

Klasifikasi Cuaca Menggunakan Metode VGG-16, Principal Component Analysis Dan K-Nearest Neighbor

Sean Alexander Suryaman¹, Rita Magdalena², Sofia Sa'idah³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Indonesia
Email: ¹seanalexanders@student.telkomuniversity.ac.id, ²ritamagdalenat@telkomuniversity.ac.id,
³sofiasaidahsfi@telkomuniversity.ac.id

Abstrak

Cuaca merupakan suatu fenomena alam yang sangat berdampak bagi manusia. Informasi tentang kondisi cuaca sangat dibutuhkan oleh manusia. Informasi ini sangat bermanfaat untuk mengetahui kejadian cuaca disekitar kita. Sistem klasifikasi saat ini mengandalkan serangkaian sensor mahal atau bantuan manusia. Kecerdasan buatan merupakan suatu cabang ilmu komputer yang membantu manusia dalam mengatasi masalah yang ada. Penelitian ini menggunakan kecerdasan buatan untuk mengklasifikasi kondisi cuaca dengan menggunakan metode VGG-16, *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (KNN). Pertama ciri akan dicari menggunakan VGG-16, lalu memanfaatkan *Principal Component Analysis* (PCA) untuk mereduksi data agar lebih efektif. Dan menggunakan *K-Nearest Neighbor* (KNN) untuk mengklasifikasikan data. *K-Nearest Neighbor* (KNN) menggunakan jarak untuk mengklasifikasikan data. Jarak yang dipilih merupakan jarak terpendek yang akan menunjukkan ketetanggaan untuk menghasilkan keluaran apakah cuaca sedang cerah, berawan, berkabut, hujan dan matahari terbit. Sistem tersebut dibuat menggunakan platform Google Colab dengan bahasa pemrograman Python. Berdasarkan hasil penelitian, diperoleh sistem klasifikasi cuaca dengan akurasi sebesar 87,50%. Hasil akurasi tersebut diperoleh ketika digunakan 450 data uji dan 1050 data latih. Adapun parameter terbaik yang dihasilkan, yaitu ukuran citra 256 x 256, jenis KNN adalah *Cosine*, nilai KNN di $k = 9$, dan Persentase PCA 30%.

Kata kunci: *k-nearest neighbor, klasifikasi cuaca, principal component analysis, VGG-16*

Abstract

Weather is a natural phenomenon that has an impact on humans. Information about weather conditions is needed by humans. This information is very useful to know the weather events around us. Current classification systems rely on expensive sensors or human assistance. Artificial intelligence is a branch of computer science that helps humans in overcoming existing problems. This study uses artificial intelligence to classify weather conditions using the VGG-16, Principal Component Analysis (PCA) and K-Nearest Neighbor (KNN) methods. First, we find the feature using VGG-16, then use Principal Component Analysis (PCA) to reduce data to be more effective. And using K-Nearest Neighbor (KNN) to classify the data. K-Nearest Neighbor (KNN) uses distance to classify data. The selected distance is the shortest distance that will show neighbors to produce output whether the weather is sunny, cloudy, foggy, rainy and sunrise. The system was created using the Google Colab platform with the Python programming language. Based on the results of the study, obtained a weather classification system with an accuracy of 87.50%. The accuracy results were obtained when 450 test data and 1050 training data were used. The best parameters produced are image size 256 x 256, the type of KNN is Cosine, the value of KNN is at $k = 9$, and the PCA percentage is 30%.

Keywords: *k-nearest neighbor, principal component analysis, VGG-16weather classification*

1. PENDAHULUAN

Semakin berkembang peradaban manusia, kemajuan teknologi pun semakin berkembang. Dengan kemajuan teknologi yang semakin pesat ini, pekerjaan-pekerjaan manusia semakin dimudahkan. Kemajuan teknologi di bidang Informasi yang sangat pesat belakangan ini, membantu kita dalam melakukan kegiatan sehari-hari. Salah satu informasi yang kita butuhkan adalah tentang cuaca. Keadaan cuaca yang tidak menentu belakangan ini membuat kita khawatir mengenai keselamatan. Cuaca yang

ekstrim dapat menyebabkan bencana alam yang menyebabkan kerugian material dan menghilangkan nyawa manusia. Selain itu, informasi tentang kondisi cuaca juga bermanfaat dalam menjalani kehidupan. Oleh karena itu, dibutuhkan suatu alat untuk memberikan informasi mengenai cuaca.

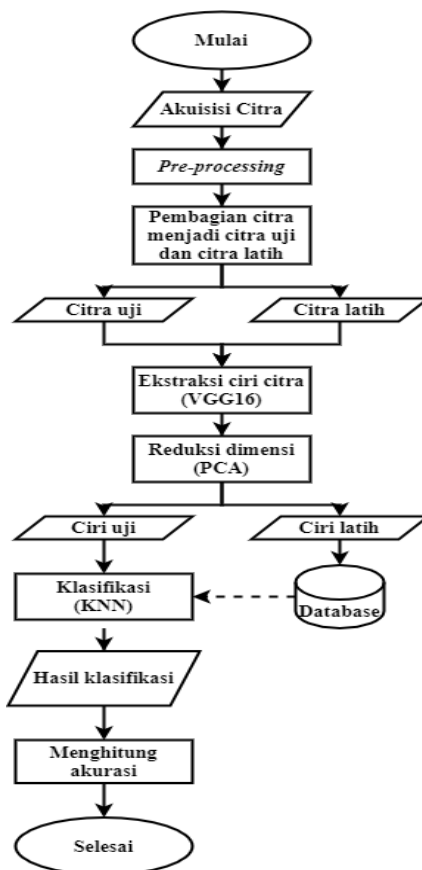
Citra atau gambar merupakan salah satu media digunakan dalam teknologi modern. Contoh pengaplikasian citra ada pada sistem pendeteksi hujan es, sistem pendeteksi asap, debu vulkanik dan lain sebagainya. Alat klasifikasi cuaca yang ada saat ini menggunakan sensor kelembapan memiliki beberapa kekurangan yakni dari segi sistem serta biaya yang dikeluarkan. Karena itu pemanfaatan citra dalam klasifikasi cuaca dapat menjadi alternatif untuk sistem klasifikasi cuaca yang lebih efektif dan efisien.

Terdapat banyak penelitian yang memanfaatkan citra untuk sistem pendeteksi. Penelitian [1] penerapan algoritma klasifikasi data mining C4.5 pada dataset cuaca wilayah Bekasi menghasilkan akurasi sebesar 88,89%. Penelitian [2] analisis perbandingan algoritma SVM, KNN dan CNN untuk klasifikasi citra cuaca mendapatkan hasil bahwa algoritma CNN memiliki performa terbaik dengan akurasi 94,2%. Penelitian [3] Identifikasi kualitas beras dengan citra digital menggunakan metode *k-fold cross validation* dengan $k=5$ mendapatkan akurasi sebesar 96,67%. Penelitian [4] *PCA model for RNA-seq malaria vector data classification using KNN and decision tree algorithm* menghasilkan akurasi 86,7%. Penelitian [5] *comparing image classification methods: K-Nearest-Neighbor and Support-Vector-Machines* mendapatkan hasil bahwa metode SVM lebih baik dari metode KNN.

Melalui permasalahan yang ada beserta penelitian-penelitian sebelumnya, penulis melakukan penelitian klasifikasi cuaca menggunakan metode *Principal Component Analysis* (PCA) dan metode klasifikasi *K-Nearest Neighbor* (KNN). Penelitian tersebut menghasilkan akurasi sebesar 87,50%.

2. METODE PENELITIAN

Metode yang dilakukan pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Diagram Alir

Langkah pertama yang akan dilakukan adalah akuisisi citra. Akuisisi citra merupakan proses pengumpulan citra cuaca yang akan digunakan dalam penelitian ini. Setelah citra kita dapatkan maka langkah selanjutnya adalah *pre-processing*. Langkah *pre-processing* dilakukan karena citra yang didapatkan tidak memiliki ukuran yang sama oleh karena itu perlu dilakukan penyeragaman ukuran. Setelah semua citra memiliki ukuran yang sama, maka akan dilakukan pembagian citra menjadi citra latih dan citra uji. Setelah citra dibagi maka akan dilakukan proses ekstraksi ciri menggunakan metode VGG16. Ciri yang didapatkan dari proses ekstraksi ciri tidak semuanya adalah ciri yang optimal karena itu ciri tersebut akan direduksi menggunakan algoritma PCA. Setelah didapatkan ciri yang optimal, maka ciri dari data latih akan menjadi acuan dalam melakukan proses klasifikasi. Proses klasifikasi dilakukan dengan menggunakan metode KNN. Setelah hasil klasifikasi didapatkan, maka akan dilakukan perhitungan akurasi untuk melihat seberapa baik sistem yang telah dibuat.

2.1. Akuisisi Citra

Akuisisi citra merupakan proses menangkap atau memindai suatu citra analog sehingga diperoleh citra digital. Akuisisi citra yang penulis lakukan pada penelitian kali ini menggunakan citra dari Kaggle sehingga didapatkan citra cuaca dalam bentuk digital. Citra tersebut akan diolah dan diproses untuk mengklasifikasikan cuaca yang sedang terjadi. [6]

2.2. Pre-Processing

Pre-processing merupakan tahap awal dalam pengolahan citra. Tahap ini dilakukan pada saat proses pelatihan dan pengujian. *Pre-processing* akan menentukan bagian dari citra yang akan diteliti serta meningkatkan kualitas dari citra masukan yang diperoleh, sehingga pada tahap selanjutnya lebih mudah untuk diolah. Pada penelitian ini *Pre-processing* yang dilakukan adalah proses *resize* dengan tujuan untuk menyeragamkan ukuran citra. [6]

2.3. Ekstraksi Ciri

Ekstraksi ciri (*Feature Extraction*) adalah teknik mendapatkan ciri dari kumpulan data untuk dapat mengenali pola pada data tersebut. Ciri yang telah didapatkan tersebut akan digunakan untuk membedakan satu objek dengan objek lainnya. Citra yang baik memiliki akurasi tinggi, mudah dalam komputasi, dan dapat diperkecil tanpa menghilangkan informasi-informasi penting.[7]

2.4. VGG16

VGG16 merupakan model *convolutional neural network*. VGG16 adalah penyempurnaan dari model AlexNet. Model AlexNet menggunakan ukuran kernel yang besar. Sedangkan VGG16 menggunakan ukuran kernel 3x3. Citra akan melalui tumpukan dari *convolutional layer*, dimana *filter* yang digunakan sangat kecil yaitu 3x3. Arsitektur VGG-16 memiliki 16 *layer* yang terdiri dari 13 *convolutional layer* dan 3 *fully connected layer*. *Network* ini adalah jaringan yang cukup besar dan memiliki sekitar 138 juta parameter.

Asitektur VGG-16 diawali oleh *convolutional layer* lalu dihubungkan dengan *max pooling* serta *fully connected layer* dan terakhir ke lapisan output. Pada lapisan *convolutional layer* dan *fully connected layer* menggunakan fungsi ReLU.[8]

2.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) merupakan algoritma yang digunakan pada pengolahan sinyal digital untuk pengurangan dimensi data dengan cara mengambil ciri penting dari suatu data atau untuk dekomposisi variabel data. PCA dapat digunakan dalam teori deteksi, pengenalan pola dan *image coding*. Keunggulan PCA adalah dapat mereduksi data dengan cara mencari pola dan melakukan kompresi data tanpa harus kehilangan informasi yang penting. Dengan dimensi data yang rendah ini, waktu komputasi dapat dikurangi.[9]

PCA menghasilkan menghasilkan vektor-vektor Eigen atau vektor-vektor karakteristik yang digunakan untuk membuat ruang Eigen. Penentuan vektor eigen diperlukan dalam mereduksi dimensi data. Urutan Vektor Eigen dimulai dari nilai eigen paling besar hingga nilai eigen paling kecil. Vektor Eigen yang dapat direduksi adalah vektor Eigen yang memiliki nilai terkecil. Hal ini mengindikasikan bahwa vektor Eigen yang direduksi dianggap tidak memiliki informasi terlalu penting sehingga data dapat direduksi tanpa kehilangan pola yang penting.[9]

Terdapat lima langkah penting dalam pembentukan PCA, berikut adalah langkah – langkahnya:

1. Normalisasi

Normalisasi dilakukan untuk menyamakan skala pada data. Cara melakukan normalisasi adalah dengan menghitung rata-rata semua dimensi kumpulan data, kecuali labelnya. Skala data sehingga setiap variabel memberikan kontribusi yang sama untuk analisis. Berikut merupakan persamaan dari normalisasi.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Dimana: z = nilai skala
 x = inisial
 μ = mean
 σ = standar deviasi

2. Perhitungan Matriks Kovarian

Matrix kovarian merupakan matrix yang menunjukkan penyebaran data ukuran dimensi lebih besar dari satu [7]. Matrix kovarian dapat menentukan ciri sebuah pola dengan cara mengukur penyebaran pola dari kumpulan data. Untuk menghitung nilai kovarian, dapat digunakan persamaan sebagai berikut:

$$\text{Cov}(x, y) = \frac{\sum_{i=1}^n \Sigma(x_i - \bar{x})(y_i - \bar{y})}{(n - 1)} \quad (2)$$

Dimana: x_i = elemen ke-i dari matrix x
 y_i = elemen ke-i dari matrix y
 \bar{x} = rata rata dari matrix x
 \bar{y} = rata rata dari matrix y

3. Hitung vektor Eigen dan nilai Eigen yang sesuai

Vektor eigen atau vektor karakteristik dari transformasi linier, adalah vektor bukan nol yang paling banyak berubah oleh faktor skalar ketika transformasi linier tersebut diterapkan padanya. Nilai eigen yang sesuai adalah faktor yang digunakan untuk menskalakan vektor eigen. Secara umum, vektor eigen dari matriks A adalah vektor yang berlaku sebagai berikut:

$$A\vec{v} = \lambda\vec{v} \quad (3)$$

$$A\vec{v} - \lambda\vec{v} = 0 \quad (4)$$

$$\approx \vec{v}(A - \lambda I) = 0 \quad (5)$$

Dimana: A = matrix A
 \vec{v} = vektor eigen
 λ = nilai eigen
 I = matrix identitas

4. Pilih vektor eigen dengan nilai eigen terbesar

Untuk mendapatkan komponen utama dalam urutan signifikansinya, kita perlu memberi peringkat dalam urutan nilai eigennya. Urutkan vektor eigen sesuai dengan urutan penurunan nilai eigennya. Pilih k, dimana k adalah jumlah dimensi yang diinginkan pada dataset baru..

Setelah memilih beberapa komponen utama, matriks vektor baru dibuat dan disebut *feature vector*.

5. Menyusun kembali data di sepanjang sumbu komponen utama

Pada langkah terakhir, kita perlu mengubah sampel kita ke subruang baru dengan mengarahkan kembali data dari sumbu asli ke sumbu yang sekarang diwakili oleh komponen utama.

$$\text{Data Akhir} = \text{Feature Vector} * \text{Transpose (Scaled (Data))} \quad (6)$$

2.6. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) merupakan algoritma *supervised machine learning*, yang digunakan untuk melakukan klasifikasi data. Klasifikasi dilakukan terhadap data latih yang akan digunakan untuk menentukan data uji berdasarkan jarak terdekat terhadap data latih tersebut. Pada *K-Nearest Neighbor* terdapat beberapa cara untuk menghitung jarak tersebut. Berikut adalah cara yang digunakan dalam penelitian kali ini.[10]

$$\text{Cityblock} = \sum_{r=1}^N |x_{ir} - x_{jr}| \quad (7)$$

$$\text{Cosine} = \frac{\sum_{r=1}^N x_{ir} \times x_{jr}}{\sqrt{\sum_{r=1}^n (x_{ir})^2} \times \sqrt{\sum_{r=1}^n (x_{jr})^2}} \quad (8)$$

$$\text{Euclidean} = \sqrt{\sum_{r=1}^N (x_{ir} - x_{jr})^2} \quad (9)$$

Dimana:

- x_{ir} = nilai titik i dengan atribut ke-r
- x_{jr} = nilai titik j dengan atribut ke-r
- r = atribut pertama
- N = jumlah atribut

Setiap algoritma pasti memiliki kelebihan dan kekurangan masing – masing. Algoritma *K-Nearest Neighbor* memiliki kelebihan yaitu sangat efektif digunakan pada data yang memiliki ukuran data latih yang besar. Adapun kelemahan Algoritma *K-Nearest Neighbor* adalah perlunya menentukan berbagai jenis parameter. Yang akan membingungkan untuk mencari parameter apa saja yang baik untuk sistem tersebut

3. HASIL DAN PEMBAHASAN

Bagian ini berisi hasil pengujian sistem terhadap beberapa skenario untuk mendapatkan akurasi terbaik. Adapun parameter yang ditinjau adalah ukuran citra, persentase PCA, jenis KNN, dan nilai k.

3.1. Skenario Pengujian

Proses pengujian dilakukan terhadap 1464 citra yang terbagi menjadi citra *cloudy*, citra *foggy*, citra *rainy*, citra *shine*, dan citra *sunrise*. Dari 1464 citra tersebut, akan dibagi menjadi 70% citra latih dan 30% citra uji. Setelah itu dilakukan pencarian parameter – parameter dengan akurasi terbaik untuk diterapkan kedalam sistem.

1. Skenario 1: Pengujian ukuran citra

Skenario pertama ini akan dilakukan pengujian terhadap ukuran citra dengan cara merubah ukuran citra menjadi 64 x 64, 128 x 128, dan 256 x 256. Parameter lain yang digunakan adalah jenis KNN = *Cosine*, nilai KNN di $k = 3$, dan Persentase PCA 10%.

Tabel 1. Hasil Pengujian Pertama

Ukuran Citra	Akurasi
64x64	75,45%
128x128	78,18%
256x256	81,36%

Dari hasil pengujian tersebut dapat diketahui bahwa ukuran citra yang paling baik digunakan adalah 256x256 dengan akurasi 81,36% ini disebabkan karena semakin besar sebuah citra maka jumlah informasi yang kita dapatkan akan semakin banyak.

2. Skenario 2: Pengujian persentase PCA

Pengujian kedua akan mencari persentase PCA terbaik antara 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%. Parameter lain yang digunakan adalah jenis KNN = *Cosine*, ukuran citra = 256x256, dan nilai KNN di $k = 3$.

Tabel 2. Hasil Pengujian Kedua

Persentase PCA	Akurasi
10%	81,36%
20%	84,09%
30%	85,00%
40%	82,04%
50%	80,22%
60%	74,77%
70%	68,40%
80%	63,18%
90%	58,86%
100%	19,09%

Dari hasil pengujian tersebut dapat diketahui bahwa persentase PCA yang paling baik digunakan adalah 30% dengan akurasi sebesar 85,00%. Hal ini menandakan bahwa ada ciri yang tidak tepat pada saat proses ekstrasi ciri. Ciri ini lah yang membuat persentase menurun ketika dipakai. Sehingga ciri yang optimal dipakai hanya 30% dari keseluruhan ciri.

3. Skenario 3: Pengujian jenis KNN

Skenario ketiga akan dilakukan pengujian terhadap jenis KNN yaitu *Cityblok*, *Cosine* dan *Euclidean*. Parameter lain yang digunakan adalah ukuran citra = 256 x 256, nilai KNN di $k = 3$, dan Persentase PCA 30%.

Tabel 3. Hasil Pengujian Ketiga

Jenis KNN	Akurasi
<i>Cityblok</i>	79,77%
<i>Cosine</i>	85,00%
<i>Euclidean</i>	80,45%

Dari hasil pengujian tersebut dapat diketahui bahwa jenis KNN yang paling baik digunakan adalah *Cosine* dengan akurasi sebesar 85,00%.

4. Skenario 4: Pengujian nilai k

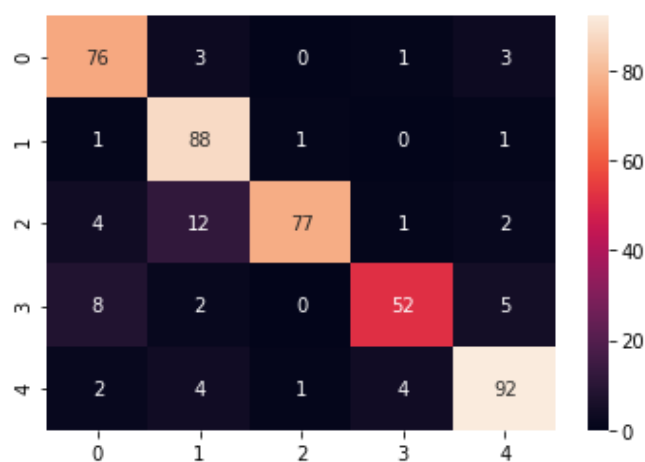
Skenario keempat ini, dilakukan pengujian terhadap nilai KNN 1, 3, 5, 7, dan 9. Parameter lain yang digunakan adalah jenis KNN = *Cosine*, ukuran citra = 256x256, dan Persentase PCA 30%.

Tabel 4. Hasil Pengujian Keempat

Nilai K	Akurasi
1	82,04%
3	85,00%
5	85,00%
7	85,90%
9	87,50%

Dari hasil pengujian tersebut dapat diketahui bahwa nilai KNN yang paling baik digunakan pada k = 9 dengan akurasi sebesar 87,50%. Hal ini menunjukkan bahwa semakin banyak ketetapan maka semakin banyak perbandingan dalam menentukan klasifikasi.

3.2. Data Hasil Pengujian Sistem



Gambar 2. Confusion Matrix

Berdasarkan hasil pengujian didapatkan parameter terbaik dengan menggunakan ukuran citra 256 x 256, persentase PCA 30%, jenis KNN menggunakan *cosine*, nilai k = 9. Melalui parameter-parameter tersebut didapatkan hasil akurasi 87,50%. Akurasi tersebut didapatkan dengan melakukan pengujian terhadap 440 data uji. Gambar 2 menunjukkan *confusion matrix* dengan menggunakan parameter terbaik. Angka 0 menunjukkan citra *cloudy*, angka 1 menunjukkan citra *foggy*, angka 2 menunjukkan citra *rainy*, angka 3 menunjukkan citra *shine*, angka 4 menunjukkan citra *sunrise*. Dari *confusion matrix* pada gambar 5 dapat dikatakan bahwa:

- Sistem mendeteksi 76 citra *cloudy* secara benar sedangkan 3 citra diprediksi sebagai *foggy*, 1 citra diprediksi sebagai *shine*, 3 citra diprediksi sebagai *sunrise*.
- Sistem mendeteksi 88 citra *foggy* secara benar sedangkan 1 citra diprediksi sebagai *cloudy*, 1 citra diprediksi sebagai *rainy*, 1 citra diprediksi sebagai *sunrise*.
- Sistem mendeteksi 77 citra *rainy* secara benar sedangkan 4 citra diprediksi sebagai *cloudy*, 12 citra diprediksi sebagai *foggy*, 1 citra diprediksi sebagai *shine*, 2 citra diprediksi sebagai *sunrise*.
- Sistem mendeteksi 52 citra *shine* secara benar sedangkan 8 citra diprediksi sebagai *cloudy*, 2 citra diprediksi sebagai *foggy*, 5 citra diprediksi sebagai *sunrise*.
- Sistem mendeteksi 92 citra *sunrise* secara benar sedangkan 2 citra diprediksi sebagai *cloudy*, 4 citra diprediksi sebagai *foggy*, 1 citra diprediksi sebagai *rainy*, 4 citra diprediksi sebagai *shine*.

4. KESIMPULAN

Penelitian kali ini merancang sistem klasifikasi cuaca dengan metode *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (KNN). Setelah penelitian dilakukan, didapatkan sistem klasifikasi cuaca dengan akurasi sebesar 87,50%. Adapun parameter – parameter terbaik dari sistem

tersebut adalah ukuran citra 256 x 256, hal ini dikarenakan semakin besar ukuran citra maka semakin banyak informasi yang terdapat dalam citra tersebut, jenis KNN adalah *Cosine*, nilai jumlah ketetangaan di $k = 9$, persentase PCA di 30%. Hal ini dikarenakan tidak semua ciri merupakan ciri yang berpengaruh terhadap performansi sistem.

DAFTAR PUSTAKA

- [1] N. Iriadi, "Penerapan Algoritma Klasifikasi Data Mining Dalam," *KNiST*, vol. XIV, no. 2, pp. 120–129, 2012.
- [2] A. Perbandingan, A. Svm, and D. A. N. C. N. N. Untuk, "Comparative Analysis of Image Classification Algorithm for," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, pp. 311–318, 2021, doi: 10.25126/jtiik.202184553.
- [3] A. A. Nurcahyani and R. Saptono, "Identifikasi Kualitas Beras dengan Citra Digital," *Sci. J. Informatics*, vol. 2, no. 1, pp. 63–72, 2016, doi: 10.15294/sji.v2i1.4530.
- [4] M. O. Arowolo, M. Adebisi, A. Adebisi, and O. Okesola, "PCA Model for RNA-Seq Malaria Vector Data Classification Using KNN and Decision Tree Algorithm," *2020 Int. Conf. Math. Comput. Eng. Comput. Sci. ICMCECS 2020*, 2020, doi: 10.1109/ICMCECS47690.2020.240881.
- [5] T. ArchanaH. and D. Sachin, "Dimensionality Reduction and Classification through PCA and LDA," *Int. J. Comput. Appl.*, vol. 122, no. 17, pp. 4–8, 2015, doi: 10.5120/21790-5104.
- [6] R. Jayanti, B. Hidayat, S. S.-P. SENIATI, and undefined 2017, "Identifikasi Pola Rugae Palatina Menggunakan Metode Watershed Dan Knn," *Ejournal.Itn.Ac.Id*, pp. 1–6, 2017, [Online]. Available: <http://ejournal.itn.ac.id/index.php/seniati/article/view/899>.
- [7] F. Muwardi *et al.*, "Pengolahan Citra Dan Pengklasifikasi Jarak," *J. Ilmu Tek. Elektro Komput. dan Inform.*, vol. 3, no. 2, pp. 124–131, 2017.
- [8] D. M. Wonohadidjojo, "Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih," *Ultim. J. Tek. Inform.*, vol. 13, no. 1, pp. 51–57, 2021, doi: 10.31937/ti.v13i1.2040.
- [9] G. Rahayu and Mustakim, "Principal Component Analysis Untuk Dimensi Reduksi Data Clustering Sebagai Pemetaan Persentase Sertifikasi Guru Di Indonesia," *Semin. Nas. Teknol. Inf. Komun. dan Ind.*, vol. 0, no. 0, pp. 201–208, 2017, [Online]. Available: <http://ejournal.uin-suska.ac.id/index.php/SNTIKI/article/view/3265>.
- [10] D. M. M. M. Reza Noviansyah, Tedy Rismawan, "Penerapan Data Mining Menggunakan Metode K-Nearest Neighbor Untuk Klasifikasi Indeks Cuaca Kebakaran Berdasarkan Data Aws (Automatic Weather Station) (Studi Kasus: Kabupaten Kubu Raya)," *J. Coding, Sist. Komput. Untan*, vol. 06, no. 2, pp. 48–56, 2018.