

Analisis Perbandingan Kinerja Algoritma Multilayer Perceptron dan K-Nearest Neighbor pada Klasifikasi Tipe Migrain

Comparative Analysis of Multilayer Perceptron and K-Nearest Neighbor Algorithms in Migraine Type Classification

Yang Agita Rindri¹, Agus Fitriyani²

Program Studi Teknologi Rekayasa Perangkat Lunak, Politeknik Manufaktur Negeri Bangka Belitung¹²

yang.agita@polman-babel.ac.id*¹, 1061904@polman-babel.ac.id²

Abstrak

Migrain merupakan sakit kepala yang biasanya terjadi pada salah satu sisi kepala saja atau dapat disebut sakit kepala sebelah. Migrain dapat terjadi pada siapa saja dengan berbagai gejala yang menandakan tipe migrain yang berbeda. Banyaknya tipe migrain yang diiringi dengan gejala-gejala yang berbeda membuat diagnosis dan perawatan terhadap penderita migrain menjadi sulit dilakukan. Penelitian ini bertujuan untuk membandingkan performa Algoritma *Multilayer Perceptron* (MLP) dan *K-Nearest Neighbor* pada Klasifikasi Tipe Migrain. Algoritma *Multilayer Perceptron* (MLP) dan *K-Nearest Neighbor* untuk klasifikasi migrain diharapkan dapat membantu dalam mengklasifikasikan jenis migrain berdasarkan gejala-gejala yang muncul secara tepat dan akurat. Klasifikasi yang tepat dan akurat dapat membantu tenaga kesehatan dalam merekomendasikan perawatan yang tepat pula bagi penderita migrain sesuai dengan jenis migrain yang dialaminya. Adapun *dataset* yang digunakan merupakan *dataset* yang diakses secara publik dan diambil situs kaggle.com. *Dataset* ini memiliki 24 atribut dan 400 baris data. Hasil penelitian menunjukkan bahwa Algoritma *Multilayer Perceptron* (MLP) menghasilkan prediksi yang cukup baik dengan tingkat akurasi 91%. Sedangkan Algoritma *K-Nearest Neighbor* menghasilkan prediksi dengan tingkat akurasi sebesar 72%.

Kata kunci: Migrain; MLP; *K-Nearest Neighbor*; Klasifikasi; Akurasi.

Abstract

Migraine is a headache that usually occurs on one side of the head or can be called a one-sided headache. Migraines can happen to anyone with a variety of symptoms that indicate different types of migraines. The many types of migraine accompanied by different symptoms make it difficult to diagnose and treat migraine sufferers. This study aims to compare the performance of the Multilayer Perceptron (MLP) Algorithm and K-Nearest Neighbor in Classification of Migraine Types. The Multilayer Perceptron (MLP) and K-Nearest Neighbor algorithms for migraine classification are expected to help classify migraine types based on the symptoms that appear precisely and accurately. Precise and accurate classification can help health workers in recommending the right treatment for migraine sufferers according to the type of migraine they are experiencing. The dataset used is a dataset that is publicly accessible and taken from the kaggle.com site. This dataset has 24 attributes and 400 rows of data. The Multilayer Perceptron (MLP) algorithm produces quite good predictions with an accuracy rate of 91%. Meanwhile, the K-Nearest Neighbor Algorithm produces predictions with an accuracy rate of 72%.

Keywords: Migraine; MLP; K-Nearest Neighbor; Classification; Accuracy.

Naskah diterima 18 Januari 2023; direvisi 26 Februari 2023; dipublikasi 1 Maret 2023.
JATI is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



1. Pendahuluan

Kecerdasan buatan merupakan salah satu cabang ilmu pengetahuan yang membantu manusia untuk menyelesaikan permasalahan kompleks di dunia nyata [1]. Salah satu pendekatan dari kecerdasan buatan adalah *machine learning* [2]. *Machine learning* terdiri dari algoritma-algoritma komputer yang dapat meningkatkan kinerja program komputer berdasarkan kumpulan data yang sudah ada. *Machine Learning* terdiri dari tiga jenis pembelajaran, yaitu 1) *Supervised Learning*; 2) *Unsupervised Learning*; dan 3) *Reinforcement Learning* [1],[2],[3]. *Supervised learning* banyak digunakan untuk memprediksi dan mengklasifikasikan objek atau pola berdasarkan label atau kategori yang diharapkan [4], sementara *unsupervised learning* mendeteksi pola pada data berdasarkan kemiripan atributnya dan tidak memiliki label pada outputnya [2],[5].

Machine learning sudah digunakan secara luas di berbagai bidang seperti bidang Kesehatan [6],[7], pertanian, *e-commerce*, transportasi, *game*, dan untuk melakukan aktifitas klasifikasi [6],[8], klusterisasi, prediksi [9], dan rekomendasi. Beberapa penerapan *machine learning* antara lain untuk mendiagnosa suatu penyakit, mengklasifikasikan topik berita, mengklasifikasikan jenis kayu, memprediksi harapan hidup

penderita penyakit radio vascular, dan masih banyak lagi. Salah satu algoritma *Supervised Learning* pada *Machine Learning* yang dapat digunakan untuk melakukan klasifikasi adalah algoritma *k-Nearest Neighbor* (k-NN) [10], [11] dan *Artificial Neural Network* (ANN) [12]. Algoritma ini telah banyak digunakan pada penelitian-penelitian terdahulu untuk kasus klasifikasi [12], [13]. Sedangkan pada *Unsupervised Learning*, salah satu algoritma yang digunakan adalah *K-Means Clustering* [14].

Di bidang kesehatan, *machine learning* salah satunya berguna untuk melakukan klasifikasi dan prediksi terhadap data yang ada dan menemukan golongan atau tipe baru dari suatu penyakit. Salah satu penyakit atau gejala yang membutuhkan klasifikasi adalah migrain atau sakit kepala sebelah. Migrain memiliki banyak gejala dan terbagi menjadi beberapa tipe [15], namun untuk menentukan tipe migrain yang diderita oleh pasien sangat sulit dilakukan dengan benar berdasarkan ciri-ciri dan gejala yang ada. Menentukan tipe migrain berdasarkan gejala yang ada dapat diselesaikan dengan mengklasifikasikan gejala-gejala yang muncul pada pasien berdasarkan tipe migrain yang ada di dunia kedokteran.

Migrain, sederhananya, merupakan sakit kepala yang biasanya terjadi pada salah satu sisi kepala saja atau dapat disebut sakit kepala sebelah. Migrain dapat terjadi pada siapa saja dengan berbagai gejala yang menandakan tipe migrain yang berbeda. Secara medis, migrain dikelompokkan menjadi 2 jenis utama migrain, yaitu migrain dengan aura dan migrain tanpa aura. Aura pada penyakit migrain merupakan suatu gejala akibat gangguan pada sistem saraf yang terjadi sesaat sebelum atau bersamaan dengan munculnya sakit kepala dan gejala migrain lainnya. Pada kasus yang lebih spesifik, jenis migrain dapat dibedakan menjadi lebih spesifik, yaitu *Basilar-type aura*, *Familial hemiplegic migraine* (FHM), migrain dengan aura, *Sporadic hemiplegic migraine* (SHM), *typical aura with migrain*, *Typical aura without migrain*, dan tipe migrain lainnya [16]. Jenis migrain hemiplegia (*hemiplegic migraine*) termasuk jenis migrain yang merupakan kelainan langka di mana penderita mengalami sakit kepala migrain bersamaan dengan kelemahan pada satu sisi tubuh (hemiplegia).

Migrain dengan aura merupakan migrain klasik dengan kondisi yang ditandai oleh gangguan pada penglihatan, kesemutan, dan kaku pada bagian leher. Migrain tanpa aura adalah sakit kepala yang tidak memiliki gejala-gejala khusus seperti pada migrain dengan aura. Sedangkan *Typical aura without migrain* adalah saat penderita mengalami gejala-gejala migrain yang lain tetapi tidak merasakan sakit kepala [16]. Banyaknya tipe migrain yang diiringi dengan gejala-gejala yang berbeda membuat diagnosis dan perawatan terhadap penderita migrain menjadi sulit dilakukan.

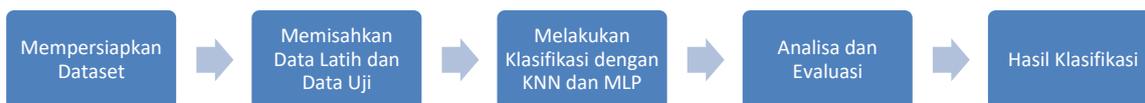
Penelitian mengenai klasifikasi migrain belum banyak dilakukan. Salah satu penelitian untuk mengklasifikasikan tipe migrain pernah dilakukan sebelumnya [15] di mana penelitian tersebut mengusulkan pendekatan *machine learning* untuk memanfaatkan *biomarker* berbasis potensi somatosensori yang ditimbulkan untuk klasifikasi migrain. *Dataset* pada penelitian [15] diambil melalui Sinyal SSEP yang ditimbulkan oleh stimulasi listrik saraf median dan direkam menggunakan perangkat CED 1401. Selanjutnya *dataset* tersebut dianalisa menggunakan algoritma *machine learning*. Sedangkan, penelitian yang penulis lakukan bertujuan untuk membandingkan performa Algoritma *Multilayer Perceptron* dan *K-Nearest Neighbor* pada Klasifikasi Tipe Migrain dengan menggunakan *dataset* numerik dalam bentuk atribut-atribut yang berkaitan dengan jenis migrain tertentu.

Algoritma *Multilayer Perceptron* dipilih dalam penelitian ini karena algoritma ini sangat banyak digunakan pada penelitian-penelitian di bidang kesehatan [17] dan sering menunjukkan performa terbaik dalam memecahkan kasus klasifikasi [18]. Sedangkan, algoritma *K-Nearest Neighbor* pada beberapa penelitian yang lain menunjukkan algoritma tersebut dapat mengklasifikasi dengan baik dan dapat mempelajari tugas-tugas kompleks menggunakan prosedur sederhana [18]. Selain itu, penulis belum menemukan penelitian yang menggunakan *dataset* migrain ini yang terpublikasi secara resmi di jurnal penelitian tertentu, terutama untuk penelitian berkaitan dengan analisis komparatif algoritma MLP dan KNN dengan *dataset* migrain.

Algoritma *Multilayer Perceptron* dan *K-Nearest Neighbor* untuk klasifikasi migrain diharapkan dapat membantu dalam mengklasifikasikan jenis migrain berdasarkan gejala-gejala yang muncul secara tepat dan akurat. Klasifikasi yang tepat dan akurat dapat membantu tenaga kesehatan dalam merekomendasikan perawatan yang tepat pula bagi penderita migrain sesuai dengan jenis migrain yang dialaminya.

2. Metode Penelitian

Untuk membandingkan kinerja Algoritma *Multilayer Perceptron* dengan Algoritma *K-Nearest Neighbor*, penulis melakukan langkah-langkah penelitian seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1 Mempersiapkan *Dataset*

Dari Gambar 1, Penelitian klasifikasi migrain ini dilakukan dengan menggunakan *dataset* publik yang diambil situs Kaggle yang dapat diakses pada <https://www.kaggle.com>. *Dataset* ini memiliki 24 atribut dan 400 baris data. Pada *dataset* yang digunakan, jenis migrain terdiri dari 7 jenis migrain, yaitu *Typical Aura with migraine*, *Typical Aura without migraine*, *Migraine without Aura*, *Basilar Type aura*, *Sporadic hemiplegic*, *Familial Hemiplegic Migrain*, dan jenis migrain lainnya. Adapun 24 atribut pada *dataset* ini ditampilkan pada Tabel 1.

Tabel 1. Atribut pada *Dataset Migrain*

No	Nama Atribut	Keterangan
1	Usia	Usia penderita migrain
2	Durasi	Durasi migrain terjadi dibagi menjadi 3 durasi yang diberi nilai 1 hingga 3
3	Frekuensi	Frekuensi migrain terjadi dibagi menjadi 3 durasi yang diberi nilai 1 hingga
4	Lokasi	Durasi migrain terjadi dibagi menjadi 3 durasi yang diberi nilai 1 hingga
5	Karakter	Karakter dari rasa sakit yang dirasakan dibagi menjadi 3 karakter yang diberi nilai dari 0 hingga 2
6	Intensitas	Intensitas migrain terjadi dibagi menjadi 4 intensitas yang diberi nilai dari 0 hingga 3
7	<i>Nausea</i>	<i>Nausea</i> atau gejala perasaan tidak nyaman pada bagian belakang tenggorokan atau lambung, jika iya maka diberi nilai 1, jika tidak maka diberi nilai 0
8	Muntah	Muntah yang terjadi, jika iya diberi nilai 1, jika tidak diberi nilai 0
9	<i>Phonophobia</i>	<i>Phonophobia</i> atau ketakutan terhadap suara, jika iya diberi nilai 1, jika tidak diberi nilai 0
10	<i>Photophobia</i>	<i>Photophobia</i> atau kondisi dimana mata terasa sakit, jika iya diberi nilai 1, jika tidak diberi nilai 0
11	<i>Visual</i>	Kondisi penglihatan dibagi menjadi 5 kondisi yang diberi nilai 0 hingga 4
12	<i>Sensory</i>	Kondisi indra dibagi menjadi 3 kondisi yang diberi nilai 0 hingga 2
13	<i>Dysphasia</i>	<i>Dysphasia</i> atau tidak dapat berbicara, jika iya diberi nilai 1, jika tidak diberi nilai 0
14	<i>Dysarthria</i>	<i>Dysarthria</i> atau kondisi ketika otot yang digunakan untuk dapat berbicara menjadi lemah, jika iya diberi nilai 1, jika tidak diberi nilai
15	Vertigo	Vertigo atau pusing hingga terasa berputar, jika iya diberi nilai 1, jika tidak diberi nilai 0
16	<i>Tinnitus</i>	<i>Tinnitus</i> atau telinga berdenging, jika iya diberi nilai 1, jika tidak diberi nilai 0
17	<i>Hypoacusis</i>	<i>Hypoacusis</i> atau kehilangan pendengaran total, jika iya diberi nilai 1, jika tidak diberi nilai 0
18	<i>Diplopia</i>	<i>Diplopia</i> atau penglihatan yang melihat 1 objek terasa ganda, jika iya diberi nilai 1, jika tidak diberi nilai 0
19	<i>Defect</i>	<i>Defect</i> atau cacat, jika iya diberi nilai 1, jika tidak diberi nilai 0
20	<i>Ataxia</i>	<i>Ataxia</i> atau gangguan keseimbangan atau koordinasi, jika iya diberi nilai 1, jika tidak diberi nilai 0
21	<i>Conscience</i>	Kemampuan untuk membedakan Tindakan yang betul atau salah, jika iya diberi nilai 1, jika tidak diberi nilai 0
22	<i>Paresthesia</i>	Kesemutan, jika iya diberi nilai 1, jika tidak diberi nilai
23	DPF	<i>Decreased Palpebral Fisure</i> atau reduksi jarak antara pinggir mata atas dan bawah relatif terhadap kontralateral mata, jika iya diberi nilai 1, jika tidak diberi nilai 0
24	Type	Tipe dari migrain, terdiri dari 7 tipe, diisi sesuai jenis migrain

Tabel 1 menunjukkan atribut pada *dataset* yang mana setiap baris data akan memiliki atribut yang nilainya bersifat bilangan numerik positif. *Dataset* tersebut disimpan dalam bentuk *Comma Separated Values (CSV)*. Adapun atribut pada *dataset* dibagi menjadi 2 jenis, yaitu atribut kondisi dan atribut kelas. Pada Tabel

1, atribut kondisi adalah atribut dengan nomor 1 sampai 23. Sementara itu, atribut kelas adalah atribut dengan nomor 24, yaitu atribut *Type*.

2.2 Memisahkan Data Latih (Data *Training*) dan Data Uji (Data *Testing*)

Dataset yang berjumlah 400 baris akan dibagi menjadi data latih yang digunakan untuk *training* model sehingga model memiliki pengetahuan dalam mengenali pola dari data yang ada dan data uji untuk menguji hasil prediksi model setelah dilakukan *training*. Adapun pembagian data untuk data latih dan data uji ditampilkan pada Tabel 2.

Tabel 2. Pembagian Data Latih dan Data Uji

No.	Nama Algoritma	Persentase Data Latih	Jumlah Data Latih	Persentase Data Uji	Jumlah Data Uji
1	MLP	75%	300	25%	100
2	KNN	75%	300	25%	100

Tabel 2 menunjukkan bahwa persentase data latih dan data uji pada kedua algoritma diberikan nilai yang sama, yakni 75% data sebagai data latih dan 25% data sebagai data uji sehingga jumlah data latih sebanyak 300 data dan jumlah data uji sebanyak 100 data.

2.3 Melakukan Klasifikasi Menggunakan Algoritma MLP dan Algoritma KNN

Selanjutnya, dilakukan data *training* dan data *testing* dengan menggunakan Algoritma *Multilayer Perceptron* dan *K-Nearest Neighbor* dengan *library* Python, yaitu *MLP Classifier* dan *KNN Classifier* pada modul Scikit-learn atau Sklearn. *MLP Classifier* ini sendiri merupakan *library* untuk melakukan klasifikasi dengan Algoritma *Multilayer Perceptron*. *MLP Classifier* pada Scikit-learn menggunakan fungsi aktivasi ReLU dengan ukuran ukuran *hidden layer* hingga 100 *layer*.

Sedangkan *KNN Classifier* merupakan *library* untuk melakukan klasifikasi dengan *K-Nearest Neighbor* (KNN). Pada tahap ini, dilakukan *training* terhadap 300 data dengan menggunakan *MLP Classifier* dan *KNN Classifier*. Hasil *training* memberi pengetahuan terhadap model dalam mengklasifikasikan data ke dalam tipe-tipe migrain yang ada sehingga dapat digunakan untuk *testing* terhadap 100 data uji. Model selanjutnya diuji menggunakan 100 data *testing* untuk melihat seberapa baik dalam melakukan klasifikasi tipe migrain. Tahap *testing* menghasilkan hasil klasifikasi yang dari masing-masing algoritma yang selanjutnya dianalisis dan dievaluasi menggunakan *Confusion Matrix*.

2.4 Analisa dan Evaluasi Algoritma MLP dan Algoritma KNN

Algoritma MLP dan KNN merupakan salah satu algoritma *Supervised Learning* yang bertujuan untuk melakukan klasifikasi dan prediksi. Kinerja dari *Machine Learning* terutama algoritma *Supervised Learning* dianalisis dengan menggunakan *Confusion Matrix*. *Confusion Matrix* membandingkan antara hasil prediksi dari model dengan kondisi aktual dari data yang ada [19]. *Confusion matrix* ditunjukkan pada Gambar 2.

		Predicted Condition	
		Positive (PP)	Negative (PN)
Actual Condition	Total Population = P+N		
	Positive (P)	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

Gambar 2. *Confusion Matrix* pada *Supervised Learning*

Gambar 2 menunjukkan *confusion matrix* yang terdiri dari kondisi aktual dan kondisi prediksi. *Confusion matrix* memiliki 4 kondisi [20], yaitu

1. *True Positive* apabila hasil prediksi dari data uji dan kondisi aktual pada data latih sama-sama menunjukkan kondisi Iya benar (positif). Dalam kasus klasifikasi migrain dalam penelitian ini, *True Positive* terjadi jika model memprediksi suatu data X sebagai tipe migrain Y dan kondisi aktual menunjukkan memang benar (*True*) bahwa data X tersebut adalah tipe migrain Y.

2. *True Negative* apabila hasil prediksi dan kondisi aktual sama-sama menunjukkan kondisi tidak benar (negative). Dalam kasus klasifikasi migrain dalam penelitian ini, *True Negative* terjadi jika model memprediksi suatu data X sebagai bukan tipe migrain Y dan kondisi aktual menunjukkan memang benar (*True*) bahwa data X tersebut adalah bukan tipe migrain Y.
3. *False Positive* apabila model salah memprediksi dimana kondisi actualnya negative tapi diprediksi menjadi positif. Dalam kasus klasifikasi migrain ini, *False Positif* terjadi jika model memprediksi suatu data X sebagai tipe migrain Y dan hasil prediksi tersebut salah (*False*) karena kondisi aktual menunjukkan bahwa data X tersebut ternyata bukan tipe migrain Y.
4. *False Negative* apabila model salah memprediksi dimana kondisi actualnya positif tapi diprediksi menjadi negatif. Dalam kasus klasifikasi migrain ini, *False Negative* terjadi jika model memprediksi suatu data X sebagai bukan tipe migrain Y dan hasil prediksi tersebut salah (*False*) karena kondisi aktual menunjukkan bahwa data X tersebut ternyata adalah tipe migrain Y.

Hasil dari *confusion matrix* dapat menunjukkan informasi mengenai nilai *precision*, *recall*, akurasi dan *F1-Score*. *Precision* merupakan rasio antara hasil prediksi yang positif yang memang positif pada kondisi aktual terhadap keseluruhan hasil prediksi yang bernilai positif. Untuk menghitung *precision* digunakan persamaan (1)[21].

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Akurasi merupakan perbandingan antara prediksi model yang menghasilkan nilai yang benar, baik benar dalam memprediksi positif ataupun negatif terhadap keseluruhan data yang ada. Rumus menghitung akurasi ditunjukkan pada persamaan (2)[21].

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (2)$$

Sedangkan *recall* adalah rasio prediksi benar bernilai positif dibandingkan dengan keseluruhan data aktual benar bernilai positif. Rumus menghitung akurasi ditunjukkan pada persamaan (3)[21].

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-score merupakan skor rata-rata antara *recall* dan *precision*. *F1-score* digunakan apabila *dataset* memiliki jumlah *False Negatif* dan *False Positif* yang hampir sama. Rumus untuk menghitung *F1-score* ditunjukkan pada persamaan (4).

$$\frac{1}{F1-score} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right) \quad (4)$$

Kinerja algoritma yang baik dipilih dengan mempertimbangkan nilai *precision*, *recall*, skor akurasi, atau *F1 Skor*. Jika jumlah data yang bernilai *False Negatif* dan *False Positif* sangat mendekati, maka algoritma yang memiliki nilai akurasi tinggi dapat dipilih, sebaliknya jika jumlah data yang bernilai *False Negatif* dan *False Positif* tidak mendekati, maka dapat memilih algoritma dengan *F1-Score* yang tinggi. Jika lebih menganggap bahwa data dengan *False Positif* lebih baik terjadi daripada *False Negatif*, maka algoritma dengan *recall* yang tinggi dapat dipilih. Akan tetapi, jika lebih menginginkan terjadinya *True Positif* dan sangat tidak menginginkan terjadinya *False Positif*, maka dapat memilih algoritma dengan *precision* yang tinggi. Pada kasus Klasifikasi Migrain, kinerja Algoritma MLP dan KNN dibandingkan dari sisi nilai *precision*, *recall*, skor akurasi dan *F1-Score*.

2.5 Multilayer Perceptron (MLP)

Multilayer Perceptron merupakan salah satu jenis algoritma dari Jaringan Saraf Tiruan (JST) atau *Artificial Neural Network* (ANN). ANN mengadopsi mekanisme kerja jaringan syaraf pada otak manusia. ANN terdiri dari sejumlah neuron yang berperan sebagai prosesor sederhana yang diumpamakan seperti neuron pada otak. Antar neuron dihubungkan oleh tautan berbobot yang berfungsi meneruskan sinyal dari satu neuron ke neuron lainnya. ANN terdiri dari lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*).

MLP yang merupakan salah satu metode ANN dengan satu atau lebih lapisan dan memiliki paling sedikit 3 lapisan jaringan perceptron yang saling terhubung, yaitu 1 lapisan masukan (*input layer*), 1 lapisan

tersembunyi (*hidden layer*), dan 1 lapisan keluaran (*output layer*). Terdapat 3 tahap utama pada *training* MLP, yaitu 1) *forward pass* untuk meneruskan *neuron input* dengan mengalikan terhadap bobot dan menambahkan bias sehingga mendapatkan output yang menjadi input untuk lapisan selanjutnya, 2) *loss calculation* untuk menghitung kesalahan prediksi, dan 3) *backward pass* yang menggunakan *error* pada tahap *loss calculation* untuk mengoreksi bobot. *Forward pass* ditunjukkan pada persamaan (5).

$$\phi_h(\sum_{i=1}^n w_{ij}x_i + b_j) = z_j \quad (5)$$

Adapun tahapan algoritma MLP meliputi tahapan sebagai berikut:

1. Inisialisasi nilai awal seperti nilai bias dan bobot secara acak dengan *range* 0 sampai 1
2. Tahapan aktivasi untuk mendapatkan *output* dari neuron pada *layer* tersembunyi menggunakan persamaan (5)
3. Menggunakan fungsi aktivasi ReLu yang memungkinkan model untuk menyelesaikan masalah nonlinier dan membangkitkan *output*.
4. Menghitung nilai *error* antara hasil prediksi dan nilai actual.
5. Kembali ke *layer* sebelumnya untuk menghitung *error* pada *hidden layer*
6. Memperbaiki bobot menggunakan nilai *error*. Tahapan ini merupakan *training* bobot untuk mendapatkan *output* yang mendekati *desired output*, dan tahapan iterasi hingga mendapatkan bobot yang paling baik untuk mendapatkan *output* yang mendekati *desired output* [8].

Penelitian ini menggunakan 25% data uji dan 75% data latih untuk Algoritma MLP sehingga total data uji sebanyak 100 data dan total data latih sebanyak 300 data. Sementara itu, jumlah maksimal iterasi yang ditetapkan pada penelitian ini sebanyak 1000 iterasi.

2.6 K-Nearest Neighbor (KNN)

Algoritma K-NN merupakan jenis algoritma *supervised learning* yang mampu melakukan klasifikasi berdasarkan jarak antara sejumlah objek (titik *training*) terhadap suatu titik *query*. Objek yang jaraknya berdekatan akan diklasifikasikan sebagai 1 kelas yang sama. Klasifikasi objek dilakukan berdasarkan pada kedekatan nilai atribut pada data latih dan data uji. Dekat atau jauhnya objek dihitung berdasarkan *Euclidean Distance* yang direpresentasikan pada persamaan (6).

$$E(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} \quad (6)$$

Algoritma *K-Nearest Neighbor* memiliki tahapan sebagai berikut:

1. Menentukan jumlah tetangga terdekat yang ditetapkan dengan bilangan ganjil misalnya 1,3,5, atau 7. Jumlah tetangga terdekat dinotasikan sebagai parameter k. Jumlah tetangga (parameter k) digunakan sebagai batasan saat melakukan pengurutan hasil perhitungan *Euclidean Distance*.
2. Menghitung jarak antara data yang diujikan dengan semua data latih (data *testing*) menggunakan *Euclidean Distance* pada persamaan (6).
3. Setelah mendapatkan jarak pada langkah kedua, selanjutnya jarak-jarak tersebut diurutkan dan selanjutnya menetapkan tetangga terdekat berdasarkan jarak terkecil sebanyak parameter k yang telah ditetapkan pada tahap 1.
4. Memeriksa kelas dari tetangga terdekat pada tahap 3 sebelumnya, yang dalam hal ini adalah tipe migrain.
5. Gunakan mayoritas kelas tetangga terdekat yang diperoleh pada tahap 4 sebagai nilai prediksi terhadap data uji selanjutnya.

Pada K-NN, harus ditentukan jumlah neighbor (tetangga) sebagai data klasifikasi. Penelitian ini menggunakan 7 *neighbor* (tetangga) untuk mengklasifikasi objek pada *dataset* dengan 25% data uji dan 75% data latih sehingga total data uji sebanyak 100 data dan total data latih sebanyak 300 data.

3. Hasil dan Pembahasan

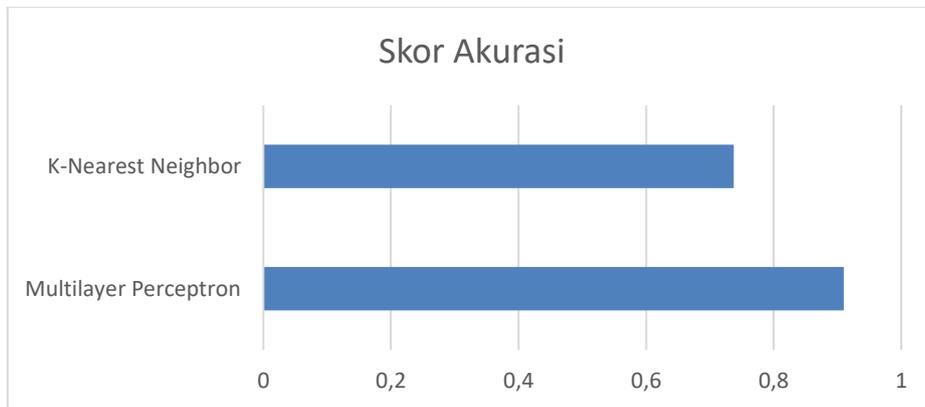
Dataset migrain yang terdiri dari 400 baris dalam format CSV digunakan sebagai data untuk memodelkan MLP dan KNN. Untuk membaca *dataset* di Python digunakan *library* Pandas yang merupakan *library* untuk memproses, memanipulasi, dan menganalisis data. *Dataset* migrain yang sudah diproses dengan *library* Pandas ditampilkan pada Gambar 3.

	Age	Duration	Frequency	Location	Character	Intensity	Nausea	Vomit	Phonophobia	Photophobia	...	Dysarthria	Vertigo	Tinnitus	Hypoacusis
0	30	1	5	1	1	2	1	0	1	1	...	0	0	0	0
1	50	3	5	1	1	3	1	1	1	1	...	0	1	0	0
2	53	2	1	1	1	2	1	1	1	1	...	0	0	0	0
3	45	3	5	1	1	3	1	0	1	1	...	0	1	0	0
4	53	1	1	1	1	2	1	0	1	1	...	0	0	0	0
...
395	39	1	1	1	1	3	1	1	1	1	...	0	1	0	0
396	20	3	2	1	1	3	1	0	1	1	...	0	0	1	1
397	28	3	2	1	1	3	1	0	1	1	...	0	1	0	0
398	20	3	1	1	1	3	1	0	1	1	...	0	1	0	0
399	37	1	1	1	1	3	1	1	1	1	...	0	1	0	0

400 rows × 23 columns

Gambar 3 Dataset Migrain

Selanjutnya dilakukan *training* terhadap 300 data latih dengan menggunakan MLP Classifier dan KNN Classifier. Hasil akurasi dari kedua algoritma ditampilkan pada Gambar 4.



Gambar 4 Hasil Skor Akurasi KNN dan MLP

Gambar 4 menunjukkan hasil skor akurasi Algoritma MLP sebesar 91%, sementara itu hasil skor akurasi Algoritma KNN sebesar 72%. Dengan demikian, hasil skor akurasi Algoritma MLP lebih tinggi daripada skor akurasi Algoritma KNN. Hasil *confusion matrix* Algoritma MLP menggunakan library *confusion_matrix* Python ditunjukkan pada Gambar 5.

```
cm = confusion_matrix(ytest,predictions)
cm
array([[ 5,  1,  0,  0,  0,  0,  0],
       [ 0,  3,  0,  0,  0,  2,  0],
       [ 0,  0, 16,  0,  0,  0,  0],
       [ 0,  0,  0,  6,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  2,  0],
       [ 0,  2,  0,  0,  2, 57,  0],
       [ 0,  0,  0,  0,  0,  0,  4]], dtype=int64)
```

Gambar 5 Confusion Matrix Algoritma MLP

Dari Gambar 5, *confusion matrix* masih dalam bentuk *array*. Apabila disajikan dalam bentuk tabel, maka *confusion matrix* Algoritma MLP ditampilkan pada Tabel 3.

Tabel 3. *Confusion Matrix* Algoritma MLP

Tipe Migrain	<i>Basilar-type aura</i>	<i>Familial hemiplegic migraine</i>	<i>Migraine without aura</i>	<i>Other</i>	<i>Familial hemiplegic migraine</i>	<i>Typical aura with migraine</i>	<i>Typical aura without migraine</i>	Total
<i>Basilar-type aura</i>	5	1	0	0	0	0	0	6
<i>Familial hemiplegic migraine</i>	0	3	0	0	0	2	0	5
<i>Migraine without aura</i>	0	0	16	0	0	0	0	16
<i>Other</i>	0	0	0	6	0	0	0	6
<i>Sporadic hemiplegic migraine</i>	0	0	0	0	0	2	0	2
<i>Typical aura with migraine</i>	0	2	0	0	2	57	0	61
<i>Typical aura without migraine</i>	0	0	0	0	0	0	4	4
Total	5	6	16	6	2	61	4	100

Tabel 3 menunjukkan jumlah data uji untuk tipe migrain *Basilar Type aura* sebanyak 6 data, *Familial Hemiplegic Migrain* sebanyak 5 data, *Migraine without Aura* sebanyak 16 data, jenis migrain lainnya sebanyak 6 data, *Sporadic hemiplegic* sebanyak 2 data, *Typical Aura with migraine* sebanyak 61 data uji, dan *Typical Aura without migraine* sebanyak 4 data sehingga total data uji sebesar 100 data. Adapun berdasarkan hasil klasifikasi pada Tabel 3, warna biru menyatakan bahwa MLP memprediksi secara tepat tipe migrain, sedangkan warna merah menunjukkan hasil prediksi MLP yang salah.

Dari Tabel 3 dapat dilihat bahwa model hanya bisa memprediksi dengan hasil Benar sebanyak 5 hasil, sementara itu terdapat 1 hasil prediksi yang tidak valid dari 6 hasil prediksi yang menyatakan tipe *Basilar-type aura*. Selain itu, terdapat 2 hasil yang tidak valid dari 5 hasil prediksi yang menyatakan tipe *Familial hemiplegic migraine*, 16 hasil valid dari 16 hasil prediksi yang menyatakan tipe *Migraine without aura*, 6 hasil valid dari 6 hasil prediksi yang menyatakan tipe *Other*, 2 hasil yang tidak valid dari 2 hasil prediksi yang menyatakan tipe *Sporadic hemiplegic migraine*, 4 hasil yang tidak valid dari 61 hasil prediksi yang menyatakan tipe *Typical aura with migraine*, dan 4 hasil valid dari 4 hasil prediksi yang menyatakan tipe *Typical aura without migraine*.

Berdasarkan Tabel 3, untuk mendapatkan nilai *precision*, pembagi menggunakan nilai total yang berada di kolom paling bawah Tabel 3, sedangkan untuk mendapatkan nilai *recall*, pembagi yang digunakan adalah nilai total yang berada di kolom paling kanan Tabel 3. Sedangkan nilai *precision*, *recall*, skor akurasi dan *F1-Score* untuk Algoritma MLP ditampilkan pada Gambar 6.

Gambar 6 menunjukkan nilai *precision*, *recall*, *F1-score*, dan rata-rata akurasi pada Algoritma MLP. Dari Gambar 6 dapat diketahui bahwa untuk tipe *Basilar-type aura*, *Typical aura without migrain*, dan jenis migrain lainnya, nilai *precision*, *recall*, *F1-score* sebesar 100% atau 1. Sedangkan untuk tipe migrain *Typical aura with migraine*, nilai *precision*, *recall*, dan *F1-score* masing-masing bernilai 0,93. Sedangkan tipe migrain *Sporadic hemiplegic migraine* mendapatkan nilai terburuk untuk *precision*, *recall*, *F1-score*, yakni sebesar 0.

Meskipun demikian, secara keseluruhan rata-rata perhitungan skor akurasi dengan Algoritma MLP sudah cukup baik, yaitu sebesar 0,91 yang artinya dari 100 data uji yang ada, terdapat 91 data yang diprediksi secara benar. Begitu juga dengan rata-rata nilai *precision*, *recall*, dan *F1-Score* dengan Algoritma MLP berturut-turut menunjukkan nilai yang baik, yaitu sebesar 0,92; 0,91; dan 0,91. Sementara itu, hasil *confusion matrix* Algoritma KNN menggunakan libari *confusion_matrix* Python ditunjukkan pada Gambar 7.

	precision	recall	f1-score	support
Basilar-type aura	1.00	0.83	0.91	6
Familial hemiplegic migraine	0.50	0.60	0.55	5
Migraine without aura	1.00	1.00	1.00	16
Other	1.00	1.00	1.00	6
Sporadic hemiplegic migraine	0.00	0.00	0.00	2
Typical aura with migraine	0.93	0.93	0.93	61
Typical aura without migraine	1.00	1.00	1.00	4
accuracy			0.91	100
macro avg	0.78	0.77	0.77	100
weighted avg	0.92	0.91	0.91	100

Gambar 6. Nilai *precision*, *Recall*, Akurasi dan *F1-Score* untuk Algoritma MLP

```
cm = confusion_matrix(y_test,y_pred)
cm
array([[ 0,  0,  0,  0,  0,  6,  0],
       [ 0,  1,  0,  0,  0,  4,  0],
       [ 0,  0,  9,  0,  0,  7,  0],
       [ 0,  1,  0,  0,  0,  5,  0],
       [ 0,  0,  0,  0,  0,  2,  0],
       [ 0,  0,  1,  0,  0, 60,  0],
       [ 0,  0,  0,  0,  0,  2,  2]], dtype=int64)
```

Gambar 7. *Confusion Matrix* KNN dalam Bentuk Array

Gambar 7 menunjukkan *Confusion Matrix* KNN dalam Bentuk Array dengan menggunakan fungsi *confusion_matrix* pada Python. *Confusion Matrix* ini apabila disajikan dalam bentuk tabel, maka *confusion matrix* Algoritma KNN ditampilkan pada Tabel 4.

Tabel 4 *Confusion Matrix* Algoritma KNN

Tipe Migrain	Basilar-type aura	Familial hemiplegic migraine	Migraine without aura	Other	Familial hemiplegic migraine	Typical aura with migraine	Typical aura without migraine	Total
Basilar-type aura	0	0	0	0	0	6	0	6
Familial hemiplegic migraine	0	1	0	0	0	4	0	5
Migraine without aura	0	0	9	0	0	7	0	16
Other	0	1	0	0	0	5	0	6
Sporadic hemiplegic migraine	0	0	0	0	0	2	0	2
Typical aura with migraine	0	0	1	0	0	60	0	61
Typical aura without migraine	0	0	0	0	0	2	2	4

Total	0	2	10	0	0	86	2	100
--------------	----------	----------	-----------	----------	----------	-----------	----------	------------

Dari Tabel 4, warna biru menyatakan jumlah prediksi tipe migrain yang dilakukan Algoritma KNN secara tepat, sedangkan warna merah menunjukkan hasil prediksi KNN yang salah. Tabel 4 menunjukkan bahwa KNN tidak dapat menghasilkan satu pun prediksi yang benar pada tipe migrain *Basilar-type aura*, *Sporadic hemiplegic migraine*, dan jenis lainnya (*other*). Sedangkan, pada tipe *Familial hemiplegic migraine*, KNN hanya mampu menghasilkan 1 prediksi yang bernilai benar dan 4 prediksi yang salah. Untuk tipe *Migraine without aura*, KNN mampu menghasilkan 9 prediksi yang benar dan 7 prediksi yang salah. Pada tipe *Typical aura with migraine*, KNN menghasilkan 60 prediksi yang benar dan 1 prediksi yang salah. Sementara itu, untuk tipe *Typical aura without migraine*, KNN menghasilkan 2 prediksi yang benar dan 2 prediksi yang salah.

Berdasarkan Tabel 4, untuk mendapatkan nilai *precision*, pembagi menggunakan total yang berada di kolom paling bawah Tabel 4, sedangkan untuk mendapatkan nilai *recall*, pembagi yang digunakan adalah nilai total yang berada di kolom paling kanan Tabel 4.

Dari *Confusion Matrix* pada Tabel 4, nilai *precision*, *recall*, skor akurasi dan *F1-Score* untuk Algoritma KNN ditampilkan pada Gambar 8.

	precision	recall	f1-score	support
Basilar-type aura	0.00	0.00	0.00	6
Familial hemiplegic migraine	0.50	0.20	0.29	5
Migraine without aura	0.90	0.56	0.69	16
Other	0.00	0.00	0.00	6
Sporadic hemiplegic migraine	0.00	0.00	0.00	2
Typical aura with migraine	0.70	0.98	0.82	61
Typical aura without migraine	1.00	0.50	0.67	4
accuracy			0.72	100
macro avg	0.44	0.32	0.35	100
weighted avg	0.63	0.72	0.65	100

Gambar 8. Nilai *Precision*, *Recall*, Akurasi dan *F1-Score* untuk Algoritma KNN

Gambar 8 menunjukkan nilai *precision*, *recall*, *F1-score*, dan rata-rata akurasi pada Algoritma KNN. Dari Gambar 8 dapat diketahui bahwa untuk tipe *Basilar-type aura*, *Sporadic hemiplegic migraine*, dan jenis migrain lainnya, mendapatkan nilai *precision*, *recall*, *F1-score* yang buruk, yakni sebesar 0. Sedangkan untuk tipe migrain *Typical aura with migraine*, nilai *precision*, *recall*, dan *F1-score* cukup baik, yaitu berturut-turut bernilai 1,00; 0,98; dan 0,82. Untuk tipe migrain *Familial hemiplegic migraine* dan *Migraine without aura* mendapatkan nilai yang tidak terlalu baik untuk *precision*, *recall*, *F1-score*, yakni berturut-turut sebesar 0,5; 0,2; dan 0,29. Adapun rata-rata perhitungan skor akurasi dengan Algoritma KNN sebesar 72% yang artinya dari 100 data uji yang ada, terdapat 72 data yang diprediksi secara benar. Adapun rata-rata nilai *precision*, *recall*, dan *F1-Score* berturut-turut menunjukkan nilai 63%, 72%, dan 65%.

4. Kesimpulan

Hasil penelitian dengan menggunakan *dataset* migrain yang tersedia secara publik dari *website* Kaggle menunjukkan klasifikasi tipe migrain dapat dilakukan dengan menggunakan Algoritma *Multilayer Perceptron* (MLP) dan Algoritma *K-Nearest Neighbor* (KNN). Algoritma *Multilayer Perceptron* (MLP) menghasilkan klasifikasi dan prediksi terhadap tipe migrain yang lebih baik daripada Algoritma *K-Nearest Neighbor* dengan skor akurasi sebesar 91%. Dari 100 data uji yang diberikan, MLP mampu mengklasifikasikan 91 data secara tepat sesuai dengan kondisi aktual yang ditampilkan pada *dataset* dan hanya keliru mengklasifikasikan 9 data. Sementara itu, Algoritma *K-Nearest Neighbor* menghasilkan prediksi dengan tingkat akurasi sebesar 72%. Ini artinya dari 100 data uji yang diberikan, KNN hanya mampu mengklasifikasikan 72 data secara tepat dan 28 data diklasifikasikan secara tidak tepat. Dari perbandingan nilai *precision*, *recall*, dan *F1-Score*, rata-rata nilai *precision*, *recall*, dan *F1-Score* pada Algoritma MLP berturut-turut menunjukkan nilai yang sangat baik, yaitu sebesar 92%, 91%, dan 91%. Sedangkan rata-rata nilai *precision*, *recall*, dan *F1-Score* pada KNN berturut-turut menunjukkan nilai 63%, 72%, dan 65%. Semakin baik nilai *F1-Score*, akurasi, *recall*, dan *precision* menunjukkan bahwa hasil

klasifikasi dan prediksi model algoritma semakin baik pula. Berdasarkan hasil dari nilai *precision*, *recall*, akurasi, dan *F1-Score* dapat disimpulkan bahwa Algoritma MLP memiliki kinerja yang lebih baik daripada Algoritma KNN dalam mengklasifikasikan tipe migrain. Dengan tingkat akurasi yang baik, Algoritma MLP dapat membantu dalam mengklasifikasikan jenis migrain berdasarkan gejala-gejala yang muncul secara akurat. Kontribusi dari hasil pemodelan yang dilakukan dengan Python pada penelitian ini dapat diekspor menggunakan *Library Pickle* pada Python untuk kemudian dikembangkan menjadi antarmuka pengguna, baik antarmuka berbasis *websie*, maupun berbasis *mobile*.

Daftar Pustaka

- [1] S. Russel and P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*, 3 rd. New Jersey: Prentice Hall, 2009. [Online]. Available: <https://zoo.cs.yale.edu/classes/cs470/materials/aima2010.pdf>
- [2] T. H. Sandhu, "Machine learning and natural language processing - a review," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, pp. 582–584, 2018, [Online]. Available: <http://www.ijarcs.info/index.php/Ijarcs/article/view/5799/4788>
- [3] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*. Cham, Switzerland: Springer, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-22475-2_1
- [4] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons.B*, vol. 4, no. December 2017, pp. 51–62, 2017, [Online]. Available: https://www.researchgate.net/publication/328146111_An_overview_of_the_supervised_machine_learning_methods
- [5] M. Batta, "Machine Learning Algorithms - A Review," *International Journal of Science and Research (IJSR)*, vol. 18, no. 8, pp. 381–386, 2018, [Online]. Available: <https://www.ijsr.net/archive/v9i1/ART20203995.pdf>
- [6] A. Puspitawuri, E. Santoso, and C. Dewi, "Diagnosis Tingkat Risiko Penyakit Stroke Menggunakan Metode K-Nearest Neighbor dan Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 4, pp. 3319–3324, 2019, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4916>
- [7] A. N. Liyantoko, I. Candradewi, and A. Harjoko, "Klasifikasi Sel Darah Putih dan Sel Limfoblas Menggunakan Metode Multilayer Perceptron Backpropagation," *Indonesian Journal of Electronics and Instrumentation Systems*, vol. 9, no. 2, p. 173, 2019, [Online]. Available: <https://jurnal.ugm.ac.id/ijeis/article/view/49943>
- [8] D. Wibowo, D. Erwanto, and D. Kusumastuti, "Klasifikasi Jenis Kayu Menggunakan Ekstrasi Fitur Gray Level Co-Occurrence Matrix dan Multilayer Perceptron," *Jurnal Nasional Teknik Elektro*, vol. 10, no. 1, 2021, [Online]. Available: <http://jnte.ft.unand.ac.id/index.php/jnte/article/view/788>
- [9] K. Khoirudin, D. Nurdiyah, and N. Wakhidah, "Prediksi Penerimaan Mahasiswa Baru Dengan Multi Layer Perceptron," *Jurnal Pengembangan Rekayasa dan Teknologi*, vol. 14, no. 1, p. 1, 2019, [Online]. Available: <https://journals.usm.ac.id/index.php/jprt/article/view/1212>
- [10] L. Wang, "Research and Implementation of Machine Learning Classifier Based on KNN," *IOP Conference Series: Materials Science and Engineering*, vol. 677, no. 5, pp. 0–5, 2019, [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/677/5/052038/pdf>
- [11] S. Wiyono and T. Abidin, "Comparative Study of Machine Learning Knn, Svm, and Decision Tree Algorithm To Predict Student'S Performance," *International Journal of Research - GRANTHAALAYAH*, vol. 7, no. 1, pp. 190–196, 2019, [Online]. Available: <http://dx.doi.org/10.29121/granthaalayah.v7.i1.2019.1048>
- [12] A. F. Hardiyanti and D. Fitriana, "Perbandingan Algoritma C4.5 dan Multilayer Perceptron untuk Klasifikasi Kelas Rumah Sakit di DKI Jakarta," *Jurnal Telekomunikasi dan Komputer*, vol. 11, no. 3, p. 198, 2021, [Online]. Available: <https://publikasi.mercubuana.ac.id/index.php/Incomtech/article/view/10632>
- [13] I. N. Purnama, "Perbandingan Klasifikasi Website Secara Otomatis Menggunakan Metode Multilayer Perceptron dan Naive Bayes," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 2, no. 2, pp. 155–161, 2021, [Online]. Available: <https://www.ejurnal.stmik-budidarma.ac.id/index.php/JSON/article/view/2703>
- [14] B. Harahap, "Penerapan Algoritma K-Means Untuk Menentukan Bahan Bangunan Laris (Studi Kasus Pada UD. Toko Bangunan YD Indarung)," *Regional Development Industry & Health Science, Technology and Art of Life*, pp. 394–403, 2019, [Online]. Available: <https://ptki.ac.id/jurnal/index.php/readystar/article/view/82>
- [15] B. Zhu, G. Coppola, and M. Shoaran, "Migraine classification using somatosensory evoked

- potentials,” *Cephalalgia International Journal of Headache*, vol. 39, no. 9, pp. 1143–1155, 2019, [Online]. Available: <https://bpb-us-w2.wpmucdn.com/sites.coecis.cornell.edu/dist/9/165/files/2019/05/03331024198399751.pdf>
- [16] D. W. Lewis, “Pediatric Migraine,” *Neurologic Clinics*, vol. 27, no. 2, pp. 481–501, May 2009, [Online]. Available: <http://dx.doi.org/10.1016/j.ncl.2008.11.003>
- [17] A. Setiadi, “Penerapan Algoritma Multilayer Perceptron untuk Deteksi Dini Penyakit Diabetes,” *Jurnal Komputer dan Informatika*, vol. 14, no. 1, pp. 46–59, 2012.
- [18] Dessy Kusumaningrum and E. M. Imah, “Studi Komparasi Algoritma Klasifikasi Mental Workload Berdasarkan Sinyal EEG,” *Jurnal Sistem Cerdas*, vol. 3, no. 2, pp. 133–143, 2020, doi: 10.37396/jsc.v3i2.69.
- [19] S. Sudianto, A. D. Sripamuji, I. Ramadhanti, R. R. Amalia, J. Saputra, and B. Prihatnowo, “Penerapan Algoritma Support Vector Machine dan Multi-Layer Perceptron pada Klasifikasi Topik Berita,” *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 11, no. 2, pp. 84–91, 2022, [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/janapati/article/view/44151>
- [20] A. Rahmawati, D. Wintana, and S. Suhada, “Klasifikasi Naïve Bayes Untuk Mendiagnosis Penyakit Pneumonia Pada Anak Balita (Studi Kasus : Uptd Puskesmas Sukaraja Sukabumi),” *Kumpulan Jurnal Ilmu Komputer (KLIK)*, vol. 06, no. 03, pp. 241–253, 2019, [Online]. Available: <http://klik.ulm.ac.id/index.php/klik/article/view/202>
- [21] C. Journal, I. W. Saputro, B. W. Sari, P. Studi, I. Komputer, and J. Informatika, “Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa,” *Creative Information Technology Journal (Citec Journal)*, vol. 6, no. 1, pp. 1–11, 2019, [Online]. Available: <https://citec.amikom.ac.id/main/index.php/citec/article/view/178>