

Implementasi Algoritma *Ant Tree Miner* Untuk Klasifikasi Jenis Fauna

Yunita Ardilla^{1*}, Wilda Imama Sabilla², Nurissaidah Ulinnuha³

¹Program Studi Manajemen Dakwah, UIN Sunan Ampel Surabaya

²Jurusan Teknologi Informasi, Politeknik Negeri Malang

³Program Studi Matematika, UIN Sunan Ampel Surabaya

^{1,3}Jln. Ahmad Yani No. 117 Surabaya, Jawa Timur, Indonesia

²Jln. Soekarno Hatta No.9 Malang, Jawa Timur, Indonesia

E-mail: yunita.ardilla@uinsby.ac.id¹, wildaimama@polinema.ac.id², nuris.ulinnuha@uinsby.ac.id³

Info Naskah:

Naskah masuk: 18 April 2021

Direvisi: 17 Agustus 2021

Diterima: 22 Agustus 2021

Abstrak

Klasifikasi adalah suatu bidang data mining yang mempunyai banyak cara metode penyelesaiannya, diantaranya yaitu *decision tree*. *Decision tree* terbukti mampu mengklasifikasikan data baik berupa data citra, data *time series* dengan baik. Namun beberapa kendala yang sering dijumpai pada metode *decision tree* yaitu *running time* yang dibutuhkan dalam eksekusi algoritma ini cukup lama, sehingga dalam penelitian ini diusulkan untuk menggunakan algoritma *ant tree miner* yang merupakan algoritma pengembangan dari *decision tree* tipe C4.5. Cara kerja *ant tree miner* yaitu dengan memanfaatkan *ant colony optimization* dalam proses pembangunan struktur *tree*-nya. Diharapkan dengan penggunaan optimasi *ant colony* dapat lebih mengoptimalkan *tree* yang akan terbentuk. Dari uji coba yang sudah dilakukan dihasilkan akurasi sekitar 95% dalam proses pengklasifikasian Zoo dataset dengan jumlah semut antara 60 – 90.

Keywords:

algoritma ant tree miner;

algoritma c4.5;

k-fold cross validation;

data mining;

klasifikasi.

Abstract

Classification is a field of data mining that has many methods, one of them is decision tree. Decision tree is proven to be able to classify many kinds of data such as image data and time series data. However, there are several obstacles that are often encountered in the decision tree method. Running time required for the execution of this algorithm is quite long, so this study proposed to use the ant tree miner algorithm which is a development algorithm from the C4.5 decision tree. Ant tree miner works by utilizing ant colony optimization in the process of building its tree structure. Use ant colony optimization expected can optimize the tree that will be formed. From the testing that have been carried out, an accuracy of about 95% is obtained in the process of classifying Zoo dataset with the number of ants between 60 - 90.

*Penulis korespondensi:

Yunita Ardilla

E-mail: yunita.ardilla@uinsby.ac.id

1. Pendahuluan

Klasifikasi adalah suatu bidang *data mining* yang mempunyai banyak cara metode penyelesaiannya. Klasifikasi dilakukan dengan pembentukan model (*classifier*) berdasarkan data latih dan melabelkan data berdasarkan model tersebut [1]. Dalam melakukan proses klasifikasi ada beberapa metode yang sering digunakan, diantaranya adalah *decision tree*. *Decision tree* merupakan algoritma klasifikasi yang populer digunakan karena hasil dari metode ini mudah diinterpretasikan [2]. Selain itu, karena permasalahan klasifikasi dapat dipandang sebagai permasalahan optimasi, dimana tujuannya adalah menemukan fungsi terbaik (model) yang merepresentasikan prediksi dalam data, maka metode klasifikasi juga dapat digabungkan dengan algoritma optimasi yang lain [3].

Decision tree dapat digunakan sebagai metode klasifikasi di berbagai bidang. Antara lain dalam bidang pendidikan untuk klasifikasi data peserta didik dengan akurasi 96,73% [4]. Contoh lain di bidang kesehatan untuk klasifikasi kanker payudara dengan akurasi sebesar 93.63% [5]. Pada penelitian untuk mengklasifikasikan keanekaragaman ikan tuna di Indonesia, dihasilkan bahwa *decision tree* mampu mengklasifikasi dengan baik namun ada kendala dimana *running time* waktu yang digunakan cukup lama [6].

Sehingga dalam penelitian ini dibuatlah uji coba implementasi *decision tree* yang digabungkan dengan algoritma optimasi untuk menemukan model *tree* yang terbaik. Optimasi dilakukan dengan menggunakan algoritma *Ant Colony Optimization* untuk menyelesaikan *running time* yang relatif lama pada algoritma *decision tree*. Optimasi *decision tree* dengan *Ant Colony Optimization* dengan metode *Ant-Tree-Miner*. Penelitian ini menerapkan metode *Ant-Tree-Miner* untuk mengklasifikasikan jenis fauna dan melakukan analisis terhadap *running time* klasifikasi dibandingkan dengan metode *decision tree*.

Sistem pengklasifikasian spesies fauna ini dibangun dengan menggunakan perangkat lunak Neatbeans 7.3 dengan menggunakan bahasa pemrograman Java. Data set yang digunakan dalam penelitian ini adalah *Zoo Database (Zoo)*, data set ini diambil dari tempat penyimpanan data *UCI machine learning* [7].

Penggunaan *Zoo Database (Zoo)* dipilih karena keanekaragaman makhluk hidup khususnya fauna sangatlah tinggi dengan berbagai karakteristiknya. Di dunia ini terdapat ribuan spesies fauna yang dapat digolongkan ke jenis tertentu, seperti mamalia, burung, ikan, reptil, dan lain-lain. Namun karena banyaknya makhluk hidup tersebut, manusia menemui berbagai kesulitan untuk menentukan pada jenis apa suatu spesies harus diklasifikasikan. Dampaknya terjadi kesulitan untuk mengetahui fauna apa yang menjadi endemik di suatu daerah tertentu. Oleh karena itu dibutuhkan suatu metode klasifikasi untuk menentukan suatu spesies termasuk ke dalam jenis tertentu melalui identifikasi ciri-ciri fisiknya. Diharapkan dengan diketahuinya jenis fauna berdasarkan ciri-cirinya, dapat diketahui pula fauna yang menjadi endemik di suatu daerah untuk tetap dijaga kelestariannya

2. Metode

Metode yang digunakan dalam penelitian ini adalah *ant tree miner*, *ant tree miner* merupakan algoritma pengembangan dari metode *decision tree* tipe C4.5 dengan memanfaatkan *ant colony optimization* dalam proses pembangunan struktur *tree*-nya. Jika dalam metode *decision tree* tipe C.45 proses pemilihan atribut yang akan digunakan sebagai *node* berdasarkan pada nilai informasi heuristik atau *information gain ratio* yang paling besar, berbeda halnya dengan *ant tree miner* [8]. Dalam metode *ant tree miner* pemilihan atribut dilakukan secara *probabilistic* dengan mempertimbangkan *heuristic* dan jumlah feromon semut [9] [10]. Untuk lebih jelasnya bisa dilihat di diagram alur Gambar 1.

Algoritma *ant tree miner* mengikuti struktur algoritma *Ant Colony Optimization (ACO)*. Mula-mula dilakukan inialisasi feromon dan menghitung nilai informasi heuristik untuk setiap atribut dalam data latih. Kemudian proses masuk kedalam iterasi dimana setiap semut dalam koloni membentuk suatu model *decision tree* baru hingga iterasi maksimal tercapai atau algoritma menjadi konvergen. Semut mementuk *decision tree* secara *top down* dengan cara memilih atribut berdasarkan probabilitas yang dipengaruhi oleh feromon dan informasi *heuristic* [11]. Prosedur pembuatan *decision tree* memiliki tiga parameter yaitu kumpulan sampel data latih, himpunan atribut predictor, dan *edge* yang diikuti semut [12].

Setelah pembentukan *tree* selesai, maka *tree* tersebut di-*pruning* untuk menyederhanakan *tree* dan menghindari *overfitting* terhadap data latih. *Overfitting* adalah fenomena dimana model terlalu cenderung ke data *training*. Setelah *pruning*, *tree* dievaluasi dan *tree* terbaik dalam iterasi di *update* jika kualitas yang baru dibuat lebih bagus dari pada kualitas *tree* sebelumnya. *Tree* terbaik hasil iterasi yang dibentuk oleh semut-semut digunakan dalam proses update feromon. Lalu *tree* terbaik global diperbarui dan dilanjutkan ke iterasi selanjutnya. Ketika sudah mencapai iterasi maksimal / algoritma konvergen, *tree* global yang terbaik dipilih sebagai model *decision tree* [13].

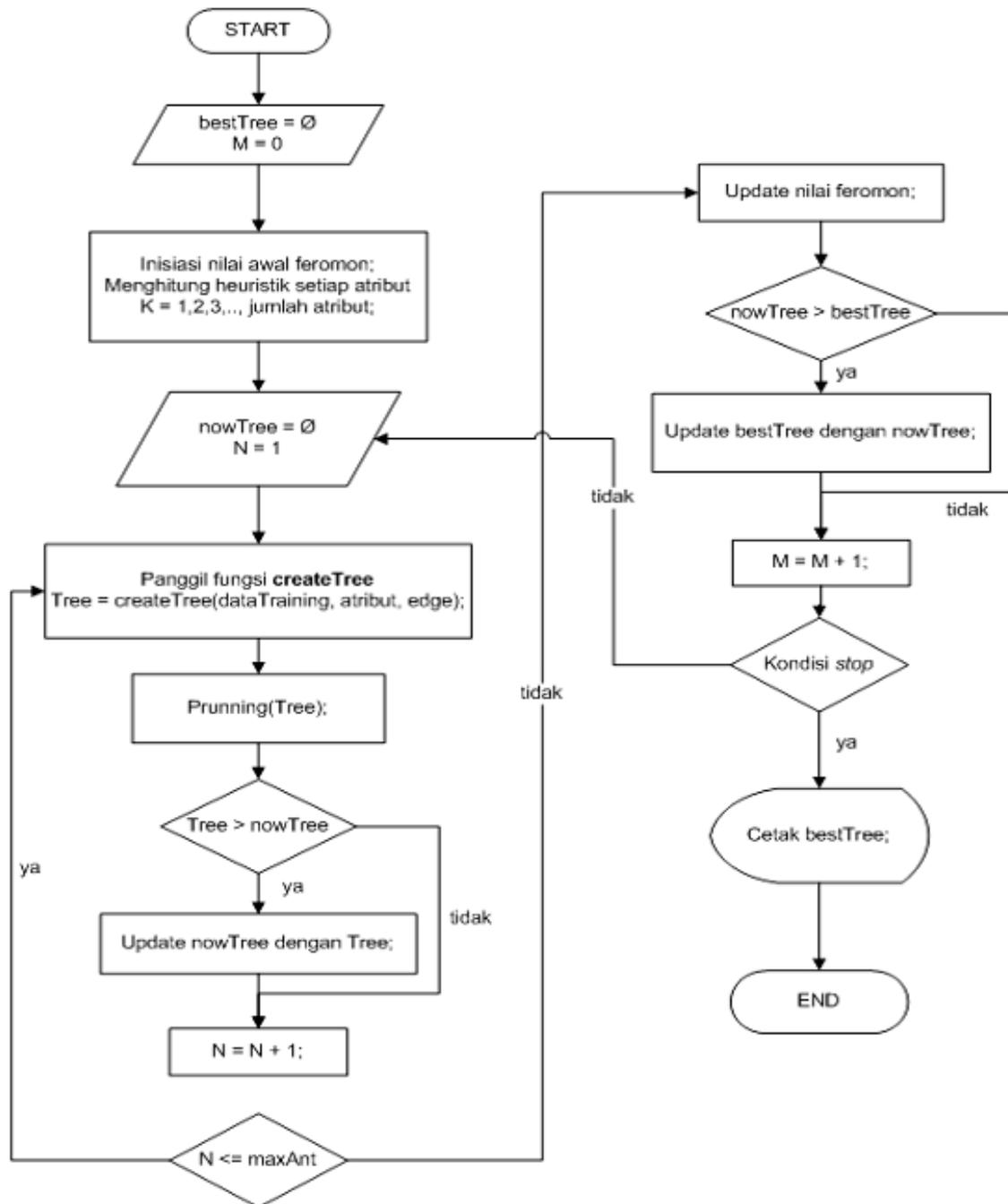
2.1 Evaluasi Performa

Terdapat beberapa macam metode yang digunakan pengukuran performa. Contohnya adalah akurasi, *precision*, *recall*, dan *F-measure*. Perhitungan performa dilakukan berdasarkan jumlah data dengan prediksi yang benar maupun yang salah prediksi. Data hasil prediksi untuk mengukur performa dapat direpresentasikan dalam matriks *confusion* seperti pada Tabel 1 [14].

Tabel 1. *Matriks Confusion*

	Data Positif Sebenarnya	Data Negatif Sebenarnya
Data Terprediksi Positif	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
Data Terprediksi Negatif	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Nilai *true positive (TP)* merupakan jumlah data dengan kelas prediksi dan kelas sebenarnya sama-sama positif. Nilai *false positive (FP)* merupakan jumlah data



Gambar 1. Diagram Alur Metode Ant Tree Miner

yang sebenarnya kelas negatif terprediksi sebagai kelas positif. Nilai *false negative* (FN) merupakan jumlah data dengan kelas positif yang terprediksi sebagai kelas negatif. Nilai *true negative* (TN) merupakan jumlah data dengan kelas negatif yang juga terprediksi sebagai kelas negatif. Akurasi merupakan perbandingan antara jumlah data yang diprediksi dengan benar terhadap jumlah data uji coba keseluruhan, dapat dirumuskan dengan persamaan (1).

$$Akurasi = \frac{TP+TN}{TP+TN+FN+FP} \quad (1)$$

2.2 K-Fold cross Validation

Cross validation adalah suatu metode statistika untuk melakukan evaluasi dan membandingkan algoritma pembelajaran. Metode ini membagi data ke dalam dua bagian, yaitu data latih dan data uji. Data latih merupakan data yang digunakan dalam proses pembelajaran, sedangkan data uji merupakan data yang digunakan dalam proses uji dengan model yang dibentuk saat proses pembelajaran. Setiap data akan menjadi data latih dan data uji, hal ini disebut dengan *cross-over*. Perlakuan terhadap data tersebut akan membuat proses latih uji lebih valid. *K-fold cross validation* merupakan salah satu jenis dari *cross*

validation. Sejumlah data akan dibagi ke dalam k bagian yang sama, lalu dilakukan proses latih dan uji [15].

Proses ini dilakukan sebanyak k iterasi dengan bagian data yang berbeda dengan $fold$ ke- i sebagai data uji dan $fold$ ke-1 hingga $fold$ ke- k tanpa $fold$ ke- i sebagai data latih. Setelah proses k -fold cross validation selesai dilakukan, langkah berikutnya yaitu menghitung performa model prediksi dengan menggunakan metode akurasi. Untuk memaksimalkan performa tersebut, proses evaluasi dengan k -fold cross validation bisa dilakukan secara berulang-ulang.

2.3 Perancangan Data

Perancangan data adalah hal yang sangat penting dalam sistem karena diperlukan data yang tepat agar perangkat lunak beroperasi dengan benar. Sistem ini membutuhkan data masukan dan data keluaran. Data masukan berupa data set yang akan digunakan sebagai data latih dan data uji. Data keluaran adalah data berupa model prediksi, evaluasi performa, dan hasil klasifikasi. Data set yang digunakan sebagai data masukan adalah Zoo Data Set yang bisa didapat di situs UCI machine learning. ZOO Data Set terdiri dari 101 sampel dengan 15 atribut kategorikal, 1 atribut numerik dan 7 label kelas. Kelas dalam data set ini adalah *mammals*, *birds*, *reptiles*, *fish*, *amphibians*, *invertebrate*, dan *crustacean* sesuai dengan Tabel 2, sedangkan atribut sesuai urutan dalam data set adalah *Hair*, *Feather*, *Eggs*, *Milk*, *Airborne*, *Aquatic*, *Predator*, *Toothed*, *Backbone*, *Breathes*, *Venomous*, *Fins*, *Legs*, *Tail*, *Domestic*, dan *Catsize*.

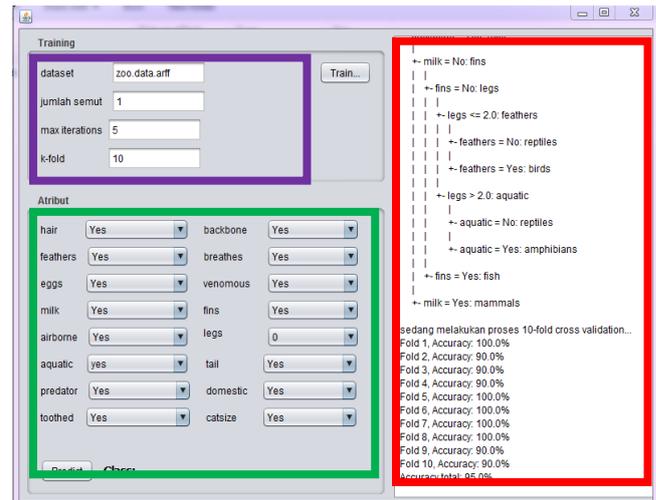
Tabel 2. Distribusi Kelas dalam Data Set

Label Kelas	Jumlah
<i>Mammals</i>	41
<i>Birds</i>	20
<i>Reptiles</i>	5
<i>Fish</i>	13
<i>Amphibians</i>	4
<i>Invertebrate</i>	8
<i>Crustacean</i>	10

Dataset ini nantinya akan dilakukan pengkategorian terlebih dahulu sebelum digunakan dalam proses pembelajaran. Tujuan data set di lakukan pengkategorian adalah untuk mengubah data *boolean* menjadi data yang bernilai [*Yes*, *No*] untuk setiap atribut kecuali pada atribut *legs*, atribut *legs* harus diubah ke dalam bentuk nominal.

3. Hasil dan Pembahasan

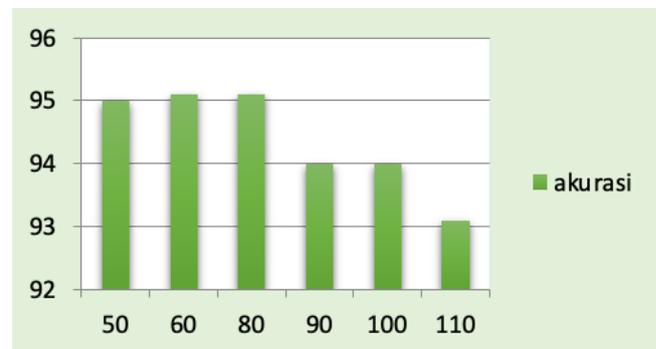
Antarmuka dalam perangkat lunak seperti pada Gambar 2, dibagi menjadi satu macam dimana antarmuka tersebut berfungsi untuk antarmuka proses latih *ant tree miner* antarmuka klasifikasi jenis fauna dan antarmuka untuk pengisian parameter meliputi jumlah semut, *max iterations* dan *k-fold*.



Gambar 2. Antarmuka Perangkat Lunak

Tabel 3. Hasil Skenario Uji Coba Penggantian Nilai Jumlah Semut

Jumlah Semut	Akurasi
50	95%
60	95.09%
80	95.09%
90	94%
100	94%
110	93%



Gambar 3. Grafik Performa Model Berdasarkan Jumlah Semut

Gambar 2 merupakan rancangan antarmuka perangkat lunak. Kotak warna ungu merupakan kolom untuk mengisi parameter *max iterations*, jumlah semut dan *k-fold*. Kotak warna hijau merupakan kolom yang digunakan untuk mengisi data masukan dari pengguna beserta hasil prediksinya. Kotak warna merah merupakan antarmuka hasil model pohon keputusan dan pengukuran performa model pohon keputusan tersebut.

Skenario uji coba yang dilakukan adalah dengan mengganti jumlah semut. Nilai maksimum iterasi ditetapkan 500. Uji coba dilakukan sebanyak empat kali dengan merubah nilai parameter jumlah semut yaitu 80, 90, 100 dan 110. Untuk setiap kali uji coba dilakukan perhitungan performa dari model yang dihasilkan dengan k -fold cross validation dengan nilai $k = 10$. Hasil dari uji coba ini ditunjukkan dalam Tabel 3 dan Gambar 3. Dari hasil tersebut, bisa ditarik kesimpulan bahwa jumlah semut dalam

proses latih mempengaruhi peningkatan performa model yang dihasilkan. Semakin banyak semut yang digunakan maka performa model cenderung semakin berkurang.

4. Kesimpulan

Penelitian ini melakukan klasifikasi jenis fauna menggunakan algoritma *ant tree miner*. *Ant tree miner* mengoptimasi algoritma *decision tree* dengan memanfaatkan *ant colony optimization* (ACO). ACO akan menghasilkan model *tree* yang dapat mengklasifikasikan fauna ke label kelas secara optimum. Model yang dihasilkan *ant tree miner* memiliki performa dengan akurasi sebesar 95,09%. Model tersebut akan memberikan performa yang maksimal ketika dilakukan algoritma ACO dengan parameter jumlah semut antara 60-80.

Daftar Pustaka

- [1] E. N. I. Karabadi *et al.*, "An evolutionary scheme for decision tree construction," *Knowledge-Based Syst.*, vol. 119, pp. 166–177, 2017.
- [2] G. G. Moisen, "Classification and Regression Trees," *Encycl. Ecol.*, vol. 1, pp. 582–588, 208AD.
- [3] R. Rismala and M. D. Sulistiyono, "Penerapan teknik klasifikasi pada system rekomendasi menggunakan algoritma genetika," *J. Ilm. Teknol. Inf. Terap.*, vol. II, no. 3, pp. 197–205, 2016, [Online]. Available: <http://journal.widyatama.ac.id/index.php/jitter/article/view/108>.
- [4] I. Sutoyo, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik," *J. Pilar Nusa Mandiri*, vol. 14, no. 2, p. 217, 2018, doi: 10.33480/pilar.v14i2.926.
- [5] P. Hamsagayathri and P. Sampath, "Performance Analysis of Breast Cancer Classification Using Decision Tree Classifiers," *Int. J. Curr. Pharm. Res.*, vol. 9, no. 2, p. 19, 2017, doi: 10.22159/ijcpr.2017v9i2.17383.
- [6] T. Arifin, "Optimasi Decision Tree menggunakan Particle Swarm Optimization untuk klasifikasi sel Pap Smear," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 3, pp. 572–579, 2020.
- [7] D. Hand, "Principles of data mining," *Drug Saf*, vol. 30, no. 7, pp. 621–622, 2007.
- [8] F. Otero, A. Freitas, and C. Johnson, "Inducing decision trees with an ant colony optimization algorithm," *Appl. Soft Comput.*, vol. 12, no. 11, pp. 3615–3626, 2012, doi: <https://doi.org/10.1016/j.asoc.2012.05.028>.
- [9] A. Amrulloh and A. P. Wibowo, "Implementasi Algoritma Decision Tree Untuk Mengklasifikasi Kondisi Kesuburan Pria," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 1, no. 1, 2019, doi: 10.26905/jasiek.v1i1.3096.
- [10] A. U. Islam and S. H. Ripon, "Rule Induction and Prediction of Chronic Kidney Disease Using Boosting Classifiers , Ant-Miner and J48 Decision Tree," in *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019, no. February, doi: 10.1109/ECACE.2019.8679388.
- [11] H. N. K. Al-behadili, K. R. Ku-mahamud, and R. Sagban, "Rule Pruning Techniques in the Ant-Miner Classification Algorithm and Its Variants : A Review," *2018 IEEE Symp. Comput. Appl. Ind. Electron.*, pp. 78–84, 2018.
- [12] S. Fidanova, "Ant Colony Optimization," *Ant Colony Optim. Appl.*, vol. 947, pp. 3–8, 2021, doi: 10.1007/978-3-030-67380-2_2.
- [13] B. Jyoti and A. Sharma, "AntMiner: Bridging the Gap Between Data Mining Classification Rule Discovery and Bio-Inspired Algorithms," in *ICIL*, 2019, pp. 269–277.
- [14] Y. Ardilla, H. Tjandrasa, and I. . Arieshanti, "Deteksi Penyakit Epilepsi dengan Menggunakan Entropi Permutasi, K-means Clustering, dan Multilayer Perceptron," *J. Tek. ITS*, vol. 3, no. 1, pp. 1–5, 2014.
- [15] J. Triajianto, Y. Purwananto, and R. Soelaiman, "Implementasi Sistem Klasifikasi Fuzzy Berbasis Optimasi Koloni Semut untuk Diagnosa Penyakit Diabetes," *J. Tek. ITS*, vol. 2, no. 1, pp. 99–104, 2013.