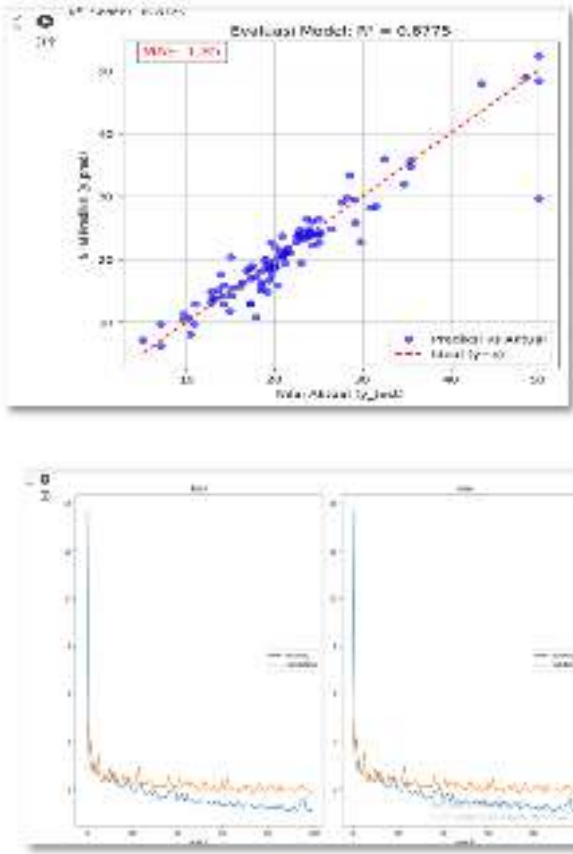
Gambar 3.1.1. Evaluasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.1.2 Mengubah Jumlah Neuron

Mengubah jumlah neuron dengan 4 hidden layer secara random sehingga menemukan hasil yang terbaik. Jumlah neuron dapat menggunakan sistem piramida mengerucut ke bawah, sistem naik, dan campuran. Menggunakan parameter nilai dari Biner kelipatan dari bit seperti 1,2,4,8,16,32 dst lebih baik dalam training disesuaikan dengan kompleksitas data.

Tabel 3.1.2. Data Pengamatan Perubahan Jumlah Neuron

No	Jumlah Hidden Layer						Akurasi (R ²)	Error (MAE) %
	1	2	3	4	5	6		
	Jumlah Neuron							
1	256	750	512	1000	-	-	0.8560 / 85.60 %	2.11
2	600	500	400	300	-	-	0.8548 / 85.48 %	1.99
3	64	128	235	256	-	-	0.8337 / 83.37 %	2.15
4	512	256	128	64	-	-	0.8775 / 87.75 %	1.85
5	16	32	64	128	-	-	0.8630 / 86.30 %	1.91



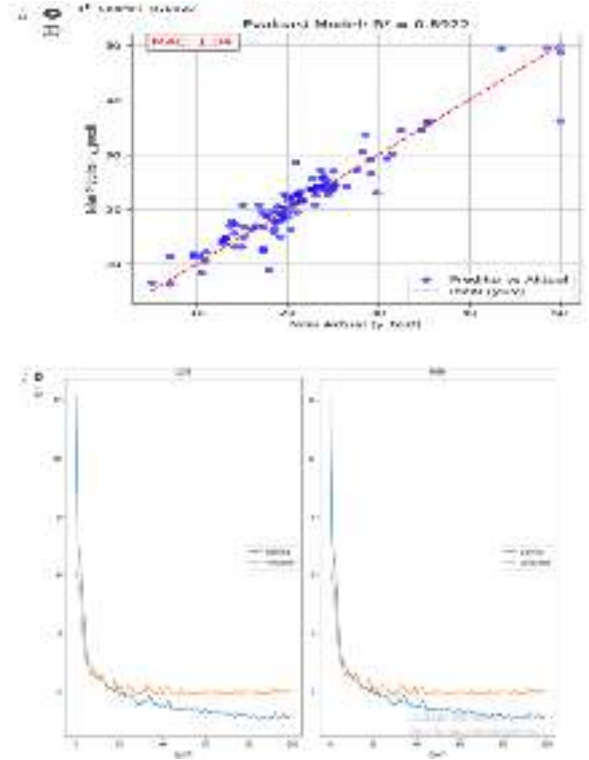
Gambar 3.1.2. Evaluasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.1.3 Mengubah Activation Function

Mengubah activation function dengan relu, sigmoid, dan tanh untuk mengetahui penggunaan arsitektur yang tepat berdasarkan kompleksitas data sehingga didapatkan nilai terbaik.

Tabel 3.1.3. Data Pengamatan Perubahan Activation Function

No	Jumlah Hidden Layer						Akurasi (R ²)	Error (MAE) %
	1	2	3	4	5	6		
	Activation Funtion							
1	sigmoid	sigmoid	sigmoid	Sigmoid	-	-	0.4262	3.54
2	tanh	tanh	tanh	Tanh	-	-	0.7981	2.26
3	tanh	relu	sigmoid	Tanh	-	-	0.7895	2.26
4	relu	sigmoid	tanh	Relu	-	-	0.8922	1.94
5	sigmoid	tanh	relu	Sigmoid	-	-	- 0.0002	5.96



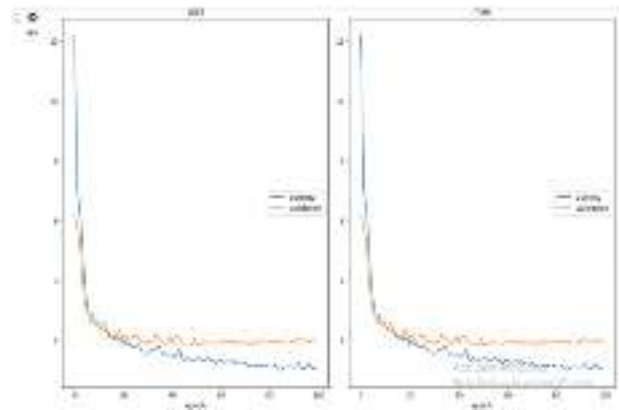
Gambar 3.1.3. Evaluasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.1.4 Mengubah Learning Rate

Setelah mendapatkan arsitektur terbaik, ubah learning rate untuk melihat kecepatan dan kestabilan proses belajar atau training sehingga akan mendapatkan nilai terbaik dan arsitektur model terbaik.

Tabel 3.1.4. Data Pengamatan Perubahan Learning Rate

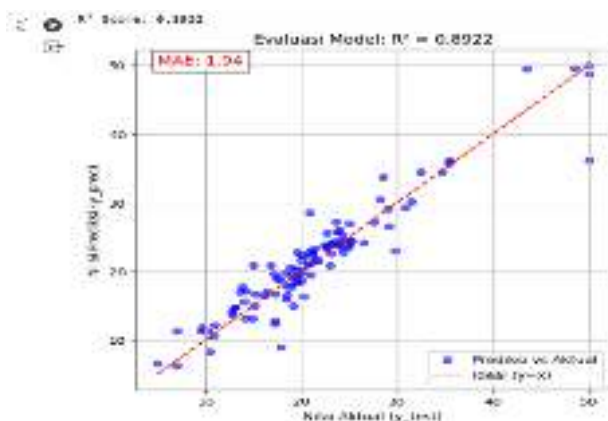
No	Learning Rate	Akurasi (R^2)	Error (MAE) %
1	0.01	0.8720 / 87.20 %	1.96
2	0.05	0.8742 / 87.42 %	1.96
3	0.001	0.8922 / 89.22 %	1.94
4	0.005	0.8782 / 87.82 %	2.07



Gambar 3.1.4. Evaluasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.2 Hasil Pengamatan Praktikum Tugas Klasifikasi

Hasil pengamatan praktikum tugas klasifikasi ini untuk mengetahui model klasifikasi dalam pengelompokan tiga jenis bunga (setosa, versicolor, virginica) dengan analisis data dari dataset Iris.csv. Proses yang dilakukan eksplorasi data, scaling data dan mencari arsitektur model terbaik dengan menambah jumlah hidden layer, mengubah jumlah neuron, activation function, learning rate, dan output hidden layer. Hasil model dianalisis menggunakan metrik konfusi yang dapat diamati dari training-validation dan akurasi model. Implementasi dari klasifikasi dapat dikembangkan untuk mendeteksi suara tembakan dengan mengumpulkan dataset dari berbagai suara tembakan sehingga dapat dikelompokkan untuk diidentifikasi jenis senjata apa yang mengeluarkan tembakan (Asmi et al., 2020).

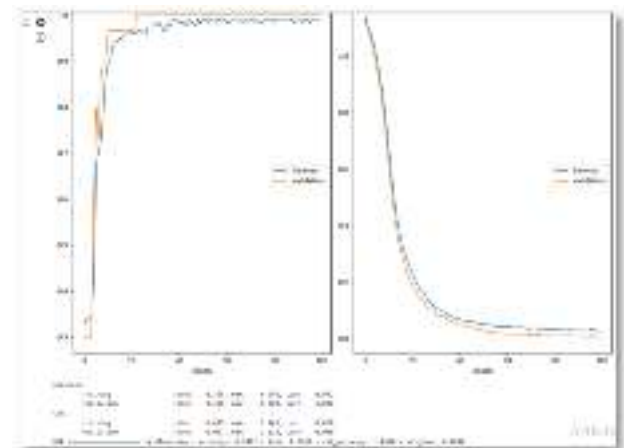
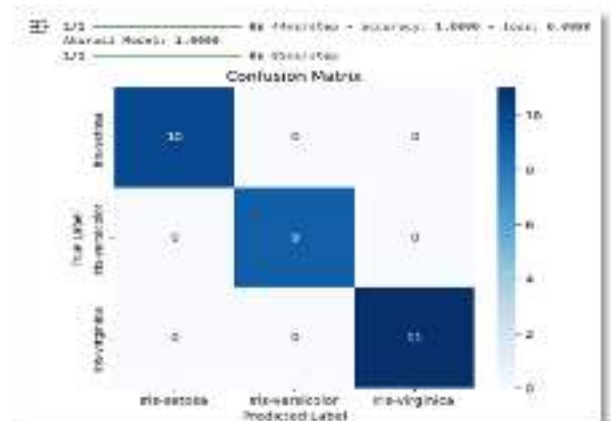


3.2.1 Menambah Hidden Layer

Mengubah desain arsitektur dari ANN dengan menambah hidden layer sampai 6 step-by-step untuk melihat fenomena dari setiap perubahan dengan jumlah neuron yang tetap (16).

Tabel 3.2.1. Data Pengamatan Jumlah Hidden Layer

Output Hidden Layer			
No	Jumlah Hidden Layer	Training-Validation	Akurasi Model Confusion Matrix
1	1	0.9340 / 93.40 %	0.9667 / 96.67 %
2	2	0.9729 / 97.29 %	1.0000 / 100 %
3	3	0.9900 / 99.00 %	1.0000 / 100 %
4	4	0.9972 / 99.72 %	1.0000 / 100 %
5	5	0.9929 / 99.29 %	1.0000 / 100 %
6	6	0.9932 / 99.32 %	1.0000 / 100 %



Gambar 3.2.1. Akurasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.2.2 Mengubah Jumlah Neuron

Mengubah jumlah neuron dengan 4 hidden layer secara random sehingga menemukan nilai yang terbaik. Jumlah neuron dapat menggunakan parameter nilai dari Biner kelipatan dari bit seperti 1,2,4,8,16,32 dst lebih baik dalam training disesuaikan dengan kompleksitas data.

Tabel 3.2.2. Data Pengamatan Perubahan
Jumlah Neuron

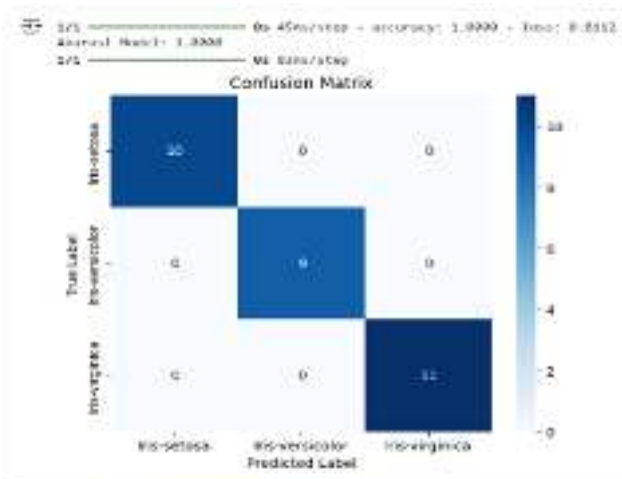
No	Jumlah Hidden Layer						Training-Validation	Akurasi Model Confusion Matrix
	1	2	3	4	5	6		
	Jumlah Neuron							
1	16	3 2	6 4	12 8	-	-	0.9918 / 99.18 %	1.0000
2	12 8	6 4	3 2	16	-	-	1.0000 / 100 %	1.0000
3	32	1 6	8	4	-	-	0.9877 / 98.77 %	1.0000
4	8	1 6	3 2	64	-	-	0.9953 / 99.53 %	1.0000
5	2	4	8	16	-	-	0.9598 / 95.98 %	0.9667

3.2.3 Mengubah Activation Function

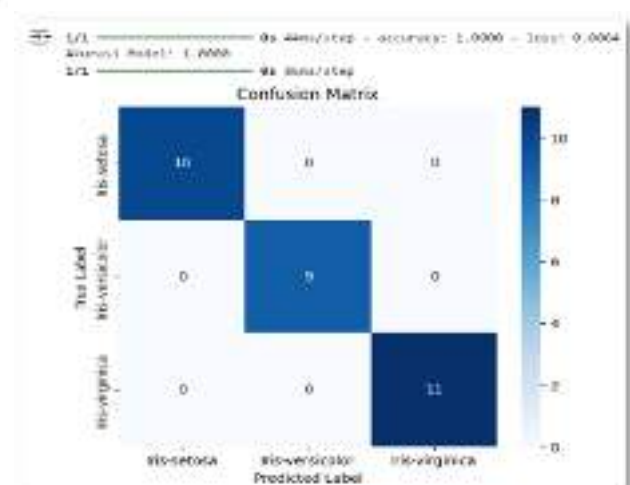
Mengubah activation function dengan relu, sigmoid, dan tanh untuk mengetahui penggunaan arsitektur yang tepat berdasarkan kompleksitas data sehingga didapatkan nilai terbaik.

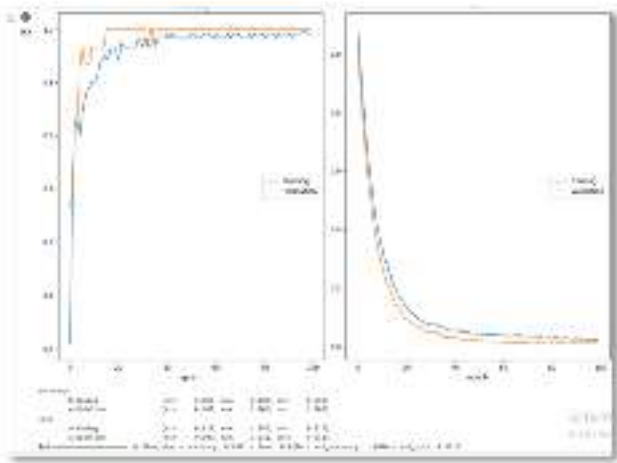
Tabel 3.2.3. Data Pengamatan Perubahan
Activation Function

No	Jumlah Hidden Layer						Training-Validation	Akurasi Model Confusion Matrix
	1	2	3	4	5	6		
	Activation Funtion							
1	tanh	tanh	Tanh	Tanh	-	-	0.9682	1.0000
2	sigmoid	sigmoid	Sigmoid	Sigmoid	-	-	0.9798	0.9667
3	relu	tanh	Sigmoid	Relu	-	-	0.9981	1.0000
4	tanh	sigmoid	Relu	Tanh	-	-	0.9831	1.0000
5	sigmoid	relu	Tanh	Sigmoid	-	-	0.9665	1.0000

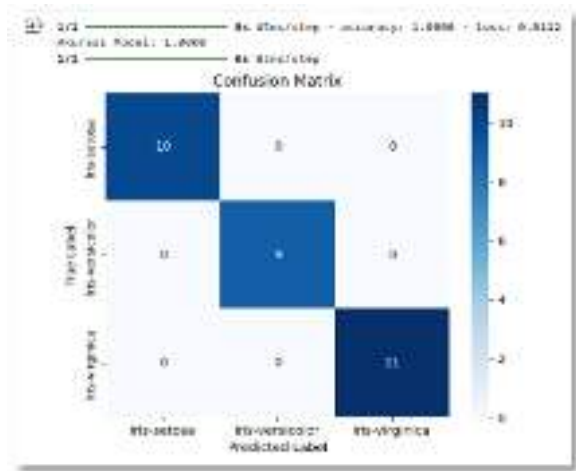


Gambar 3.2.2. Akurasi Model dan Grafik
Training-Validation (Hasil Terbaik)





Gambar 3.2.3. Akurasi Model dan Grafik Training-Validation (Hasil Terbaik)



Gambar 3.2.4. Akurasi Model dan Grafik Training-Validation (Hasil Terbaik)

3.2.4 Mengubah Learning Rate

Setelah mendapatkan arsitektur terbaik, ubah learning rate untuk melihat kecepatan dan kesetabilan proses belajar atau training sehingga akan mendapatkan nilai terbaik dan arsitektur model terbaik.

Tabel 3.2.4. Data Pengamatan Perubahan Learning Rate

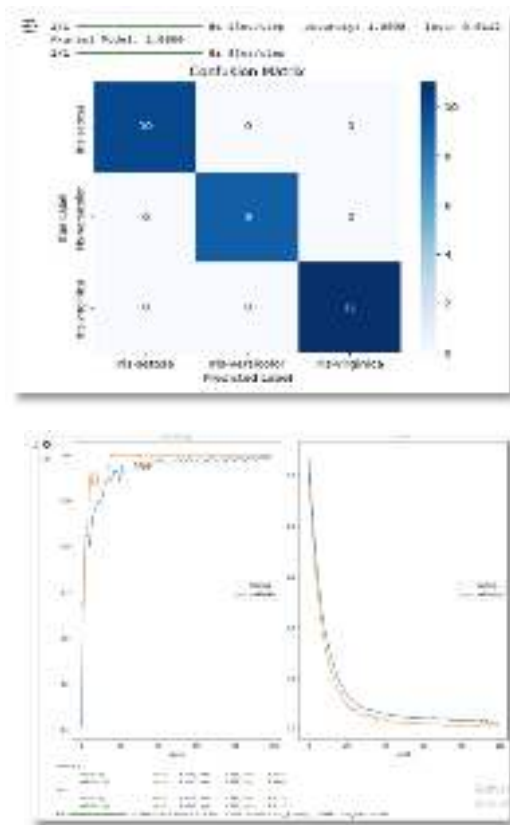
No	Learning Rate	Training_VValidation	Akurasi Model Confusion Matrix
1	0.01	0.9743 / 97.43 %	1.0000 / 100 %
2	0.05	0.9960 / 99.60 %	1.0000 / 100 %
3	0.001	0.9981 / 99.81 %	1.0000 / 100 %
4	0.005	0.9946 / 99.46 %	1.0000 / 100 %

3.2.5 Mengubah Activation Function Output

Setelah mendapatkan arsitektur terbaik, ubah activation function output dengan softmax, relu, dan sigmoid untuk melihat perbandingan hasil proses belajar atau training sehingga akan mendapatkan hasil terbaik dan arsitektur model terbaik serta prediksi yang akurat.

Tabel 3.2.5. Data Pengamatan Perubahan Activation Function Output

No	Activation Function Layer	Training-Validation	Akurasi Model Confusion Matrix
1	Softmax	0.9981 / 99.81 %	1.0000 / 100 %
2	Relu	0.3227 / 33.27 %	0.3667 / 36.67 %
3	Sigmoid	0.9928 / 99.28 %	1.0000 / 100 %



Gambar 3.2.5. Akurasi Model dan Grafik Training-Validation (Hasil Terbaik)

IV. KESIMPULAN

Penelitian ini membahas penerapan metode Deep Learning melalui desain arsitektur Jaringan Syaraf Tiruan (JST) untuk menyelesaikan dua jenis tugas utama, yaitu regresi dan klasifikasi. Tujuan utamanya adalah memahami bagaimana membangun arsitektur JST yang efektif dan menganalisis pengaruh variasi arsitektur terhadap kinerja model. Dataset **BostonHousing.csv** digunakan untuk regresi, sedangkan **Iris.csv** digunakan untuk klasifikasi.

Hasil praktikum menunjukkan bahwa JST merupakan metode yang fleksibel dan

efisien dalam menyelesaikan kedua jenis tugas. Untuk regresi, arsitektur JST dengan empat hidden layer, masing-masing berisi 256 neuron dan fungsi aktivasi ReLU, memberikan performa terbaik. Evaluasi menggunakan Mean Absolute Error (MAE) dan koefisien determinasi (R^2) menunjukkan hasil akurat, dengan R^2 mendekati 90%. Arsitektur terlalu sederhana cenderung mengalami underfitting, sementara arsitektur terlalu kompleks bisa menyebabkan overfitting, khususnya pada dataset kecil.

Dalam klasifikasi, JST mampu mengklasifikasikan tiga spesies bunga dalam dataset Iris dengan akurasi di atas 95%. Model dengan satu hingga dua hidden layer, fungsi aktivasi ReLU di hidden layer, dan softmax di output layer menunjukkan performa optimal. Precision dan recall yang seimbang antar kelas menunjukkan kemampuan generalisasi yang baik.

Kesimpulannya, variasi arsitektur JST, termasuk jumlah layer, neuron, fungsi aktivasi, dan algoritma optimasi, sangat memengaruhi performa model. Pemilihan parameter pelatihan, normalisasi data, dan fungsi loss yang sesuai juga berperan penting. JST terbukti andal dalam menyelesaikan tugas regresi dan klasifikasi bila arsitekturnya disesuaikan dengan kompleksitas dan karakteristik data.

DAFTAR PUSTAKA

- Akbar Nugraha, F., Ayu Fazira, H., Iqbal Nugraha, M., Febriansyah, A., Setiya Putra, Y. W., Adhim, M. F., Mubarak, H., Andri Nugraha Ramdhon, Fadly Febriya, ADITIYA, D. N., NUGRAHA, C., PRASSETIYO, H., Saputra, I. G., Ibrahim, I., Tanuwijaya, E., & Roseanne, A. (2022). Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN). *Jurnal Inovasi Teknologi Terapan*, 16(1), 191–198. <https://eproceeding.itenas.ac.id/index.php/fti/article/view/1718%0A>
<https://eproceeding.itenas.ac.id/index.php/fti/article/download/1718/1463>
- Aminullah, M. (2021). Perbandingan Performa Klasifikasi Machine Learning dengan Teknik Resampling pada Dataset Tidak Seimbang. In *Repository.Uinjkt.Ac.Id*. <https://repository.uinjkt.ac.id/dspace/bitstream/123456789/57648/1/MUHAMMAD AMINULLAH-FST.pdf>
- Andriani, A. Z. (2021). *PREDICTIVE MAINTENANCE (PdM) MENGGUNAKAN ACTIVE DAN SEMI- SUPERVISED MACHINE LEARNING PADA MESIN INDUSTRI*.
- Asmi, J., Saputra, J., & Kholid, F. (2020). Rancang bangun pendeteksi suara tembakan menggunakan sensor ultrasonik dan frekuensi radio pada pos pos satuan tugas daerah rawan. *Jurnal Elkasista*, 1–6. <http://journal.poltekad.ac.id/index.php/elka/article/view/70>
- Blessinda, K. (2024). *COMPUTING. December*.
- Efrian, M. R., & Latifa, U. (2022). Image Recognition Berbasis Convolutional Neural Network (Cnn) Untuk Mendeteksi Penyakit Kulit Pada Manusia. *Power Elektronik: Jurnal Orang Elektro*, 11(2), 276. <https://doi.org/10.30591/polektro.v12i1.3874>
- Elkasista, J., Mei, V., Sensor, P., Distance, L., Prodi, D., Elektronika, T., Senjata, S., Elektro, J. T., Anggrek, J. L. R., Telp, B., Elkasista, J., Mei, V., Sensor, P., & Distance, L. (2022). *Penerapan Sensor Light Distance and Ranging Pada Robot Berkaki 6 (Hexapod) Untuk Pemetaan Lokasi Berbasis Raspberry Pi 3 Application of Light Distance and Ranging Sensors on 6 Legged Robot (Hexapod) for Mapping Location Based on Raspberry Pi 3*. 3, 1–7.
- Elmuna, E. A. F. (2023). *Optimasi metode random forest menggunakan Principal Component Analysis untuk memprediksi harga rumah*. <http://etheses.uin->

- malang.ac.id/50422/
- Gunawan, M. A., Wiyanto, S., & Kholid, F. (2021). Penerapan Metode Back Propagation Pada Raspberry Pi 4 Untuk Mengenal Suara Tembakan Senjata Ringan Ss2- V1. *Jurnal Elkasista*, 2(Mei), 34–39. <https://doi.org/10.54317/elka.v2imei.158>
- Hayati, M. S. U., & Hadiprajitno, P. B. (2021). Departemen Akuntansi Fakultas Ekonomika dan Bisnis Universitas Diponegoro. *Diponegoro Journal of ...*, 1203011612(2022), 19840503. <https://ejournal3.undip.ac.id/index.php/accounting/article/view/32979%0Ahttps://ejournal3.undip.ac.id/index.php/accounting/article/download/32979/26344>
- Ismail, J., No, M., Cakranegara, K., Mataram, K., & Bar, T. (n.d.). *ANALISIS METODE KALMAN FILTER PADA GYROSCOPE UNTUK MENGURANGI NOISE GUNA MENINGKATKAN RESPONSIBILITAS ANALYSIS OF THE KALMAN FILTER METHOD ON GYROSCOPES TO REDUCE*. 5(2).
- Nwadiugwu, M. C. (2020). *Neural Networks, Artificial Intelligence and the Computational Brain*. <http://arxiv.org/abs/2101.08635>
- Pratama, Y., Rasywir, E., Fachruddin, F., Kisbianty, D., & Irawan, B. (2023). Eksperimen Layer Pooling menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN. *Building of Informatics, Technology and Science (BITS)*, 5(1), 200–210. <https://doi.org/10.47065/bits.v5i1.3604>
- Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, S., & Zaqiah, Q. Y. (2022). Deep Learning dan Penerapannya dalam Pembelajaran. *JlIP - Jurnal Ilmiah Ilmu Pendidikan*, 5(9), 3258–3267. <https://doi.org/10.54371/jljp.v5i9.805>
- Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4), 1875–1897. <https://doi.org/10.1214/19-AOS1875>
- Setiawan, S. I. A. (2011). Penerapan Jaringan Saraf Tiruan Metode Backpropagation Menggunakan VB 6. *Jurnal ULTIMATICS*, 3(2), 23–28. <https://doi.org/10.31937/ti.v3i2.301>
- Syahputro, P. A., Saputra, J., & Sridaryono, A. (2020). PADA RANCANG BANGUN SISTEM KONTROL ROBOT INTAI MENGGUNAKAN METODE ARTIFICIAL NEURAL NETWORK Metode Artificial Neural Network guna membantu kerja pasukan pengintai agar prosesor

data yang diterima dari Arduino Ide
merupakan software yang digunakan
untuk me. *Jurnal Elkasista*.

[http://journal.poltekad.ac.id/index.php
/elka/article/download/103/73](http://journal.poltekad.ac.id/index.php/elka/article/download/103/73)