

Ekstraksi Fitur Menggunakan Scale Invariant Feature Transform untuk Klasifikasi Jenis Sampah

1st Ernawati
Program Studi Informatika,
Universitas Bengkulu
Bengkulu, Indonesia
ernawati@unib.ac.id

2nd Desi Andreswari
Program Studi Informatika,
Universitas Bengkulu
Bengkulu, Indonesia
desi.andreswari@unib.ac.id

3rd Aan Erlanshari
Program Studi Sistem Informasi,
Universitas Bengkulu
Bengkulu, Indonesia
aan_erlanshari@unib.ac.id

4th Funny Farady Coastera
Program Studi Informatika,
Universitas Bengkulu
Bengkulu, Indonesia
ffaradyc@unib.ac.id

Abstract—Waste includes all materials generated as a result of human and societal activities but are no longer functional or have been discarded. In waste management, the community needs to sort waste according to its type and nature, making it easier to recycle. The waste sorting process can be hazardous for sanitation workers and the community. AI-based waste sorting technology is now capable of recognizing waste types based on images. To achieve image recognition, a feature extraction process using the SIFT method is carried out, which captures specific features of waste objects that describe their characteristics. The extracted features are then used as input in the classification process with the K-NN method, which identifies the waste based on its type. This study achieved an accuracy of 56.7%.⁸

Abstrak—Sampah mencakup semua material yang dihasilkan sebagai akibat dari aktivitas manusia dan masyarakat namun telah tidak lagi berfungsi atau sudah tidak digunakan lagi. Dalam pengelolaan sampah, masyarakat perlu memilah sampah sesuai dengan jenis dan sifatnya, sehingga lebih mudah untuk didaur ulang. Proses pemilahan sampah menjadi aktivitas yang berbahaya bagi petugas kebersihan dan masyarakat. Pemilahan sampah berbasis teknologi AI saat ini mampu mengenali jenis sampah berdasarkan citra. Untuk menghasilkan pengenalan citra, perlu dilakukan proses ekstraksi fitur dengan metode SIFT, mengambil fitur khusus dari objek sampah yang dapat menggambarkan karakteristik objek tersebut. Fitur yang diperoleh akan menjadi input pada proses klasifikasi dengan metode K-NN yang akan mengenali sampah berdasarkan jenisnya. Penelitian ini menghasilkan akurasi 56,7%.

Keywords— *Sampah, SIFT, ekstraksi, klasifikasi, K-NN*

I. PENDAHULUAN

Sampah mencakup semua material yang dihasilkan sebagai akibat dari aktivitas manusia dan masyarakat namun telah tidak lagi berfungsi atau sudah tidak digunakan lagi [1]. Sampah dapat berupa barang yang termasuk dalam tiga kelompok utama: (i) bahan anorganik yang dapat didaur ulang dan cocok untuk digunakan kembali, yaitu plastik, logam; (ii) bahan organik yang dapat dialihkan dari mana energi dan

pupuk dapat dihasilkan, yaitu makanan dan vegetasi; dan (iii) bahan anorganik yang memerlukan tempat pembuangan akhir, yaitu keramik, kayu yang telah diolah [2].

Dalam pengelolaan sampah, masyarakat perlu memilah sampah sesuai dengan jenis dan sifatnya, sehingga lebih mudah untuk didaur ulang. Proses pemilahan sampah menjadi aktivitas yang berbahaya bagi petugas kebersihan dan masyarakat. Terlebih lagi, jika sampah sudah membusuk selama berhari-hari sehingga menjadi sarang kuman dan penyakit. Pengelolaan sampah yang tidak tepat, akan menyebabkan pencemaran lingkungan yang dapat mempengaruhi kesehatan masyarakat [3].

Pengembangan teknologi kecerdasan buatan saat ini, telah mampu mengenali jenis sampah berdasarkan citranya [3]. Untuk menghasilkan pengenalan citra, perlu dilakukan proses ekstraksi fitur dengan mengambil fitur khusus dari objek sampah yang dapat menggambarkan karakteristik objek tersebut [3]. Fitur yang diperoleh akan menjadi input pada proses klasifikasi yang akan mengenali sampah berdasarkan jenisnya. Ekstraksi fitur merujuk pada penelitian tentang SIFT yang telah dilakukan [4][5][6][7][8], dapat disimpulkan bahwa penggunaan algoritma SIFT memberikan hasil yang baik. Algoritma SIFT sangat baik karena mampu mendeteksi dan menggambarkan fungsi keypoint dan deskriptor dari citra sampah yang tidak terpengaruh oleh noise, perspektif, skala, rotasi, translasi, dan pencahayaan.

K-nearest neighborhood (K-NN) adalah salah satu algoritma klasifikasi populer yang dikenal bekerja dengan baik untuk mengklasifikasikan data [9][10][11]. Konsep dan mekanisme K-NN sederhana, ketika diberikan data uji yang tidak berlabel, K-NN akan mencari k tetangga terdekat dalam dataset pelatihan berdasarkan jarak yang telah ditentukan, lalu menetapkan label kelas yang paling umum diantara tetangga terpilih [9].

II. METODE

A. Akusisi Data

Penelitian diawali dengan mengumpulkan dataset sampah yang diperoleh dari <https://archive.ics.uci.edu/dataset/908/realwaste>. Dataset ini dalam bentuk RGB dan berukuran 512x512. Pada penelitian ini dataset terdiri dari 51 data dan 3 kelas, yaitu glass, metal dan textile.

B. Metode Penelitian

1. Scale Invariant Feature Transform (SIFT)

Metode SIFT telah diterapkan dalam bidang computer vision secara luas untuk deteksi dan pengenalan, seperti temu kembali citra, pengenalan objek pada citra, deteksi objek pada citra. Fitur SIFT telah diterapkan di hampir semua bidang visi komputer yang terkait dengan pengenalan dan deteksi, seperti mozaik citra, pencarian citra, pengenalan target, deteksi target, pelacakan target, augmented reality, rekonstruksi tiga dimensi, dan lain-lain. Ini adalah salah satu algoritma visi komputer paling terkenal sebelum era deep learning [12].

Fitur adalah karakteristik unik suatu citra. Ekstraksi fitur merupakan hal pertama pada tahap pencocokan citra, dimana fitur yang diekstraksi berperan penting dalam pencocokan dengan citra referensi. Setelah itu, ekstraksi fitur yang membuat masing-masing keypoint pada area sekitarnya yang harus tahan terhadap perubahan rotasi dan skala. Metode SIFT mempunyai 4 langkah berikut [6] :

- Deteksi ruang skala extrema: mencari ruang skala menggunakan fungsi Difference of Gaussian (DoG) untuk mengidentifikasi titik-titik yang diinginkan serta invariant terhadap skala dan orientasi. Skala runag dari citra didefinisikan sebagai fungsi $L(x, y, \sigma)$ pada Persamaan 1 yang dihasilkan oleh konvolusi fungsi Gaussian $G(x, y, \sigma)$ pada persamaan 2 dengan input citra $I(x, y)$.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Secara efisien untuk mendeteksi lokasi keypoint yang stabil, fungsi DoG yang dikonvolusi dengan citra, menggunakan persamaan 3 yang dapat dihitung dari perbedaan dua skala terdekat yang dipisahkan oleh faktor penskalaan konstan k dengan citra input.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\pi) - L(x, y, \sigma) \quad (3)$$

- Lokalisasi keypoint : lokasi dan skala setiap titik kandidat ditentukan dan titik fitur dipilih berdasarkan ukuran stabilitas dengan membandingkan dengan 3x3x3 tetangga pada level yang sama, level diatas dan dibawahnya.
- Penentuan orientasi: satu orientasi atau lebih ditempatkan ke masing-masing lokasi titik fitur berdasarkan arah gradient citra lokal. Untuk mencapai invariant rotasi, masing-masing keypoint diberi orientasi. Untuk setiap citra input pada skala $L(x, y)$, besarnya gradient magnitude $m(x, y)$ pada persamaan 4 dan orientasi $\theta(x, y)$ pada persamaan 5 dihitung dengan selisih piksel.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L_y L(x, y+1) - L(x, y-1)}{L(x, y+1) - L(x, y-1)} \right) \quad (5)$$

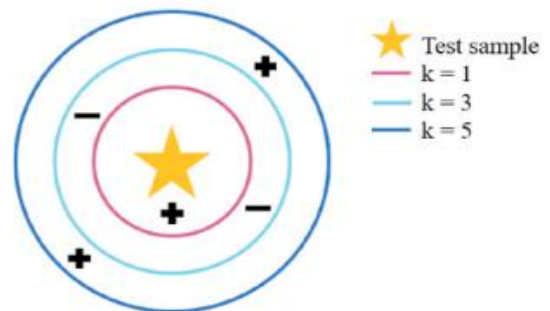
- Deskriptor keypoint : deskriptor fitur dibuat dengan terlebih dahulu menghitung besarnya gradient dan orientasi pada setiap titik citra di sekitar wilayah lokasi titik fitur. Citra ini kemudian diakumulasikan ke orientasi histogram dan dibagi menjadi 4x4 sub-region, dengan orientasi 8-bin.
 - Jadi setiap fitur memiliki 128 elemen fitur [6].
- #### 2. K-Nearest Neighbors (KNN)

Sistem klasifikasi menggunakan algoritma K-NN. Algoritma ini banyak digunakan dalam *machine learning* dan termasuk dalam jenis *supervised learning*. Metode ini menggunakan model ruang vektor untuk klasifikasi. Konsep utamanya adalah bahwa karakteristik fitur dari kelas yang sama memiliki kesamaan yang lebih tinggi. Oleh karena itu, kesamaan terhadap karakteristik fitur dari kelas yang dikenal dihitung untuk memprediksi kemungkinan fitur dari kelas yang tidak diketahui. Algoritma K-NN menghitung jarak antara titik data yang berbeda untuk mengklasifikasikan titik data baru. Kelas yang terdiagnosis untuk setiap sampel uji adalah kelas yang sesuai dengan k titik terdekat [13].

Jika himpunan pelatihan adalah $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, x_i mengacu pada vektor fitur dan y_i adalah label klasifikasi data, $y_i = C_1, C_2, \dots, C_k, i = 1, 2 \dots n$. Untuk sampel pelatihan (x, y) , algoritma K-NN akan menemukan k fitur karakteristik terdekat dengan x sesuai dengan fungsi jarak yang diberikan. $N_k(x)$ mewakili lingkungan dari k fitur terdekat dan label klasifikasi sampel uji x ditentukan berdasarkan aturan keputusan. Algoritma K-NN dapat ditunjukkan dalam persamaan (6) :

$$y = \operatorname{argmax}_{C_j} \sum_{x_i \in N_k(x)} I(y_i = C_j) \quad (6)$$

Dimana $i = 1, 2, \dots, N$ dan $j = 1, 2, \dots, K; I(x)$ adalah fungsi indikator. Setelah melabeli sampel pelatihan, kelas dari sampel uji dapat ditentukan dengan mengamati klasifikasi algoritma K-NN. Metode ini ditunjukkan pada Gambar 1. Garis tebal mewakili garis dengan jarak yang sama. Dimana $k = 1$ atau $k = 5$, sampel uji diklasifikasikan sebagai sampel positif karena ketika $k = 5$, terdapat tiga sampel positif yang lebih banyak dari dua sampel negatif dalam lingkaran dengan jarak yang sama. Demikian pula, ketika $k = 3$, sampel diklasifikasikan sebagai sampel negatif karena terdapat dua sampel negatif yang lebih banyak dari satu sampel positif dalam lingkaran merah. Aturan klasifikasi keputusan seperti ini dikenal sebagai keputusan mayoritas.



Gambar 1. Dasar-dasar klasifikasi dari K-NN algoritma

Selain nilai k , fungsi jarak juga sangat penting untuk akurasi prediksi sistem. Misalkan x_s dan y_t mewakili n variabel dari x dan y dalam ruang vektor 2D, misal $x_s = (x_1, x_2, x_3, \dots, x_n)$, $y_t = (y_1, y_2, y_3, \dots, y_n)$.

Diberikan x_s dan y_t merepresentasikan n variabel dari x dan y dalam ruang vektor 2D i.e., $x_s = (x_1, x_2, x_3, \dots, x_n)$, $y_t = (y_1, y_2, y_3, \dots, y_n)$. Fungsi jarak euclidean antara vektor didefinisikan sebagai berikut:

Titik data diklasifikasikan berdasarkan tetangga di ruang eigen. Dalam penelitian ini, perbedaan kuadrat dari piksel dihitung terlebih dahulu, kemudian akar dari jumlah piksel tersebut diambil. Jarak Euclidean digunakan untuk menghitung jarak antara dua titik dalam sistem koordinat, yang dinyatakan sebagai persamaan (7):

$$d_{st}^2 = \sum_{i=1}^n (x_i - y_i)^2 \quad (7)$$

d_{st} adalah jarak antara titik s dan titik t , n adalah jumlah elemen vektor, x_i adalah elemen ke- i pada ruang vektor x dan y_i adalah elemen ke- i pada ruang vektor y [13].

3. Evaluasi Sistem

Evaluasi sistem dilakukan dengan mengukur tingkat akurasi, *precision*, *recall* dan *F1-score*. Tingkat akurasi dapat dikalkulasikan dengan persamaan (8):

$$\text{Tingkat Akurasi} = \frac{\sum \text{Tingkat Akurasi}}{\sum \text{Total Data Uji}} \times 100\% \quad (8)$$

Precision (P) dan *Recall* (R) dihitung dengan persamaan (9) dan (10) [14].

$$P = \frac{tp}{N_{TR}} = \frac{tp}{tp+fp} \quad (9)$$

tp menyatakan citra yang relevan yang diambil dan fp menyatakan citra yang diambil tetapi tidak relevan. *Recall* (R) dinyatakan sebagai rasion citra relevan yang diambil dengan banyaknya citra yang relevan dalam basis data.

$$R = \frac{tp}{N_{RI}} = \frac{tp}{tp+fn} \quad (10)$$

tp menyatakan citra yang relevan yang diambil dan N_{RI} menyatakan citra yang relevan di basis data. N_{RI} didapatkan dari harga $tp + fn$, fn menyatakan banyaknya citra yang sebenarnya termasuk dalam kelompok yang relevan tetapi tidak diambil.

F1-Score digunakan untuk mengukur rerata harmonik P dan R , dinyatakan dengan persamaan (11). Harga *F-Score* menandakan prediksi yang lebih baik. P dan R adalah harga dari *precision* dan *recall*.

$$F = 2 \frac{P \cdot R}{P + R} \quad (11)$$

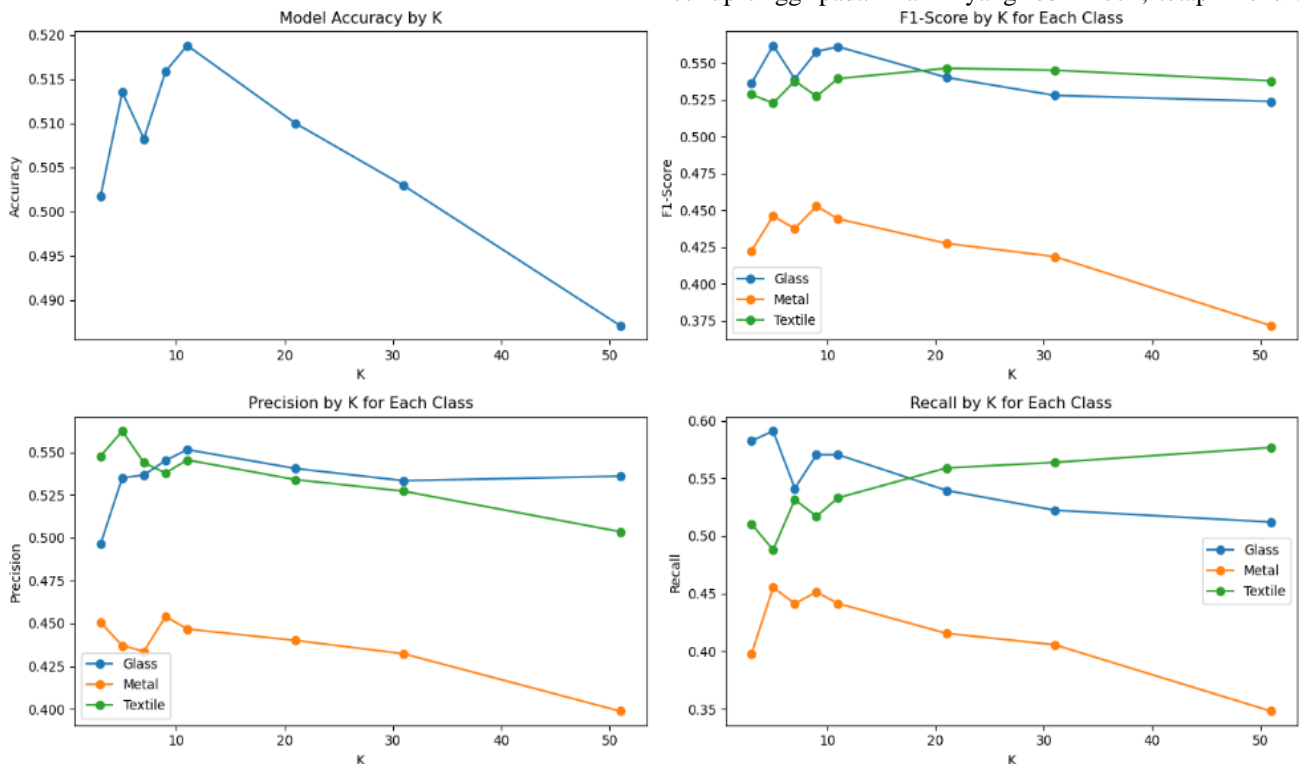
Harga *precision*, *recall* dan *F-Measure* yang tinggi (mendekati 1.0) menunjukkan sistem bekerja dengan efektif [14].

III. HASIL DAN PEMBAHASAN

Grafik performa sistem dapat dilihat pada Gambar 2 dengan mengukur akurasi, *precision*, *recall* dan *f1-score* pada berbagai nilai k untuk setiap kelas.

Akurasi meningkat pada titik tertentu, mencapai puncak pada $K=10$ dan kemudian menurun secara signifikan seiring dengan peningkatan nilai K . Hal ini memperlihatkan bahwa dengan nilai yang lebih kecil, sistem dapat lebih fokus pada tetangga terdekatnya yang relevan. Namun, dengan peningkatan nilai K , sistem mulai memperhitungkan lebih banyak tetangga yang kurang relevan, yang dapat menyebabkan penurunan akurasi.

F1-score pada kelas *glass* dan *textile* menunjukkan performa yang relative stabil dengan nilai *F1-score* yang cukup tinggi pada nilai K yang lebih kecil, tetapi menurun



Gambar 2. Grafik hasil dari performa model dengan berbagai nilai K

perlahan dengan meningkatnya nilai K. Pada kelas Metal memiliki F1-score yang lebih rendah dibandingkan dengan kelas lainnya dan menurun tajam setelah K=20. Hal ini memperlihatkan bahwa F1-score yang lebih tinggi pada nilai K kecil menunjukkan keseimbangan yang baik antara precision dan recall. Penurunan F1-score untuk K besar menunjukkan bahwa model mengalami kesulitan mempertahankan keseimbangan. Precision pada kelas glass dan textile mempertahankan precision yang stabil atau sedikit meningkat pada K kecil hingga menengah. Pada kelas metal memiliki precision yang lebih rendah dan terus menurun saat nilai K meningkat. Precision yang lebih stabil untuk kelas glass dan textile pada K kecil menunjukkan bahwa sistem cukup baik dalam meminimalkan prediksi citra yang terambil tetapi tidak relevan (*fp*). Penurunan precision untuk “metal” pada nilai K yang besar menunjukkan model mulai tidak relevan (*fp*). Recall pada kelas glass dan textil memiliki recall yang lebih stabil pada nilai K kecil hingga menengah. Kelas metal menunjukkan penurunan yang lebih signifikan dalam recall seiring dengan peningkatan nilai K. Hal ini menunjukkan recall yang stabil untuk glass dan textil menunjukkan bahwa sistem tetap mampu mendeteksi sebagian besar contoh citra yang relevan. Namun, penurunan recall untuk metal mengindikasikan bahwa sistem kehilangan lebih banyak contoh citra relevan dengan meningkatnya nilai K.

IV. KESIMPULAN

Nilai K optimal berada disekitar K=10, dimana sistem menunjukkan kinerja terbaik dalam hal akurasi, *F1-score*, *precision* dan *recall*. Kelas metal lebih sensitif terhadap perubahan nilai K dibandingkan dengan glass dan textil, dengan penurunan yang lebih cepat saat K meningkat. Overfitting dan underfitting menjadi masalah dengan nilai K yang terlalu kecil atau terlalu besar. Nilai K yang kecil dapat menyebabkan overfitting, sedangkan nilai K yang besar dapat menyebabkan underfitting.

REFERENCES

- [1] C. S. Burke, E. Salas, K. Smith-Jentsch, dan M. A. Rosen, “Measuring macrocognition in teams: Some insights for navigating the complexities,” *Macrocognition Metrics Scenar. Des. Eval. Real-World Teams*, hal. 29–43, 2012, doi: 10.1201/9781315593173-4.
- [2] S. Single, S. Iranmanesh, dan R. Raad, “RealWaste: A Novel Real-Life Data Set for Landfill Waste Classification Using Deep Learning,” *Inf.*, vol. 14, no. 12, 2023, doi: 10.3390/info14120633.
- [3] A. P. Puspaningrum dkk., “Waste Classification Using Support Vector Machine with SIFT-PCA Feature Extraction,” *ICICoS 2020 - Proceeding 4th Int. Conf. Informatics Comput. Sci.*, hal. 4–9, 2020, doi: 10.1109/ICICoS51170.2020.9298982.
- [4] B. Yang, “Sports Image Classification and Application based on SIFT Algorithm,” *2nd IEEE Int. Conf. Adv. Technol. Intell. Control. Environ. Comput. Commun. Eng. ICATIECE 2022*, hal. 1–5, 2022, doi: 10.1109/ICATIECE56365.2022.10047818.
- [5] R. M. Akbar dan N. Sunarmi, “Pencocokan Citra Untuk Pengenalan Produk Belanja Menggunakan Sift (Scale-Invariant Feature Transform),” *Pros. SENTRA (Seminar ...)*, hal. 77–84, 2019, [Daring]. Tersedia pada: <http://research-report.umm.ac.id/index.php/sentra/article/view/2282>
- [6] K. Song, F. Zhu, dan L. Song, “Moving Target Detection Algorithm Based on SIFT Feature Matching,” *Proc. - 2022 Int. Conf. Front. Artif. Intell. Mach. Learn. FAIML 2022*, hal. 196–199, 2022, doi: 10.1109/FAIML57028.2022.00045.
- [7] R. Mukesh dan P. Dass, “Detection by Clinicians Comparison of Intelligence Bone Fracture Detection System with SIFT algorithm for Identification of Bone Fracture,” *Proc. 3rd Int. Conf. Intell. Eng. Manag. ICIEM 2022*, hal. 533–537, 2022, doi: 10.1109/ICIEM54221.2022.9853197.
- [8] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, hal. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [9] A. X. Wang, S. S. Chukova, dan B. P. Nguyen, “Ensemble k-nearest neighbors based on centroid displacement,” *Inf. Sci. (Ny.)*, vol. 629, no. January, hal. 313–323, 2023, doi: 10.1016/j.ins.2023.02.004.
- [10] L. Li, X. Chen, dan C. Song, “A robust clustering method with noise identification based on directed K-nearest neighbor graph,” *Neurocomputing*, vol. 508, hal. 19–35, 2022, doi: 10.1016/j.neucom.2022.08.029.
- [11] J. Yang, X. Tan, dan S. Rahardja, “Outlier detection: How to Select k for k-nearest-neighbors-based outlier detectors,” *Pattern Recognit. Lett.*, vol. 174, no. August, hal. 112–117, 2023, doi: 10.1016/j.patrec.2023.08.020.
- [12] Z. Wu dan H. Wu, “Improved SIFT image feature matching algorithm,” *Proc. - 2022 2nd Int. Conf. Comput. Graph. Image Virtualization, ICCGIV 2022*, hal. 223–226, 2022, doi: 10.1109/ICCGIV57403.2022.00051.
- [13] Y. Tang, Y. Chang, dan K. Li, “Applications of K-nearest neighbor algorithm in intelligent diagnosis of wind turbine blades damage,” *Renew. Energy*, vol. 212, no. 2, hal. 855–864, 2023, doi: 10.1016/j.renene.2023.05.087.
- [14] Eri Zuliarso, Sulastri, dan Yunus Anis, “Temu Kembali Berbasis Citra untuk Menemukan Kemiripan Merek Menggunakan Algoritma SIFT dan SURF,” *J. Buana Inform.*, vol. 13, no. 02, hal. 116–125, 2022, doi: 10.24002/jbi.v13i02.6328.