

# Optimasi dan Perancangan Antena Menggunakan Metode *Modified Efficient K-Nearest Neighbors*

Deden Rustiana<sup>\*1</sup>, Nina Rahayu<sup>2</sup>, Hindriyanto Dwi Purnomo<sup>3</sup>, Ahmad Bayu Yadila<sup>4</sup>, Hendra Kusumah<sup>5</sup>

<sup>1,2,4,5</sup> Prodi Sistem Komputer, Universitas Raharja

<sup>3</sup> Prodi Doktor Ilmu Komputer, Universitas Kristen Satya Wacana

<sup>\*1</sup>deden.rustiana@raharja.info; <sup>2</sup>nina.rahayu@raharja.info; <sup>3</sup>hindriyanto.purnomo@uksw.edu;

<sup>4</sup>ahmadbayu@raharja.info, <sup>5</sup>hendra.kusumah@raharja.info

## Abstrak

Untuk memastikan akurasi dan mencegah perilaku machine learning yang tidak diinginkan terjadi ketika model machine learning memberikan prediksi akurat untuk data pelatihan tetapi tidak untuk data baru yang biasa disebut Overfitting, teknik machine learning yang efektif biasanya dilatih pada kumpulan data besar. Namun, ketika pengumpulan data rumit, kumpulan data yang besar menghambat penyebaran teknik machine learning. Algoritma K-Nearest Neighbors (KNN) ditingkatkan dalam penelitian ini untuk mengatasi masalah dan menyajikan pendekatan machine learning yang unik sehingga dapat mengekstraksi lebih banyak fitur dari kumpulan data yang besar. Metode ini bekerja 5 hingga 30 kali lebih cepat daripada teknik machine learning konvensional seperti jaringan syaraf tiruan (ANN) dan pengoptimalan Bayesian. Parameter antena digunakan untuk mengoptimalkan kemudian dioptimalkan menggunakan metode yang disarankan, dan cabang terpisah dibuat untuk menjalankan alat simulasi (seperti HFSS) dan memperbarui dataset saat pelatihan daripada membuatnya sebelumnya. Empat contoh antena lainnya, serta machine learning tambahan dan teknik berbasis gradien, digunakan untuk mendukung validitas dan efektivitas pendekatan yang disarankan. Kesimpulannya, metode ini disarankan dapat menghasilkan desain antena ideal dan harga terjangkau.

**Kata Kunci**—Machine learning, K-Nearest Neighbors, Optimasi Antena

## Abstract

*To ensure accuracy and prevent undesirable machine learning behaviors that occur when a machine learning model provides accurate predictions for training data but not for new data, commonly called Overfitting, effective machine learning techniques are usually trained on large datasets. However, when data collection is complicated, large datasets hinder the deployment of machine learning techniques. The K-Nearest Neighbors (KNN) algorithm is improved in this study to address the problem and presents a unique machine learning approach that can extract more features from large datasets. This method works 5 to 30 times faster than conventional machine learning techniques such as artificial neural networks (ANN) and Bayesian optimization. Antenna parameters are used to optimize then optimized using the suggested method, and a separate branch is created to run simulation tools (such as HFSS) and update the dataset during training rather than creating it beforehand. Four other antenna examples, as well as additional machine learning and gradient-based techniques, are used to support the validity and effectiveness of the suggested approach. In conclusion, this method is suggested to produce ideal antenna designs at affordable prices.*

**Keywords**—Machine learning, K-Nearest Neighbors, Antenna Optimization

## 1. PENDAHULUAN

Baru-baru ini, semakin banyak insinyur mulai meningkatkan dan mengoptimalkan desain mereka dengan bantuan metode machine learning untuk memenuhi persyaratan antena berkinerja tinggi [1]. Meskipun model yang terlatih dengan baik biasanya bekerja secara akurat dan efisien, proses pelatihan mereka biasanya melibatkan sejumlah besar sampel pelatihan dan akan menimbulkan masalah kritis jika akuisisi sampel pelatihan sangat memakan waktu [2]. Terkadang biaya proses pelatihan dalam metode machine learning bisa lebih besar dari biaya metode optimasi tradisional lainnya [3]. Misalnya, model JST yang menggunakan 12.800.000 sampel data dilatih untuk mengoptimalkan metamaterial fotonik 10 lapis, yang berisi sepuluh variabel untuk ketebalan sepuluh lapisan [4]. Dalam, jaringan saraf dengan 600 sampel data pelatihan digunakan untuk mengoptimalkan filter mikrostrip bandstop dari lima variabel geometri [5]. Pada, model JST dengan 100 sampel data latih digunakan untuk mengoptimalkan antena resonator Fabry-Perot (FP) dari tiga variabel geometri [6]. Semua metode ini memerlukan waktu nontrivial untuk menyiapkan dataset, membatasi penerapannya dalam kehidupan nyata [7]. Dari sudut pandang ini, bukanlah pilihan yang baik untuk menggunakan metode machine learning ketika biaya proses pelatihan di luar toleransi [8]. Sedangkan pada contoh yang diberikan pada bagian IV kasus C, metode ini hanya membutuhkan 8 data pelatihan dan sampel pengujian untuk mengoptimalkan antena resonator 3 parameter; pada contoh diberikan pada bagian IV kasus D, hanya diperlukan 30 sampel data latih dan uji untuk model antena 9 parameter dengan menggunakan metode ini [10].

Untuk mengurangi kompleksitas sistem jaringan saraf dan jumlah sampel data pelatihan dan pengujian, metode baru yang diajukan berdasarkan algoritma K-Nearest Neighbors (KNN) yang dimodifikasi [11]. Model belum tentu terlatih dengan baik sebelum digunakan untuk memprediksi hasil yang optimal dan oleh karena itu dapat dimulai dengan sampel data pelatihan dan pengujian yang sangat sedikit (setidaknya lima) [12]. Model dapat memprediksi hasil yang optimal selama proses pelatihan [13]. Kemudian prediksi serta hasil eksaknya akan direkam kembali ke dataset pelatihan asli. Untuk mendapatkan hasil yang tepat selama proses pelatihan, kami membangun cabang untuk mengontrol alat simulasi (misalnya HFSS). Oleh karena itu, metode ini menghindari pemilihan random data pelatihan dan malah menghasilkan data pelatihan yang lebih berharga berdasarkan model yang ada tetapi tidak dipersiapkan dengan baik. Model dapat berkembang dengan memprediksi dirinya sendiri yang disebut belajar mandiri [14].

Metode regresi berbasis K-Nearest Neighbors (KNN) dapat digunakan dalam kasus di mana label data kontinu, yang merupakan parameter geometri target antena dalam makalah ini [15]. Label yang ditetapkan ke titik query dihitung berdasarkan rata-rata label dari K-Nearest Neighbors (KNN) (standarnya adalah lima) [16]. Karena sedikit pengetahuan tentang antena yang tersedia, kami memilih bobot yang seragam dalam kasus ini dan setiap titik di lingkungan lokal kemudian berkontribusi secara seragam pada regresi titik kueri [17]. Singkatnya, satu-satunya informasi awal yang perlu kita masukkan adalah domain wajar dari parameter geometris dan model antena target dasar yang akan dioptimalkan [18]. Oleh karena itu, algoritma dapat dengan mudah digeneralisasikan ke skenario lain [19]. Untuk memverifikasi universalitas dan keefektifan algoritma, empat kasus yang sangat berbeda akan diuji dan diberikan nanti dalam makalah ini [20].

Penulisan ini disusun sebagai berikut. Bagian II menjelaskan metode K-Nearest Neighbors (KNN) konvensional dan skema optimisasi yang diusulkan. Bagian III mendemonstrasikan penerapan metode yang diusulkan. Bagian IV memberikan empat kasus yang berbeda termasuk desain untuk reflektor Bragg, model antena stasiun basis industri, model antena yang dioptimalkan dengan metode machine learning lainnya dan proses desain model antena dasar untuk menguji dan memverifikasi algoritma yang diusulkan. Akhir penulisan ini dirangkum pada kesimpulan dalam bagian V.

## 2. METODE PENELITIAN

### 2.1. Algoritma

Pendekatan regresi pembelajaran berbasis K-Nearest Neighbors (KNN) yang diawasi, algoritma k-nearest neighbor digunakan dalam penelitian ini. Menemukan jumlah sampel pelatihan yang telah ditentukan sebelumnya yang paling dekat dengan titik baru adalah ide mendasar di balik algoritma tetangga terdekat. Dengan menetapkan nilai ke setiap titik data baru bergantung pada seberapa miripnya dengan set pelatihan, algoritma memprediksi titik data baru menggunakan "kemiripan fitur". K menentukan jumlah *Neighbors* yang diperlukan untuk menyelesaikan prediksi saat menggunakan regression KNN. X singkatan dari vektor input, Y untuk vektor output, dan K untuk jumlah *neighbors*. Meskipun angka lain juga dapat digunakan, dalam penelitian ini kami menggunakan k=5 untuk mencapai kompromi antara akurasi dan kecepatan. Meskipun bobot  $1/d$  atau bobot lainnya, di mana d adalah jarak ke *neighbors*, juga dapat digunakan, kami memilih bobot yang seragam untuk metode pembobotan dalam penelitian ini. Selain itu, jarak *Euclidean* adalah jenis jarak yang kami gunakan pada penulisan ini. *Neighbors* berfungsi sebagai set pelatihan algoritma dan dipilih dari set nilai objek dalam data input.

Dengan melakukan langkah-langkah tersebut, kita dapat mempraktikkan pendekatan KNN:

- Jarak antara setiap titik latihan dan titik baru harus dihitung.
- *K titik* terdekat dipilih berdasarkan jarak.
- Sebagai prediksi akhir untuk titik baru, kami menggunakan nilai rata-rata tertimbang dari titik data terpilih ini.

Secara matematis, asumsikan nilai *k* untuk KNN dan titik prediksi  $x_0$ , lalu gunakan  $N_0$  untuk menyatakan *k* observasi pelatihan terdekat dengan titik prediksi  $x_0$ . KNN mengembalikan estimasi  $f(x_0)$  menggunakan rata-rata dari semua respons di  $N_0$

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_0} y_i$$

Nilai pada setiap titik *mesh* ditentukan dalam makalah ini dengan terlebih dahulu membuat jaring padat dan kemudian menggunakan regression KNN untuk menentukan nilainya. Berikut adalah langkah-langkah khusus untuk mengimplementasikan KNN.

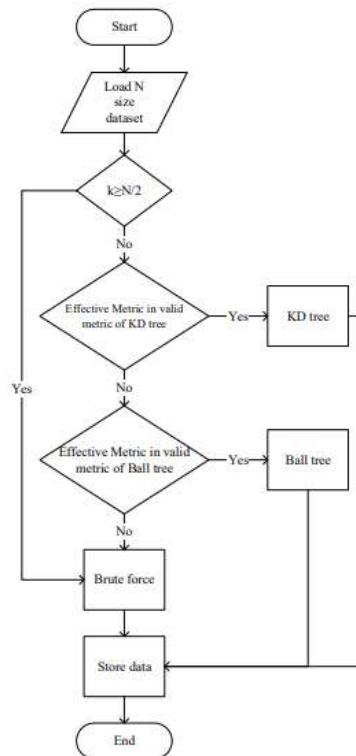
#### 2.1.1. Menghitung jarak

Untuk menentukan jarak dalam KNN, ada banyak teknik. Jarak Euclidean, Manhattan, dan Hamming adalah tiga teknik yang paling banyak digunakan.

$$\begin{aligned} \text{Euclidean distance} &= \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \\ \text{Manhattan distance} &= \sum_{i=1}^k |x_i - y_i| \\ \text{Hamming distance} &= \sum_{i=1}^k D_i, D_i = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \end{aligned}$$

Biasanya, kategorisasi dilakukan dengan menggunakan jarak Hamming. Di sini, kita hanya diberi pilihan untuk memilih jarak Euclidean.

Metode KNN pada Gambar. 1 melalui prosedur pelatihan yang melibatkan penghitungan dan penyimpanan efektif jarak antara setiap pasang titik dalam dataset. Untuk menghitung neighbors terdekat, biasanya digunakan tiga teknik efisien: brute force, KD tree, dan ball tree. Ketiga strategi tersebut harus digunakan dalam berbagai keadaan, menurut.



**Gambar 1. Menunjukkan Prosedur Pelatihan KNN.**

Jika  $k \geq N/2$  atau metrik efektif (fitur komputasi jarak, dalam kasus kami Euclidean), tidak ada dalam daftar metrik yang sesuai baik untuk pohon KD atau pohon bola, untuk nilai  $k$  dan  $N$  yang disediakan adalah jumlah data dalam kumpulan data, kekerasan akan digunakan. Jika tidak, keputusan akan didasarkan pada ukuran efektif, yang termasuk dalam daftar metrik yang dapat diterima untuk pohon KD atau pohon bola. Pohon KD adalah algoritma pilihan dalam makalah ini berdasarkan metrik efisien dari situasi kita.

**2.1.2. Memilih K-Factor**

Pertukaran bias-varians umumnya mempengaruhi nilai ideal  $k$ .

$$\hat{f}_k(X) = E_D[f_k(X)]$$

di mana  $X$  adalah data masukan,  $Y$  adalah kebenaran dasar dari  $X$ ,  $D$  adalah kumpulan data,  $f_k(X)$  adalah prediksi berdasarkan kumpulan data dan nilai  $k$ .  $f_k(X)$  adalah prediksi model yang diharapkan.

Secara khusus, dalam kasus penelitian ini data pelatihan dan pengujian berasal dari solusi analitik atau alat simulasi numerik, yang dianggap sebagai solusi yang tepat. Oleh karena itu, varian dalam kasus ini memiliki kesalahan yang sangat kecil. Perbedaannya adalah

$$Var(X) = E_D \left[ \left( f_k(X) - \hat{f}_k(X) \right)^2 \right]$$

Dengan kata lain,  $k$  yang digunakan pendekatan kita tidak perlu terlalu besar. Selanjutnya, prasangka adalah

$$\text{bias}^2(X) = E_D \left[ \left( Y - \hat{f}_k(X) \right)^2 \right]$$

Dalam sistem, kesalahan prediksi yang diprediksi adalah

$$\begin{aligned} EPE &= E_D \left[ \left( Y - f_k(X) \right)^2 \right] \\ &= \text{Var}(X) + \text{bias}^2(X) \\ &= \text{bias}^2(X) \\ &= E_D \left[ \left( Y - \hat{f}_k(X) \right)^2 \right] \end{aligned}$$

Untuk mencegah overfitting dalam kebanyakan keadaan, kita harus menggunakan  $k = 10$ . Untuk menyeimbangkan akurasi dan kecepatan, kami menggunakan nilai berbasis pengalaman  $k = 5$  dalam makalah ini.

### 2.1.3. Pemilihan Berat

Ada sedikit keraguan bahwa prediksi hanya dipengaruhi oleh  $k$  negihbors terdekat, tetapi masing-masing negihbors ini mungkin memiliki pengaruh yang berbeda. Jarak harus diterjemahkan ke dalam besaran lain yang dapat digunakan sebagai bobot untuk mendapatkan efek ini. Beberapa fungsi kernel harus diikuti saat berpindah dari jarak ke bobot. Kernel persegi panjang, kernel segitiga, kernel cosinus, kernel Gauss, dan kernel inversi adalah contoh tipikal dari jenis fungsi ini. Secara empiris, pemilihan kernel khusus tidak terlalu penting, khususnya untuk situasi antenna dalam penelitian ini, karena hanya sedikit informasi sebelumnya yang diekspos (selain dari kasus khusus kernel persegi panjang yang memberikan bobot yang sama untuk semua negihbors). Untuk menghindari menghukum negihbors mana pun, kami memutuskan untuk menggunakan bobot yang seragam (inti persegi panjang) dalam penelitian ini.

## 3. HASIL DAN DISKUSI

### 3.1. Pendahuluan Metode KNN Efisien Modifikasi

Pendekatan KNN efisien yang diperbarui diimplementasikan dalam penelitian ini menggunakan alat scikit-learn versi 0.21.2, solusi analitik dihitung menggunakan MATLAB R2018b, dan solusi numerik diperoleh menggunakan ANSYS 18.2.0 (HFSS). RAM 8 GB, komputer Intel Core i7 2,5 GHz digunakan untuk semua perhitungan dalam penelitian ini.

Seluruh alur kerja metode KNN efisien yang disarankan digambarkan pada Gambar. 2. Berikut adalah tahapan yang menunjukkan bagaimana metode ini diterapkan:

**Langkah 1:** Sebagai permulaan, buatlah dataset. Untuk pendekatan ini, ini adalah tahap yang krusial. Seperti teknik machine learning tradisional, kami tidak harus memilih sampel dalam jumlah besar untuk tahap ini. Menurut penelitian empiris, sebagian besar waktu hanya diperlukan 10.100 sampel pada langkah ini. Pemilihan sampel yang seragam di seluruh rentang variabel juga diperlukan. Dalam kasus individual berikut, informasi set data lebih lanjut ditampilkan.

**Langkah 2:** Bagi dataset asli menjadi dua bagian secara acak: 90% akan digunakan untuk pelatihan, dan 10% akan digunakan untuk pengujian. Standarisasi dataset dilakukan

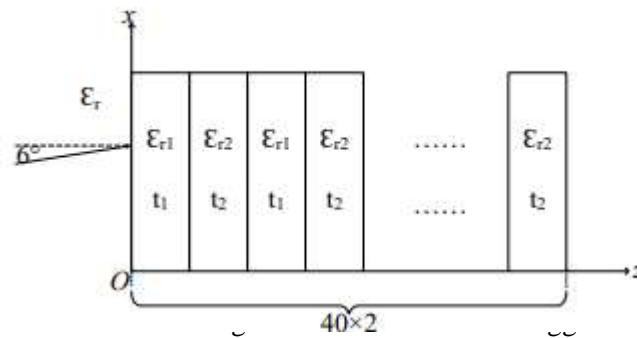
dengan menggunakan tool Standard Scaler preprocessing, kemudian training dan testing tool K-Neighbors Regressor. Pada langkah ini, kita memperoleh model  $\Omega: x \rightarrow f(x)$  untuk KNN.



**Gambar 2. Seluruh Alur Kerja Metode KNN Yang Diperbarui Ditampilkan.**

**Langkah 3:** Buat jaring untuk vektor masukan  $x$ . Keakuratan variabel harus dijamin oleh kerapatan mesh. Selain itu, untuk menghindari hasil yang tidak jelas, mesh tidak boleh menyediakan grup titik yang berbagi  $K$  neighbors terdekat yang sama. Domain variabel juga harus ditetapkan di awal. Meningkatkan kepadatan mesh akan membantu jika loop algoritma berakhir terlalu cepat tanpa menghasilkan hasil yang memuaskan.

**Langkah 4:** Masukkan mesh dari Langkah 3 ke model dari Langkah 2, lalu gunakan regresi KNN untuk memprediksi setiap nilai pada mesh. tergantung pada permintaan untuk nilai ekstrim, tentukan proyeksi maksimum atau minimum. Masukkan nilai terkait vektor, ekspor ke pemecah untuk jawaban yang tepat, lalu impor kembali (baik secara analitik melalui MATLAB atau secara numerik melalui pemecah seperti HFSS). Dataset awal harus diperbarui dengan menambahkan vektor  $x$  dan solusi tepat.



**Langkah 5:** Jika solusi eksak sama dengan catatan terakhir dalam kesalahan yang diizinkan dalam iterasi  $ke^{-n}$ , sebagai  $|f(xn) - f(xn-1)| < |ef(xn)|$ , hentikan perulangan, karena dalam hal ini algoritma tidak dapat memprediksi nilai yang lebih baik. Jika solusi eksak tidak sama dengan *record* terakhir atau jika ini adalah iterasi pertama, ulangi langkah 1 sampai 4. Dalam kasus loop tak terbatas, kami menetapkan penghitung untuk menghentikan loop pada iterasi ke-500, tetapi loop selalu berhenti sebelum angka maksimum ini dalam percobaan kami sejauh ini.

Teknik simulasi numerik yang mahal yang digunakan pada langkah 4 dari pendekatan ini membuat langkah 4 membutuhkan waktu paling lama untuk diselesaikan. Dengan kata lain, lamanya waktu yang diperlukan untuk menjalankan alat simulasi memiliki dampak yang signifikan terhadap seberapa efektif teknik pengoptimalan. Penurunan ukuran sampel untuk penelitian ini juga disebabkan oleh hal ini.

#### 4. KESIMPULAN

Untuk membangun dan mengoptimalkan antenna dengan jumlah sampel data pelatihan dan pengujian yang terbatas, metode KNN baru yang efisien disarankan dalam penelitian ini. Model dapat belajar sendiri dan memprediksi nilai optimal menggunakan beberapa alat simulasi sejumlah kecil sampel yang diperlukan untuk kumpulan data dan beberapa pengetahuan awal sebelumnya untuk membatasi domain target.

Karena sampel data yang lebih berharga dapat dihasilkan selama fase pelatihan, teknik yang disarankan memiliki efisiensi yang sangat tinggi. Penelitian di masa depan bisa berkonsentrasi pada pengoptimalan banyak target, karena secara umum semakin banyak parameter dan target yang kami coba optimalkan, semakin banyak sampel data yang kami perlukan. Dengan penggunaan sampel data pelatihan dan pengujian kecil, lebih banyak masalah desain antenna yang harus diselesaikan.

#### 5. SARAN

Untuk mengoptimalkan antenna dengan sampel data terbatas, disarankan untuk menggunakan metode KNN yang efisien dengan pendekatan pembelajaran mandiri. Manfaatkan alat simulasi khusus untuk menghasilkan data tambahan yang berkualitas, serta teknik augmentasi data untuk meningkatkan variasi sampel. Penerapan teknik active learning dapat meningkatkan efisiensi selama fase pelatihan dengan memilih sampel data yang paling informatif. Fokuskan juga penelitian pada optimasi multi-target menggunakan algoritma seperti Pareto Front untuk mengatasi kebutuhan sampel data yang lebih banyak. Kolaborasi dengan pakar domain dan eksperimen berkelanjutan akan memastikan hasil yang optimal dalam berbagai skenario desain antenna.

## DAFTAR PUSTAKA

- [1] HM El Misilmani, T. Naous, dan SK Al Khatib, “Tinjauan tentang desain dan optimalisasi antena menggunakan algoritme dan teknik machine learning,” *Int. J.RF Microw. Komputer. Eng.* , vol. 30, tidak. 10, hal. e22356, 2020.
- [2] J. Wang, Q. Gao, X. Ma, Y. Zhao, dan Y. Fang, “Belajar untuk merasakan: Pembelajaran mendalam untuk penginderaan nirkabel dengan upaya pelatihan yang lebih sedikit,” *IEEE Wirel. Komunal.* , vol. 27, tidak. 3, hlm. 156–162, 2020.
- [3] U. Rahardja, Q. Aini, MDA Ngadi, M. Hardini, dan FP Oganda, “The Blockchain Manifesto,” pada tahun 2020 *2nd International Conference on Cybernetics and Intelligent System (ICORIS)* , 2020, hlm. 1–5.
- [4] H. Sakdiyah, Q. Aini, dan T. Yulianto, “Pengenalan Karakter Seseorang Melalui Bentuk Tanda Tangan Menggunakan Metode Radial Basis Function Neural Network (RBFNN),” dalam *Journal of Physics: Conference Series* , 2020, vol . 1538, no. 1, hal. 12059.
- [5] L. Yuan, X.-S. Yang, C. Wang, dan B.-Z. Wang, "Pemodelan jaringan saraf tiruan multicabang untuk estimasi terbalik dari directivity array antena," *IEEE Trans. Propag Antena.* , vol. 68, tidak. 6, hlm. 4417–4427, 2020.
- [6] L.-Y. Xiao, W. Shao, F.-L. Jin, dan B.-Z. Wang, “Pemodelan multiparameter dengan JST untuk desain antena,” *IEEE Trans. Propag Antena.* , vol. 66, tidak. 7, hlm. 3718–3723, 2018.
- [7] V. Arya dkk. , “Satu penjelasan tidak cocok untuk semua: Perangkat dan taksonomi teknik penjelasan ai,” *arXiv Prepr. arXiv1909.03012* , 2019.
- [8] R. Sweke, MS Kesselring, EPL van Nieuwenburg, dan J. Eisert, “Reinforcement learning decoders for fault-tolerant kuantum komputasi,” *Mach. Mempelajari. Sains. Technol.* , vol. 2, tidak. 2, hal. 25005, 2020.
- [9] N. Dvornik, J. Mairal, dan C. Schmid, “Pemodelan konteks visual adalah kunci untuk menambah kumpulan data deteksi objek,” dalam *Prosiding Konferensi Eropa tentang Visi Komputer (ECCV)* , 2018, hlm. 364–380.
- [10] T. Zhang dkk. , "Peramalan lalu lintas berbasis jaringan saraf grafik untuk internet kendaraan yang terhubung," *IEEE Trans. Jaringan Sains. Eng.* , vol. 9, tidak. 5, hlm. 3015–3027, 2021.
- [11] Y. Wang, Z. Pan, dan Y. Pan, “Metode Pembersihan Kumpulan Data Pelatihan dengan Peringkat Kemampuan Klasifikasi untuk  $k$ -Nearest Neighbor Classifier,” *IEEE Trans. jaringan saraf Pelajari. Sistem.* , vol. 31, tidak. 5, hlm. 1544–1556, 2019.
- [12] W. Zhang, J. Gao, X. Jiang, dan W. Sun, “Migrasi Waktu Terbalik Kuadrat Terkecil yang Konsisten Menggunakan Jaringan Syaraf Konvolusional,” *IEEE Trans. Geosci. Penginderaan Jauh* , vol. 60, hlm. 1–18, 2021.
- [13] G. Bhat, M. Danelljan, L. Van Gool, dan R. Timofte, “Mempelajari prediksi model diskriminatif untuk pelacakan,” dalam *Prosiding konferensi internasional IEEE/CVF tentang visi komputer* , 2019, hlm. 6182–6191.
- [14] PP Angelov dan X. Gu, "Menuju machine learning antropomorfik," *Komputer (Long. Beach. California).* , vol. 51, tidak. 9, hlm. 18–27, 2018.
- [15] E. GEDIK, “DETEKSI TARGET MELALUI DINDING MENGGUNAKAN FITUR TEKSTUR.” UNIVERSITAS TEKNIS TIMUR TENGAH, 2019.
- [16] R. Jia dkk. , “Penilaian data khusus tugas yang efisien untuk algoritme tetangga terdekat,” *arXiv Prepr. arXiv1908.08619* , 2019.
- [17] I. Lang, A. Manor, dan S. Avidan, "Samplenet: Pengambilan sampel awan titik yang dapat dibedakan," dalam *Prosiding Konferensi IEEE/CVF tentang Visi Komputer dan Pengenalan Pola* , 2020, hlm. 7578–7588.
- [18] S. Koziel, N. Çalık, P. Mahouti, dan MA Belen, “Pemodelan struktur antena yang akurat melalui pengurangan domain dan jaringan saraf dalam piramidal,” *IEEE Trans. Propag Antena.* , vol. 70, tidak. 3, hlm. 2174–2188, 2021.

- [19] A. Barreto *dkk.*, “Transfer dalam pembelajaran penguatan mendalam menggunakan fitur penerus dan peningkatan kebijakan umum,” dalam *Konferensi Internasional tentang machine learning*, 2018, hlm. 501–510.
- [20] DW Berger, KF Herkenhoff, dan S. Mongey, “Sebuah model penyakit menular seir dengan pengujian dan karantina bersyarat,” Biro Riset Ekonomi Nasional, 2020.