

KOMPARASI ALGORITMA DECISION TREE DAN KNN DALAM MENGGKLASIFIKASI DAERAH BERDASARKAN PRODUKSI LISTRIK

Muhammad Alfathan Harriz¹, dan Harlis Setiyowati²

¹Magister Teknologi Informasi, Universitas Pradita, Indonesia

²Manajemen Retail, Universitas Pradita, Indonesia

Email: harrizsb@gmail.com¹, harlis.setiyowati@pradita.ac.id²

Abstrak

Pilihan negara yang sedang tumbuh dan berkembang, memiliki jumlah penduduk yang besar dan luas wilayah yang luas yaitu India sebagai contoh untuk diteliti. Listrik komponen vital dan berperan penting. Peneliti melakukan perbandingan akurasi antara dua algoritma pembelajaran mesin yang populer, yaitu Decision Tree dan KNN (K-Nearest Neighbor). Dataset yang berisi sampel sebanyak 345273 digunakan dan validasi dengan metode StratifiedKFold sebanyak 33 bagian dilakukan untuk mengevaluasi hasil klasifikasi dari kedua algoritma tersebut. Hasil dari penelitian ini menunjukkan bahwa algoritma Decision Tree yaitu 85.78% dan memiliki akurasi yang lebih tinggi dibandingkan dengan KNN yang memiliki akurasi sebesar 80.34% dalam mengklasifikasi daerah di India berdasarkan produksi listrik yang dihasilkan. Selain itu, temuan lainnya yaitu Decision Tree memiliki waktu komputasi yang lebih cepat yaitu 51.66 detik dibandingkan dengan KNN yang memiliki waktu komputasi sebesar 56.27 detik. Kesimpulan dari penelitian ini adalah bahwa algoritma Decision Tree memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma KNN. Selain itu, Decision Tree juga memiliki waktu komputasi yang lebih cepat. Dengan demikian, algoritma Decision Tree dapat menjadi pilihan yang lebih baik dalam melakukan klasifikasi.

Kata Kunci: Decision Tree, K-Nearest Neighbor, Klasifikasi.

Abstract

Growing and developing countries with a large population and a large area, such as India, are chosen as examples for study. Electricity is a vital and important component. Researchers have compared the accuracy of two popular machine learning algorithms, Decision Tree and KNN (K-Nearest Neighbor). A dataset containing 345273 samples and StratifiedKFold validation with 33 parts was carried out to evaluate the classification results of the two algorithms. The results of this study show that the Decision Tree algorithm is 85.78% and has higher accuracy than KNN, which has an accuracy of 80.34% in classifying areas in India based on the generated electricity. In addition, another finding is that the Decision Tree has a faster computation time of 51.66 seconds compared to KNN, which has a computation time of 56.27 seconds. This study concludes that the Decision Tree algorithm has higher accuracy than the KNN algorithm. In addition, the Decision Tree also has a faster computation time. Therefore, the Decision Tree algorithm can be a better choice for classification.

KeyWords: Decision Tree, K-Nearest Neighbor, Klasifikasi.

I. PENDAHULUAN

Listrik memegang peranan penting dalam masyarakat modern, dan produksi dan distribusinya merupakan komponen vital dari infrastruktur dan ekonomi suatu negara. Sebagai negara yang sedang tumbuh dan berkembang, India memiliki jumlah penduduk yang besar, yaitu sebesar 1,2 miliar orang, dan luas wilayah yang luas, yaitu sebesar 3,29 juta km², sehingga menjadikannya negara ke-7 terbesar di dunia [1]. Pada tahun 2020 akses listrik di India adalah 96.7% [2].

Sebagai contoh, sebagian besar penduduk di Chhattisgarh, India, yaitu sebanyak 86% rumah tangga memiliki akses ke listrik. Persentase ini lebih tinggi daripada rata-rata nasional di daerah pedesaan di India, yaitu 78%. Namun, masih terdapat sejumlah rumah tangga di daerah tersebut yang tidak memiliki akses ke listrik, sehingga masih diperlukan upaya untuk meningkatkan akses listrik secara keseluruhan di India [3]. Peningkatan akses listrik di India merupakan salah satu fokus utama pemerintah untuk meningkatkan kualitas hidup masyarakat.

Salah satu metode yang dapat digunakan untuk menyelesaikan masalah ini adalah dengan mengklasifikasi lokasi yang telah terjangkau berdasarkan wilayah produksi listrik. Dengan melakukan klasifikasi tersebut, pemerintah dapat mengidentifikasi wilayah mana yang belum terjangkau akses listrik secara, atau menentukan lokasi yang tidak terjangkau listrik sama sekali. Analisis ini dapat membantu pemerintah dalam mengambil keputusan strategis untuk meningkatkan akses listrik di berbagai lokasi di negara tersebut.

Proses klasifikasi merupakan teknik yang berguna untuk membagi atau mengategorikan objek mengacu pada karakter atau jenis tertentu, seperti makhluk hidup yang berusaha mengorganisir obyek yang satu dengan lainnya. Mesin pembelajaran dapat menghasilkan prediksi atau regresi dengan menggunakan data citra yang diproses berdasarkan data yang telah dipelajari dalam proses pembelajaran mesin. Klasifikasi merupakan salah satu aplikasi penting dari mesin pembelajaran, karena dapat digunakan untuk mengategorikan objek atau sampel ke dalam kelas yang sesuai berdasarkan ciri-ciri yang terkandung di dalamnya. Proses ini sering digunakan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks, di mana mesin

pembelajaran dapat membantu dalam mengelompokkan objek atau sampel ke dalam kelas yang sesuai berdasarkan ciri-ciri yang terkandung di dalamnya [4].

Klasifikasi merupakan salah satu aplikasi penting dari mesin pembelajaran, karena dapat digunakan untuk mengategorikan objek atau sampel ke dalam kelas yang sesuai berdasarkan ciri-ciri yang terkandung di dalamnya. Proses ini sering digunakan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks, di mana mesin pembelajaran dapat membantu dalam mengelompokkan objek atau sampel ke dalam kelas yang sesuai berdasarkan ciri-ciri yang terkandung di dalamnya. Mesin pembelajaran dapat menghasilkan prediksi atau regresi dengan menggunakan data citra yang diproses berdasarkan data yang telah dipelajari dalam proses training. Hal ini memungkinkan mesin pembelajaran untuk mengenali dan mengklasifikasi objek baru yang tidak terdapat dalam data training, sehingga memiliki aplikasi yang luas dalam bidang yang beragam.

Dalam penelitian ini, kami akan menggunakan mesin pembelajaran untuk mengklasifikasi daerah di India berdasarkan tingkat produksi listrik yang dihasilkan. Kami akan menggunakan dataset [5] yang berisi sampel sebanyak 345273 dan melakukan validasi dengan metode *StratifiedKfold* sebanyak 33 bagian untuk mengevaluasi hasil klasifikasi yang dihasilkan oleh algoritma yang dipilih. Kami akan membandingkan akurasi dan kecepatan komputasi antara dua algoritma pembelajaran mesin yang populer, yaitu *Decision Tree* dan KNN (*K-Nearest Neighbor*), dalam mengklasifikasi daerah di India tersebut.

II. METODE PENELITIAN

A. *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) adalah strategi yang menggolongkan kelas suatu objek berdasarkan kelas dari objek-objek lain yang terdekat dalam ruang fitur. Metode ini merupakan metode supervised learning yang membutuhkan data dalam proses pelatihan untuk mengklasifikasikan objek terdekat. Prinsip kerja KNN adalah dengan mencari jangkauan terpendek antara data yang akan dievaluasi dan tetangga dalam data latih [6]–[8]. Perhitungan *K-Nearest Neighbor* digambarkan sebagai berikut (1).

$$y = KNN(x, D, K, d_M) \quad (1)$$

Nilai y adalah kelas atau nilai yang akan diberikan pada data x . x adalah data yang akan diklasifikasikan atau diregresikan. D adalah dataset yang digunakan sebagai acuan dalam pembuatan keputusan. K adalah jumlah tetangga terdekat yang akan dijadikan sebagai acuan dan d_M (2) adalah *Minkowski distance* dengan nilai $p = 2$, yaitu:

$$d_M(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{\frac{1}{2}} \quad (2)$$

B. *Decision Tree*

Decision Tree adalah algoritma klasifikasi yang memproyeksikan keanggotaan objek ke dalam divisi atau kelas dengan meninjau atribut (variabel penentu). Algoritma ini dapat menghadapi sifat data yang *continuous* dan diskrit, mengeksekusi pembelajaran data pada *missing value*, dan memanfaatkan *gain ratio* bertujuan memperbaiki *information gain*. *Decision tree* juga mampu melakukan penyusunan aturan dalam bentuk pohon keputusan yang fleksibel tergantung pada data latih. Struktur pohon *Decision Tree* terdiri dari *node* yang menggambarkan karakteristik data, cabang yang menggambarkan nilai dari atribut, dan daun yang mendeskripsikan kelas [9]–[11].

Metode *decision tree* yang digunakan oleh scikit secara standar adalah *Classification and Regression Tree* (CART). Metode *Classification and Regression Tree* (CART) merupakan sebuah algoritma yang digunakan untuk memproduksi model klasifikasi atau regresi dengan mengategorikan data ke dalam kelas atau nilai yang sesuai dengan pertanyaan-pertanyaan yang diajukan secara bertahap. CART memecah data menjadi beberapa bagian yang semakin kecil dengan cara mengambil kondisi yang paling bermanfaat untuk memisahkan data ke dalam kelas atau nilai yang sesuai [12], [13]. Kriteria standar yang digunakan CART pada scikit adalah kriteria *Gini Index*. Kriteria *Gini Index* dirumuskan sebagai berikut (3).

$$Gini = \sum_{i=1}^C (p_i(1 - P_i)) \quad (3)$$

Nilai C adalah jumlah kelas yang ada, dan p_i adalah proporsi kelas i dalam *node* tersebut. Dalam memproses data, CART akan mencari kondisi yang paling berguna untuk memisahkan data menjadi kelas atau nilai yang sesuai dengan menggunakan *Gini Index* sebagai acuan. Setelah data terpecah menjadi beberapa bagian, CART akan menghitung *Gini Index* setiap bagian yang dihasilkan. Kemudian, CART akan memilih bagian yang memiliki *Gini Index* terendah sebagai *node* baru. Proses ini akan dilakukan terhadap setiap bagian yang dihasilkan hingga tidak ada lagi bagian yang dapat dipisahkan. Proses tersebut akan berhenti jika semua bagian telah murni (hanya terdiri dari satu kelas saja) atau jika tidak ada lagi atribut yang dapat digunakan untuk memisahkan data. Pada akhir proses, CART akan menghasilkan sebuah pohon keputusan yang dapat digunakan untuk melakukan klasifikasi terhadap data baru.

III. HASIL

A. Preprocessing

Setelah data yang diperoleh, tahap selanjutnya yang dilakukan adalah *preprocessing*. Proses ini merupakan tahap penting dalam pengolahan data karena di sini data akan di persiapkan untuk diolah lebih lanjut. Selain itu, *preprocessing* juga bertujuan untuk menghilangkan masalah-masalah yang mungkin terdapat pada data seperti *missing value* [14], [15] dan ketidakseimbangan data [14].

Ketika melakukan *preprocessing*, faktor-faktor yang perlu diawasi adalah: Pertama, masalah *missing value*. *Missing value* merupakan data yang tidak lengkap atau hilang pada suatu baris atau kolom. Hal ini mungkin terjadi karena beberapa faktor seperti kesalahan pengisian data, kerusakan data, atau tidak terdeteksinya data. *Missing value* dapat menjadi masalah karena data yang tidak lengkap akan mempengaruhi hasil analisis yang dilakukan. Oleh karena itu, perlu diatasi dengan cara mengisi data yang hilang tersebut dengan nilai yang sesuai. Kedua, masalah ketidakseimbangan data merupakan masalah yang sering dijumpai dalam pengolahan data. Ketidakseimbangan data terjadi ketika data yang ada memiliki skala yang berbeda-beda antar kolom. Misalnya, jika kolom A memperoleh nilai rata-rata 10 sedangkan kolom B nilai rata-ratanya adalah 1000, maka data tersebut disimpulkan tidak seimbang. Ketidakseimbangan data bisa mempengaruhi hasil analisis yang dieksekusi oleh pembelajaran mesin. Oleh karena itu, perlu dilakukan tindakan untuk menyeimbangkan data agar hasil analisis yang diperoleh lebih *valid*. Pada Tabel I ditunjukkan data produksi listrik di India yang digunakan dalam penelitian ini.

Tabel I: Data produksi listrik di India

Dates	Power Station	Monitored Cap. (MW)	Total Cap. Under Maintenance (MW)	Planned Maintenance (MW)	Forced Maintenance (MW)	Other Reasons (MW)	Programme or Expected (MU)	Actual (MU)	Excess (+) /Shortfall (-)	Deviation
2017-09-01	Delhi	2235.4	135.00	0.00	135.00	0	13	18	5.00	0.0
2017-09-01	Haryana	2720.0	2470.0	0.00	2470.0	0	28	7	-21.80	0.0
...
2022-04-13	Tripura	137.0	32.0	0.0	0.0	32	0	1	0.29	0.0
2022-04-13	NEEPCO	2027.0	963.5	313.5	650.0	0	0	12	-1.59	0.0

Setelah melakukan analisis terhadap data yang tersedia pada Tabel I, terdapat keputusan untuk menghapus kolom *Dates* dan *Deviation* dan menentukan kolom *Power Station* sebagai kategori yang akan diklasifikasi. Setelah dilakukan pengecekan, tidak ditemukan data yang kosong atau *missing value* pada dataset tersebut. Oleh karena itu, proses pengubahan data pada *missing value* tidak diperlukan lagi.

Namun, kolom *Power Station* masih dalam bentuk teks. Untuk dapat digunakan dalam proses klasifikasi, kategori tersebut perlu dikonversi menjadi angka. Oleh karena itu, dilakukan tindakan transformasi dengan menggunakan metode *LabelEncoder* dari pustaka *scikit*. Metode ini akan mengubah teks tersebut menjadi angka sesuai dengan urutan yang ditentukan.

Selain itu, fitur data yang ada juga dilakukan normalisasi dengan mengaplikasikan fungsi *MinMaxScaler* dari pustaka *scikit*. Normalisasi bertujuan untuk mendapatkan rentang angka yang sama pada semua fitur agar tidak ada fitur yang dominan dalam proses klasifikasi. Rumus *MinMaxScaler* digambarkan sebagai berikut (4).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4)$$

Dalam rumus tersebut x' adalah nilai yang sudah di normalisasi. x adalah nilai asli dari data. x_{min} adalah nilai minimum dari data yang dinormalisasi. x_{max} adalah nilai maksimum dari data yang akan dinormalisasi. Dengan menggunakan rumus ini, setiap nilai dari x akan dikonversi ke dalam rentang $[0,1]$. Nilai x' yang dihasilkan merupakan nilai relatif dari x terhadap nilai minimum dan maksimum yang ada dalam data. Rumus ini dapat digunakan untuk menormalisasi data numerik pada setiap fitur dari suatu dataset agar tidak ada fitur yang dominan dalam proses analisis.

Setelah dilakukan normalisasi, diperoleh bentuk data yang sudah siap untuk diolah seperti yang terlihat pada Tabel II.

Tabel II: Data produksi listrik yang sudah di normalisasi

41	0.11282155	0.01393838	0.	...	0.04276316	0.03862661	0.46932865
64	0.13727951	0.25502065	0.	...	0.09210526	0.01502146	0.38675786
...
118	0.03785281	0.02581181	0.	...	0.	0.02360515	0.4676341
122	0.0366718	0.	0.	...	0.	0.027897	0.45946945

B. Latihan dan Evaluasi

Setelah melakukan *preprocessing* terhadap data, langkah selanjutnya adalah membagi data menjadi data latihan dan tes menggunakan fitur *train_test_split* yang disediakan oleh pustaka *scikit*. Rasio yang dipilih dalam pembagian data tersebut

adalah 75% data latih dan 25% data tes. Rasio ini dipilih agar model klasifikasi yang digunakan dapat mempelajari dengan baik data latih dan diuji dengan menggunakan data tes yang terpisah.

Untuk mengevaluasi hasil klasifikasi yang dihasilkan oleh model tersebut, kami menggunakan metode *cross-validation*. *Cross-validation* atau yang juga dikenal sebagai *rotation estimation*, merupakan teknik validasi model yang digunakan untuk mengevaluasi bagaimana hasil analisis statistik akan diterapkan pada himpunan data yang mandiri [16]. Dalam melakukan *cross-validation*, kami memilih metode *StratifiedKfold* sebagai pilihan yang tepat. Metode *StratifiedKfold* memperhatikan proporsi kelas pada setiap *fold* agar sama dengan proporsi kelas pada data latih secara keseluruhan. Hal ini bertujuan untuk memastikan distribusi kelas pada setiap *fold* tidak terlalu berbeda, sehingga hasil validasi model lebih representatif dan dapat dipercaya. Dengan demikian, kami dapat memperoleh hasil evaluasi yang lebih akurat dari model klasifikasi yang kami gunakan.

Penggunaan metode *cross-validation* dapat membantu kami mengevaluasi model klasifikasi yang kami gunakan dengan lebih baik. Hal ini karena *cross-validation* membagi data menjadi beberapa *fold* yang terpisah, sehingga kami dapat melakukan pengujian terhadap model dengan data yang tidak terlalu tergantung pada data latih yang digunakan. Dengan demikian, kami dapat memperoleh estimasi yang lebih akurat mengenai kemampuan model klasifikasi yang kami gunakan dalam mengklasifikasi data baru.

IV. PEMBAHASAN

Selanjutnya kami mengevaluasi dengan menggunakan metode *StratifiedKfold cross-validation*, dan telah mendapatkan kesimpulan bahwa algoritma *Decision Tree* memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma *K-Nearest Neighbor*, dengan nilai akurasi sebesar 85.80% dan standar deviasi sebesar 2.21%. Selain itu, algoritma *Decision Tree* juga memiliki proses komputasi yang lebih cepat disandingkan dengan algoritma *K-Nearest Neighbor*, dengan waktu proses sebesar 51.66 detik. Hasil evaluasi dapat dilihat pada Tabel III.

Tabel III: Hasil evaluasi menggunakan *StratifiedKfold cross-validation*

Algoritma	Akurasi (persen)	Standar Deviasi (persen)	Proses Komputasi (detik)
K-Nearest Neighbor (K = 3)	80.34%	3.08%	56.27
Decision Tree	85.80%	2.21%	51.66

V. SIMPULAN DAN SARAN

Berdasarkan hasil evaluasi tersebut, bahwa algoritma *Decision Tree* lebih terdepan dalam faktor akurasi serta efisiensi proses komputasi dibandingkan dengan algoritma *K-Nearest Neighbor* pada data yang telah kami gunakan dalam penelitian ini. Namun, perlu diingat bahwa hasil evaluasi ini hanya berlaku untuk data yang telah kami gunakan dalam penelitian ini. Untuk memastikan kinerja algoritma *Decision Tree* pada data lain, diperlukan evaluasi ulang dengan menggunakan metode yang sesuai.

Berdasarkan pembahasan di atas, kami memberikan saran agar algoritma *Decision Tree* dipertimbangkan sebagai salah satu pilihan dalam menyelesaikan masalah klasifikasi pada data yang sesuai. Namun, perlu diingat bahwa hasil evaluasi hanya merupakan salah satu aspek yang perlu dipertimbangkan dalam proses seleksi algoritma *machine learning*. Faktor lain yang perlu dipertimbangkan antara lain adalah kompleksitas model, kejelasan, dan keterbatasan data yang tersedia. Oleh karena itu, penting untuk melakukan evaluasi dan pemilihan algoritma secara tepat dan sesuai dengan kebutuhan dan kondisi spesifik dari setiap masalah yang akan dipecahkan.

Simpulan dari terhadap kasus ini adalah bahwa algoritma *Decision Tree* memiliki kinerja yang lebih baik dibandingkan dengan algoritma *K-Nearest Neighbor* pada data yang telah kami gunakan dalam penelitian ini. Kami menyarankan agar algoritma *Decision Tree* dipertimbangkan sebagai salah satu pilihan dalam menyelesaikan masalah klasifikasi pada data yang sesuai, namun perlu diingat bahwa hasil ini hanya berlaku untuk data yang telah dipergunakan oleh kami dan mungkin tidak dapat diterapkan pada data lain. Sebagai tambahan, *Decision Tree* juga terbukti efektif dalam melakukan beberapa hal lainnya seperti prediksi konsumsi listrik pada gedung kantor [17], penggolongan tingkat pertumbuhan tumor tulang lytic [18], dan pada *mobile healthcare network* [19].

PUSTAKA

- [1] L. S. Tiewsoh, J. Jirásek, dan M. Sivek, "Electricity generation in India: Present state, future outlook and policy implications," *Energies* (Basel), vol. 12, no. 7, 2019, doi: 10.3390/en12071361.
- [2] S. Agrawal, S. Mani, A. Jain, dan K. Ganesan, "State of Electricity Access in India," 2020.
- [3] T. Winther dkk., "In the light of what we cannot see: Exploring the interconnections between gender and electricity access," *Energy Res Soc Sci*, vol. 60, 2020, doi: 10.1016/j.erss.2019.101334.
- [4] N. Tri, S. Saptadi, P. Chyan, dan V. M. Widjaja, "DESAIN MODEL KLASIFIKASI SAMPAH ORGANIK MENJADI BAHAN BAKU BRIKET BIOMASSA MENGGUNAKAN METODE DEEP LEARNING," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, hlm. 160–168, Sep 2022, Diakses: Des 27, 2022. [Daring]. Available: <https://ejournal.akakom.ac.id/index.php/jiko/article/view/559>
- [5] "Daily Power Generation Data — Kaggle." <https://www.kaggle.com/datasets/arvindnagaonkar/power-generation-data> (diakses Des 27, 2022).
- [6] M. M. Baharuddin, H. Azis, dan T. Hasanuddin, "ANALISIS PERFORMA METODE K-NEAREST NEIGHBOR UNTUK IDENTIFIKASI JENIS KACA," *ILKOM Jurnal Ilmiah*, vol. 11, no. 3, 2019, doi: 10.33096/ilkom.v11i3.489.269-274.

- [7] S. A. Nurhusni, R. I. Adam, dan C. Carudin, “Klasifikasi Kadar Kolesterol Menggunakan Ekstraksi Ciri Moment Invariant dan Algoritma K-Nearest Neighbor (KNN),” *Journal of Applied Informatics and Computing*, vol. 5, no. 2, 2021, doi: 10.30871/jaic.v5i2.3273.
- [8] Deti Fusvita, Asnawati, Feri Hari Utami, “KNN (K-NEAREST NEIGHBOUR) PENERAPAN ALGORITMA KNN (K-NEAREST NEIGHBOUR)DALAM KLASIFIKASI DATA PINJAMAN ANGGOTA KOPERASI,” *Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya Lubuklinggau*, vol. 3, no. 1, 2021, doi: 10.52303/jb.v3i1.32.
- [9] Y. W. Pamungkas, A. Adiwijaya, dan D. Q. Utama, “Klasifikasi Gambar Gigitan Ular Menggunakan Regionprops dan Algoritma Decision Tree,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 1, no. 2, 2020, doi: 10.30865/json.v1i2.1789.
- [10] M. F. Hidayattullah, T. Abidin, I. N. Nururriqzi, dan I. Khoeriyah, “Sistem Pendukung Keputusan Latihan Memanah Menggunakan Algoritma Decision Tree,” *Komputika: Jurnal Sistem Komputer*, vol. 10, no. 2, 2021, doi: 10.34010/komputika.v10i2.3774.
- [11] K. C. Kirana, C. B. Wira Winata, I. Astuti, dan I. R. Putra, “Prediksi rating reksadana berbasis algoritma decision tree pada sistem informasi reksadana,” *TEKNO*, vol. 29, no. 2, 2019, doi: 10.17977/um034v29i2p140-151.
- [12] F. Aziz dan A. Lawi, “Increasing electrical grid stability classification performance using ensemble bagging of C4.5 and classification and regression trees,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, hlm. 2955–2962, Jun 2022, doi: 10.11591/ijece.v12i3.pp2955-2962.
- [13] J. M. Luna dkk., “Building more accurate decision trees with the additive tree,” *Proc Natl Acad Sci U S A*, vol. 116, no. 40, hlm. 19887–19893, Okt 2019, doi: 10.1073/pnas.1816748116.
- [14] J. N. Br. S. Ringo, W. J. Mursalin, N. C. Nurfadilah, D. R. Ramadhan, dan W. O. Z. Madjida, “Perbandingan Metode Klasifikasi Multiclass untuk Pemetaan Zona Risiko COVID-19 di Pulau Jawa,” *Jurnal Komputer dan Informatika*, vol. 9, no. 1, 2021, doi: 10.35508/jicon.v9i1.3602.
- [15] I. Oktanisa dan A. A. Supianto, “Perbandingan Teknik Klasifikasi Dalam Data Mining Untuk Bank Direct Marketing,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 5, 2018, doi: 10.25126/jtiik.201855958.
- [16] F. Tempola, M. Muhammad, dan A. Khairan, “Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 5, 2018, doi: 10.25126/jtiik.201855983.
- [17] D. Ramos, P. Faria, A. Morais, dan Z. Vale, “Using decision tree to select forecasting algorithms in distinct electricity consumption context of an office building,” *Energy Reports*, vol. 8, 2022, doi: 10.1016/j.egyr.2022.01.046.
- [18] M. Benndorf, F. Bamberg, dan P. M. Jungmann, “The Lodwick classification for grading growth rate of lytic bone tumors: a decision tree approach,” *Skeletal Radiology*, vol. 51, no. 4. 2022. doi: 10.1007/s00256-021-03868-8.
- [19] S. Alex, K. J. Dhanaraj, dan P. P. Deepthi, “Private and Energy-Efficient Decision Tree-Based Disease Detection for Resource-Constrained Medical Users in Mobile Healthcare Network,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3149771.