

## Analisis Klasifikasi *Neural Network* Pada Deteksi Penyakit Liver

Putu Meidyka Ariani<sup>1)</sup>, Gede Angga Pradipta<sup>2)</sup>, Ni Kadek Sukerti<sup>3)</sup>

Teknologi Informasi<sup>1)</sup>, S2-Sistem Informasi<sup>2)</sup>, Sistem Informasi<sup>3)</sup>

Institut Teknologi dan Bisnis STIKOM Bali

Denpasar, Indonesia

e-mail: [210040197@stikom-bali.ac.id](mailto:210040197@stikom-bali.ac.id)<sup>1)</sup>, [angga\\_pradipta@stikom-bali.ac.id](mailto:angga_pradipta@stikom-bali.ac.id)<sup>2)</sup>,  
[nikadek\\_sukerti@stikom-bali.ac.id](mailto:nikadek_sukerti@stikom-bali.ac.id)<sup>3)</sup>

### Abstrak

Penyakit liver adalah salah satu penyebab kematian yang utama berbagai negara, penyakit ini dapat menyerang semua kelompok usia, mulai dari anak-anak hingga lanjut usia, baik di negara berkembang seperti Indonesia maupun di negara maju. Penderita penyakit liver sulit untuk dideteksi, terutama pada tahap awal penyakit. Banyaknya penelitian yang mengatakan bahwa kunci menangani penyakit liver adalah dengan mendeteksi secara dini. Pendekatan dengan *Machine Learning* digunakan untuk prediksi penyakit liver, dengan model klasifikasi *Neural Network* jenis *MultiLayer Perceptron (MLP)* untuk meningkatkan performa deteksi penyakit. Untuk meningkatkan optimasi hyperparameter pada model *MLP* dilakukan menggunakan *GridSearchCV*. Dataset yang digunakan dalam penelitian ini adalah *Indian Liver Patient Dataset*. Hasil penelitian menunjukkan bahwa model klasifikasi penyakit liver dengan *Neural Network* pada dataset ini mendapatkan nilai *accuracy* sebesar 70.29% dengan nilai *precision* 71.43%, nilai *recall* sebesar 100% dan nilai *f1-score* sebesar 83.33%. Kinerja yang baik juga tercermin dari hasil *confusion matrix*, yang menunjukkan bahwa model tidak melakukan kesalahan dalam memprediksi penderita penyakit hati sebagai bukan penderita.

**Kata kunci:** *Liver Disease, Neural Network, MultiLayer Perceptron, GridSearchCV, Machine Learning*

### 1. Pendahuluan

Hati (liver) adalah organ terbesar dalam tubuh manusia. Hati memiliki peran penting dalam berbagai proses, seperti penyimpanan energi, produksi protein dan asam empedu, pengaturan metabolisme kolesterol, serta detoksifikasi racun atau obat yang masuk ke dalam tubuh manusia [1]. *Global Cancer observatory (GLOBOCAN) 2020*, memberitahukan bahwa kanker hati merupakan salah satu penyakit utama kematian di Indonesia, dengan angka kematian mencapai 21.392 jiwa. Kanker hati dianggap sebagai salah satu penyakit dengan kematian tertinggi di dunia. Hannan (2010), sebagaimana dikutip oleh Rudianto (2016), menyatakan bahwa penyakit hati menjadi masalah kesehatan yang cukup serius di berbagai negara, penyakit ini dapat menyerang semua kelompok usia, mulai dari anak-anak hingga lanjut usia, baik di negara berkembang seperti Indonesia maupun di negara maju [2]. Berbagai masalah yang muncul akibat gangguan fungsi hati dapat memicu penyakit hati, seperti hepatitis, sirosis, liver, dan kanker hati.

Gejala awal kanker hati sulit untuk terdeteksi, sehingga banyak orang tidak menyadari bahwa mereka mengidap penyakit ini. Hal ini menyebabkan keterlambatan diagnosis dan penanganan. Berbagai penelitian menunjukkan bahwa deteksi dini adalah kunci untuk menangani penyakit liver secara efektif. Deteksi dini dapat dilakukan dengan mengamati ekspresi gen yang terdapat dalam DNA. Diagnosis penyakit liver dapat dilakukan dengan menggunakan proses segmentasi. Namun, segmentasi manual membutuhkan tenaga ahli khusus dan waktu yang lama [3].

Deteksi penyakit liver perlu dilakukan segera agar kondisinya tidak memburuk. Dalam banyak kasus, dokter sering menghadapi kesulitan dalam mengidentifikasi penyakit liver secara akurat karena faktor-faktor seperti gejala yang dapat tumpang tindih dengan penyakit lain [4]. Mengurangi risiko penyakit liver di masyarakat, diperlukan sistem yang mampu mendeteksi penyakit liver secara dini. Dengan perkembangan teknologi informasi, terutama di bidang kecerdasan buatan (*artificial intelligence*), khususnya pada cabang *machine learning* yang dimana sistem dapat belajar secara mandiri untuk mengambil keputusan tanpa perlu terus-menerus diprogram oleh manusia kini semakin berkembang, memungkinkan komputer menjadi cerdas melalui pembelajaran dari data yang ada [5]. Salah satu teknik yang digunakan adalah *data mining*, yang berfungsi untuk mengekstrak aturan dan pola dari berbagai kumpulan data. Tujuan utama *data mining* dalam diagnosis adalah untuk mendeteksi secara akurat, melakukan diagnosis

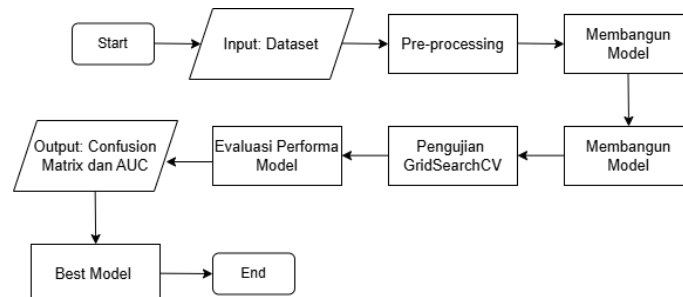
dini, dan mengidentifikasi faktor risiko. Hasil dari proses ini menjadi dasar penting dalam memilih metode pengobatan yang paling tepat [6].

Penelitian terkait tentang deteksi penyakit menggunakan model *Neural Network* sudah pernah dilakukan. Penelitian Irmawati, dkk (2022) tentang implementasi *Artificial Neural Network* untuk mendeteksi penyakit hati (liver) dengan python digunakan untuk mengetahui pengaruh variabel input berdasarkan data dalam literatur dan mendapatkan hasil akurasi sebesar 74% [7]. Kemudian terdapat penelitian David Galih dkk (2022) dimana pada penelitian berjudul “Klasifikasi Penyakit Jantung Menggunakan Metode *Artificial Neural Network*,” penelitian ini melibatkan pengukuran kinerja (akurasi, presisi, *recall*, dan *f-score*) metode ANN menggunakan 304 data pasien penyakit jantung yang diambil dari dataset pusat Kaggle. Hasil pengukuran kinerja menunjukkan akurasi sebesar 73,77%, presisi 80,43%, *recall* 84,09%, dan *f1-score* 82,22% [8]. Selain itu ada Bakhtiar Rifai (2013) dengan penelitian berjudul “Algoritma *Neural Network* Untuk Prediksi Penyakit Jantung” dengan menggunakan algoritma tersebut diperoleh nilai *accuracy* adalah 91.45 % dengan nilai *precision* 92.79 % dan nilai AUC adalah 0.937. Hal ini menyimpulkan bahwa *Neural Network* memberikan solusi yang lebih akurat dalam mendeteksi penyakit jantung [9].

Dari penelitian tersebut, menunjukkan bahwa algoritma *Neural Network* dapat terbukti efektif dalam melakukan klasifikasi dengan baik dan akurat. Namun, masih ada peluang untuk melakukan penelitian lebih lanjut untuk menentukan model *MultiLayer Perceptron* (MLP) yang optimal dengan mempertimbangkan pengaruh dari berbagai *hyperparameter*. Maka, pada penelitian ini akan membandingkan variasi nilai *hyperparameter* seperti *learning rate*, *optimizer*, *epoch* dan *batch size* menggunakan *GridSearchCV* pada dataset *Indian Liver Patient Dataset* (ILPD) by Kaggle.

## 2. Metode Penelitian

Pada metode penelitian terdapat pada gambar 1 di bawah ini menggambarkan alur yang merepresentasikan tahapan-tahapan penelitian yang akan dilaksanakan. Model klasifikasi dikembangkan dengan menggunakan bahasa pemrograman *Python*. Berikut ini adalah diagram yang menggambarkan tahapan penelitian yang telah dilakukan.



Gambar 1. Alur Penelitian

### 2.1 Dataset

Data dalam penelitian ini merupakan dataset pasien penyakit hati (*liver*) yang diperoleh dari *website Kaggle*, yaitu *Indian Liver Patient Dataset* (ILPD). Dataset ini berisi informasi yang dikumpulkan dari pasien di wilayah timur laut Andhra Pradesh, India. Dataset ini memiliki 11 atribut yakni *age*, *gender*, total bilirubin, direct bilirubin, alkphos alkaline phosphotase, SGPT alamine aminotransferase, SGOT aspartate aminotransferase, total proteins, albumin, albumin and globulin ration, dan *selector/label*. Di mana 10 atribut merupakan *feature*, sementara 1 atribut digunakan sebagai kelas atau *label* dan memiliki 583 *instance* atau isi data. Dari total 583 data, terdapat 416 pasien yang menderita penyakit liver dan 167 pasien yang tidak menderita penyakit liver. Berdasarkan distribusi jumlah pasien, dapat disimpulkan bahwa dataset ini memiliki ketidakseimbangan pada label kelas.

Beberapa atribut memiliki peran penting dalam menentukan apakah seorang pasien menderita penyakit liver. Atribut-atribut tersebut meliputi: 1) peningkatan kadar total bilirubin (TB), 2) peningkatan enzim SGPT (Alamine Aminotransferase), 3) peningkatan enzim SGOT (Aspartate Aminotransferase), dan 4) penurunan kadar albumin (ALB). Sistem deteksi dini penyakit liver ini diusulkan dengan menggunakan algoritma *Neural Network* berjenis *MultiLayer Perceptron* (MLP). Setiap algoritma akan diuji dengan *hyperparameter* yang berbeda untuk menemukan konfigurasi terbaik yang dapat menghasilkan performa yang optimal pada model *MultiLayer Perceptron* (MLP).

## 2.2 Neural Network MultiLayer Perceptron (MLP)

Jaringan saraf tiruan atau *Neural Network*, merupakan model buatan yang dirancang untuk merepresentasikan cara kerja otak manusia dan meniru proses pembelajarannya. Dalam *Neural Network*, struktur yang digunakan diilustrasikan menyerupai jaringan saraf pada otak manusia, di mana setiap *neuron* saling mengirimkan sinyal dan terhubung dalam sebuah jaringan [10]. Jaringan *neuron* terdiri dari tiga jenis lapisan utama, yaitu lapisan input (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan output (*output layer*), di mana setiap lapisan dilengkapi dengan matriks bobot  $W$  dan vektor bias  $b$  [11].

Penelitian ini menggunakan konfigurasi jaringan dengan dua *hidden layer* dan satu *output layer*. Setiap *hidden layer* terdiri dari 50 *neuron* dan menggunakan fungsi aktivasi ReLU untuk memperkenalkan non-linearitas. Pemilihan ReLU didasarkan pada kemampuannya mengatasi masalah *vanishing gradient* sekaligus meningkatkan kemampuan model dalam mempelajari pola yang kompleks. Pada *output layer*, digunakan fungsi aktivasi sigmoid karena model ini dirancang untuk tugas klasifikasi biner, sehingga menghasilkan *output* berupa probabilitas dengan rentang nilai antara 0 hingga 1. Selama proses *GridSearchCV*, model akan mengevaluasi berbagai kombinasi *hyperparameter* (termasuk jumlah struktur lapisan/arsitektur) untuk menentukan konfigurasi terbaik berdasarkan metrik evaluasi yang telah ditentukan.

## 2.3 GridSearchCV

*GridSearchCV* merupakan metode yang digunakan untuk menentukan kombinasi *hyperparameter* terbaik pada algoritma *machine learning* yang memiliki banyak parameter [12]. Dalam algoritma *Neural Network* jenis *MultiLayer Perceptron* (MLP), proses pelatihan model dilakukan dengan menggunakan *GridSearchCV* untuk mengoptimalkan berbagai *hyperparameter* yang tersedia [13]. Berikut ini adalah *hyperparameter* yang dilibatkan dalam proses *GridSearchCV* pada model MLP.

Tabel 1. Daftar *Hyperparameter* pada model MLP

Parameter	Value
Optimizer	Adam, SGD, RMSProp
Learning rate	0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; 0.8 ; 0.9 ; 1.0
Batch size	32, 64, 128
Epoch	1, 2, 4, 8, 16, 20

## 2.4 Pelatihan Model

Tahap pelatihan merupakan proses di mana *Neural Network* jenis *MultiLayer Perceptron* (MLP) dilatih menggunakan data yang telah melalui tahap *preprocessing* sebelumnya. Proses pelatihan dilakukan dengan menggunakan *GridSearchCV* untuk mengidentifikasi kombinasi *hyperparameter* terbaik, termasuk pengaturan *learning rate*, *batch size*, *epoch*, dan *optimizer*. Pelatihan *MultiLayer Perceptron* dijalankan berulang kali untuk setiap kombinasi *hyperparameter* yang diuji.

## 3. Hasil dan Pembahasan

Pada bagian hasil dan pembahasan, akan diinformasikan terkait proses-proses yang telah dilakukan untuk membandingkan *accuracy*, *precision*, *recall*, dan *f1-score* dengan berbeda optimizer. Adapun proses yang dilakukan, diantaranya:

### 3.1 Pre-Processing

Proses awal dalam tahap data *preprocessing* dimulai dengan menginput dataset sekunder pasien penyakit *liver* yang diperoleh dari website Kaggle, yaitu *Indian Liver Patient Dataset* (ILPD) ke dalam aplikasi *Visual Studio Code*. Dilakukan perubahan terhadap tipe data pada atribut *gender* yang semula bertipe kategorik diubah menjadi tipe numerik menggunakan teknik *Encoding*. Selanjutnya, penanganan *missing value* dilakukan dengan menghapus baris data yang mengandung nilai kosong, yang ditemukan pada atribut Albumin and Globulin Ratio. Setelah itu, data duplikat dihapus, di mana terdapat 13 baris duplikat dalam dataset. Langkah berikutnya adalah memisahkan data menjadi fitur dan label. Proses penyeimbangan data pada kedua kelas label dilakukan menggunakan metode *Synthetic Minority Oversampling Technique* (SMOTE). Setelah itu, data dibagi menjadi data *training* dan data *testing* dengan perbandingan 70% untuk *training* dan 30% untuk *testing*. Data kemudian diolah menggunakan model *MultiLayer Perceptron* (MLP). Proses pelatihan model dilakukan dengan bantuan metode *GridSearchCV*, yang digunakan untuk mengevaluasi berbagai kombinasi *hyperparameter* dan menentukan kombinasi yang menghasilkan performa terbaik.

### 3.2 Pengujian Parameter *GridSearchCV*

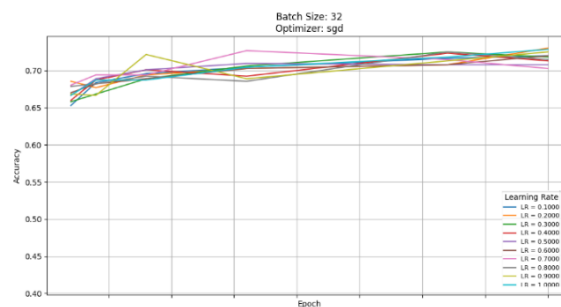
Pada tahap pengujian parameter *GridSearchCV*, dataset yang telah diproses memasuki proses klasifikasi menggunakan model *MultiLayer Perceptron* (MLP). Penyetelan *hyperparameter* dilakukan dengan mengatur parameter seperti *learning rate*, *batch size*, *epoch*, dan *optimizer*. Penyesuaian ini bertujuan untuk mencapai performa optimal yang memungkinkan model MLP memberikan hasil klasifikasi yang terbaik. Pada tabel 2 menunjukkan hasil *GridSearchCV* dengan *hyperparameter* optimal yang ditemukan untuk model MLP. Sementara itu, pada tabel 3 menunjukkan hasil *hyperparameter* terbaik yang ditemukan untuk model MLP sebagai berikut:

Tabel 2. Hasil *Hyperparameter* pada model MLP

Parameter	Value		
Optimizer	Adam	SGD	RMSProp
Learning rate	0.1	0.2	0.1
Batch size	64	32	32
Epoch	20	20	16

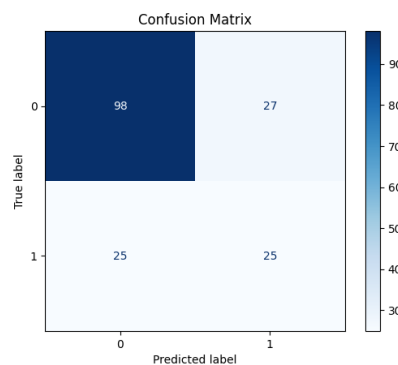
Tabel 3. *Hyperparameter* Terbaik pada model MLP

Parameter	Value
Optimizer	SGD
Learning rate	0.2
Batch size	32
Epoch	20



Gambar 2. Hasil model MLP pada *Optimizer* Terbaik

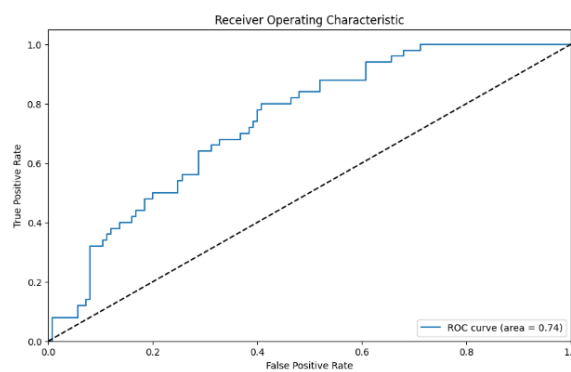
Hasil percobaan yang terdapat pada gambar 2 dan tabel 3 menunjukkan bahwa *hyperparameter* optimal yang diperoleh melalui *GridSearchCV* menghasilkan kombinasi *hyperparameter* yang sederhana namun efektif untuk model MLP. Penggunaan *optimizer* SGD (*Stochastic Gradient Descent*) dengan *learning rate* sebesar 0.2 menunjukkan bahwa model dapat melakukan penyesuaian bobot yang lebih cepat, tetapi masih cukup terkendali untuk menghindari *overshooting* selama proses pelatihan. Kinerja model dievaluasi berdasarkan efisiensi dan tingkat kesalahan menggunakan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Metrik evaluasi ini dihitung melalui *confusion matrix*. Berikut adalah *confusion matrix* yang dihasilkan oleh model MLP.



Gambar 3. *Confusion Matrix* dari model MLP

Matrix ini menampilkan jumlah prediksi yang benar dan salah dalam bentuk tabel, dengan membandingkan hasil prediksi model terhadap label sebenarnya. *Confusion matrix* ini terdiri dari komponent: True True Positives (TP): 25, yang menggambarkan model benar memprediksi penderita penyakit liver. False Positives (FP): 27, menunjukkan model salah memprediksi bukan penderita sebagai penderita penyakit liver. False Negatives (FN): 25, yaitu model salah memprediksi penderita penyakit liver sebagai bukan penderita. True Negatives (TN): 98, yang menunjukkan model benar memprediksi bukan penderita penyakit liver.

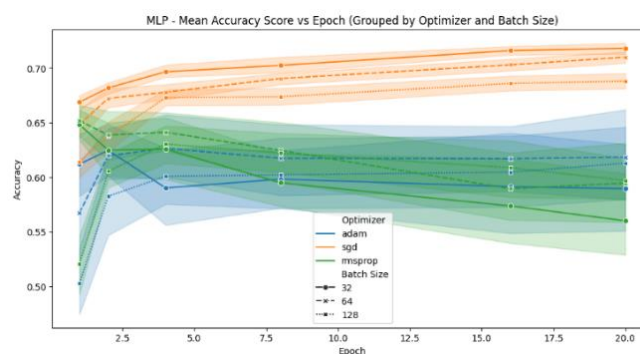
Evaluasi lanjutan menggunakan *classification report* menunjukkan bahwa model MLP terbaik memiliki performa yang sangat baik dalam mendeteksi penderita penyakit liver, model ini menghasilkan *accuracy* sebesar 70.29%, yang mengindikasikan bahwa model berhasil memprediksi dengan benar sekitar 70.29% dari semua sampel. *Precision* sebesar 71.43% mengindikasikan ketepatan model dalam memprediksi kelas positif, sementara *recall* 100% menunjukkan bahwa model mampu mendeteksi seluruh kasus positif tanpa kesalahan. *F1-score* 83.33% mengindikasikan keseimbangan yang baik antara *precision* dan *recall*, menjadikan model efektif dalam mendeteksi penderita penyakit liver.



Gambar 4. Kurva ROC

Visualisasi kurva ROC ini menggambarkan kemampuan model MLP dalam membedakan antara kelas positif dan negatif. *Area Under Curve* (AUC) sebesar 0.74 menunjukkan performa model yang cukup baik, meskipun belum optimal. Nilai AUC ini berarti model memiliki peluang 74% untuk membedakan antara kelas positif dan negatif secara benar.

### 3.2 Perbandingan Kinerja Optimizer



Gambar 5. Perbandingan *Accuracy* dengan *Epoch* berbeda *Optimizer*

Perbandingan antara *mean accuracy score* terhadap jumlah epoch untuk model MLP menunjukkan perbedaan yang signifikan berdasarkan jenis *optimizer* dengan *batch size* yang digunakan adalah 32, 64, dan 128. *Optimizer* adam cenderung membutuhkan lebih banyak epoch untuk mencapai akurasi yang stabil dibandingkan SGD. Namun pada *optimizer* SGD secara konsisten memiliki akurasi rata-rata yang lebih tinggi dibandingkan adam dan RMSProp di semua batch size. Sedangkan *optimizer* RMSProp memiliki akurasi rata-rata yang lebih rendah dibandingkan adam dan SGD.

#### 4. Kesimpulan

Dari uji eksperimen yang dilakukan pada model *Neural Network* jenis *MultiLayer Perceptron* (MLP) untuk deteksi pasien penyakit liver dengan menggunakan *GridSearchCV* telah berhasil dilakukan dengan kinerja yang sangat baik. Model MLP yang diuji untuk mendapatkan *accuracy*, *precision*, *recall*, dan *f1-score* sehingga model MLP menunjukkan performa terbaik didapatkan oleh parameter *optimizer* SGD, dengan *learning rate* 0.2, *batch size* 32, dan *epoch* 20 menghasilkan nilai *accuracy* sebesar 70.29% dengan nilai *precision* 71.43%, dengan nilai *recall* sebesar 100%, *F1-score* 83.33%, dan kurva ROC sebesar 0.74, pengujian model MLP menunjukkan bahwa model tersebut termasuk dalam kategori klasifikasi yang cukup baik. Untuk penelitian selanjutnya, diharapkan untuk meningkatkan arsitektur model. Dengan lebih banyak arsitektur model yang diuji dapat membantu meningkatkan performa. Penelitian juga dapat melibatkan metode *hyperparameter* tuning yang lebih canggih, seperti Bayesian Optimization atau *Genetic Algorithms*, untuk mendapatkan parameter terbaik.

#### Daftar Pustaka

- [1] F. Y. Zebua, S. H. Mulyani, and M. E. H, "Pemodelan Deteksi Penyakit Sirosis Hati dengan Menggunakan Jaringan Syaraf Tiruan," 2012, vol. 2, no. 2, pp. 99–117, 2012.
- [2] R. Rudianto, "Penentuan Penyakit Peradangan Hati Dengan Menggunakan Neural Network Backpropagation," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 1, no. 1, pp. 27–33, 2016, doi: 10.31294/ijcit.v1i1.416.
- [3] L. Geng, H. Che, Z. Xiao, and Y. Liu, "Extracting retinal anatomy and pathological structure using multiscale segmentation," *Appl. Sci.*, vol. 9, no. 18, pp. 1–15, 2019, doi: 10.3390/app9183669.
- [4] T. A. Assegie, "Support Vector Machine And K-Nearest Neighbor Based Liver Disease Classification Model," *Indones. J. Electron. Electromed. Eng. Med. informatics*, vol. 3, no. 1, pp. 9–14, 2021, doi: 10.35882/ijeemi.v3i1.2.
- [5] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, 2020, doi: 10.51211/biict.v7i2.1422.
- [6] N. Esfandiari, M. R. Babavalian, A. M. E. Moghadam, and V. K. Tabar, "knowledge discovery in medicine: Trend and issue," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4434–4463, 2014.
- [7] I. Irmawati, K. Widiyanto, F. Aziz, A. Rifai, and A. Rahmawati, "Implementasi Artificial Neural Network Dalam Mendeteksi Penyakit Hati (Liver)," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 6, no. 1, pp. 193–198, 2022, doi: 10.52362/jisamar.v6i1.694.
- [8] D. Pradana, M. Luthfi Alghifari, M. Farhan Juna, and D. Palaguna, "Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network," *Indones. J. Data Sci.*, vol. 3, no. 2, pp. 55–60, 2022, doi: 10.56705/ijodas.v3i2.35.
- [9] B. Rifai, "Algoritma Neural Network Untuk Prediksi," *Techno Nusa Mandiri*, vol. IX, no. 1, pp. 1–9, 2013.
- [10] I. Fauzan, "Artificial Intelligence (Ai) Pada Proses Pengawasan Dan Pengendalian Kepegawaian – Sebuah Eksplorasi Konsep Setelah Masa Pandemi Berakhir," *e-Journal BKN*, vol. 14, no. 1, pp. 31–42, 2020.
- [11] A. S. Shirazi and I. Frigaard, "Slurynet: Predicting critical velocities and frictional pressure drops in oilfield suspension flows," *Energies*, vol. 14, no. 5, Mar. 2021, doi: 10.3390/en14051263.
- [12] M. I. Gunawan, D. Sugiarto, and I. Mardianto, "Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Search pada Algoritma Logistic Regression," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, p. 280, 2020, doi: 10.26418/jp.v6i3.40718.
- [13] M. M. RAMADHAN, I. S. SITANGGANG, F. R. NASUTION, and A. GHIFARI, "Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency," *DEStech Trans. Comput. Sci. Eng.*, no. cece, 2017, doi: 10.12783/dtsc/cece2017/14611.