

# Perbandingan Performance Algoritma KNN dan Liner Regresi Dalam Percepatan Masa Tanam Sawi Berdasarkan Suhu air dan Nutrisi

M. Zaky Fanany\*<sup>1</sup>, Handoyo Widi Nugroho <sup>2</sup>

<sup>1,23</sup>Jurusan Teknik Komputer, IIB Darmajaya

Jl. ZA. Pagar Alam No.93, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung,  
Lampung 35141, telp/ (0721) 787214

e-mail: \*<sup>1</sup>[mzakyfanany@radenintan.ac.id](mailto:mzakyfanany@radenintan.ac.id), <sup>2</sup>[handoyo.w.d@darmajaya.ac.id](mailto:handoyo.w.d@darmajaya.ac.id)

## Abstrak

*Hidroponik merupakan kemajuan teknologi dibidang pertanian yang memungkinkan penanaman tanpa menggunakan tanah dan mengandalkan air serta nutrisi untuk pertumbuhan tanaman. Metode ini efektif dalam mengatasi masalah dibidang pertanian seperti kekeringan dan hama yang dapat menghambat pertumbuhan tanaman sayur. Studi ini menginvestigasi penggunaan algoritma K-Nearest Neighbors (KNN) dan Linear Regression untuk memprediksi waktu tumbuh tanaman sawi berdasarkan suhu air dan kadar nutrisi. Dataset yang digunakan mencakup variabel-variabel ini yang diukur pada periode pertumbuhan. Langkah-langkah eksperimental termasuk membagi data menjadi set pelatihan dan pengujian, standarisasi fitur, pelatihan model, dan evaluasi menggunakan metrik seperti Mean Mean Squared Error (MSE): Model KNN (3.69625) memiliki MSE lebih rendah dibandingkan Linear Regression (4.33562), Root Mean Squared Error (RMSE): KNN (1.92156) memiliki RMSE lebih rendah dibandingkan Linear Regression (2.08121) R<sup>2</sup> Score: KNN (0.96567) memiliki R<sup>2</sup> Score sedikit lebih tinggi dibandingkan Linear Regression (0.95932),*

*Kata Kunci: K-Nearest Neighbors, KNN, Linear Regression, waktu tumbuh, tanaman sawi, prediksi, evaluasi model*

## Abstract

*Hydroponics is a technological advancement in agriculture that allows planting without using soil and relying on water and nutrients for plant growth. This method is effective in overcoming problems in the agricultural sector such as drought and pests which can inhibit the growth of vegetable plants. This study investigates the use of K-Nearest Neighbors (KNN) and Linear Regression algorithms to predict the growth time of mustard greens based on water temperature and nutrient levels. The dataset used includes these variables measured during the growth period. Experimental steps included dividing the data into training and testing sets, feature standardization, model training, and evaluation using metrics such as Mean Squared Error (MSE): KNN model (3.69625) had lower MSE compared to Linear Regression (4.33562), Root Mean Squared Error (RMSE): KNN (1.92156) had lower RMSE compared to Linear Regression (2.08121), R<sup>2</sup> Score: KNN (0.96567) had a slightly higher R<sup>2</sup> Score compared to Linear Regression (0.95932).*

*Keywords: K-Nearest Neighbors, KNN, Linear Regression, growing time, mustard greens, prediction, model evaluation*

## 1. PENDAHULUAN

Perkembangan teknologi saat ini telah mencapai tingkat yang sangat pesat dan mencakup berbagai sektor, tidak hanya terbatas pada komunikasi. Beragam inovasi baru yang muncul telah mempengaruhi kehidupan masyarakat, membuktikan bahwa teknologi berkembang dengan sangat cepat. Teknologi memiliki peran penting dalam kehidupan masyarakat karena memudahkan akses informasi tanpa batasan jarak, tempat, dan waktu, yang biasanya menjadi kendala. Perkembangan teknologi berdampak positif pada berbagai bidang seperti kesehatan, pendidikan, industri, pertanian, dan lain-lain[1][2][3].

Salah satu kemajuan teknologi dibidang pertanian adalah metode hidroponik. Hidroponik memungkinkan penanaman tanpa menggunakan tanah dan mengandalkan air serta nutrisi untuk pertumbuhan tanaman. Metode ini efektif dalam mengatasi masalah dibidang pertanian seperti kekeringan dan hama yang dapat menghambat pertumbuhan tanaman sayur [4][5]. Metode *K-Nearest Neighbor (KNN)* adalah salah satu teknik dalam machine learning.[6][7], KNN merupakan algoritma sederhana yang diperkenalkan oleh *Fix* dan *Hodges* dan digunakan untuk mengklasifikasi data. Dalam penelitian ini, *KNN* digunakan untuk menentukan tanaman mana yang lebih cepat siap panen. Algoritma ini akan dibandingkan dengan algoritma lain yang menunjukkan pertumbuhan tanaman melalui indikator seperti *pH* air dan nutrisi[8][9]. Regresi Linear adalah algoritma yang digunakan untuk memprediksi hubungan antara variabel output dan variabel input independen dengan belajar dari sejumlah sampel data[10][11].

Penelitian terdahulu yang pernah diangkat adalah Perbandingan Metode linier *Regression*, *Random Forest* dan *KNN* Untuk Prediksi Hasil Panen Padi di Provinsi Jawa Barat. Metode *Random Forest* memberikan akurasi prediksi tertinggi dibandingkan dengan *Linear Regression* dan *KNN*. Penerapannya menggunakan teknologi *machine learning* dengan dataset hasil panen padi. Perbandingan tiga algoritma *machine learning (Linear Regression, Random Forest, dan KNN)* dalam memprediksi hasil panen padi[8][9]. Berikutnya prediksi hasil panen padi tahun 2023 menggunakan metode *Regresi Linier* di Kabupaten Indramayu. Hasilnya berupa regresi linier memberikan prediksi yang cukup akurat untuk hasil panen padi tahun 2023, menggunakan teknologi statistik dan analisis data. Analisis regresi linier pada data historis hasil panen padi untuk memprediksi hasil panen di masa mendatang[10].

Pengukuran Panjang-Berat Ikan dan Sayuran secara Otomatis pada Budidaya dalam ember (Budi Daya Ikan dalam Ember) Menggunakan Visi Komputer dan *Regresi Linier* Pengukuran otomatis menggunakan visi komputer dan regresi linier menunjukkan akurasi tinggi dalam pengukuran panjang dan berat ikan serta sayuran. Visi komputer dan algoritma regresi linier. Metode menggunakan kamera dan *software* analisis gambar untuk mengukur panjang dan berat ikan serta sayuran secara otomatis [11], Perbandingan *Algoritma Cart dan KNN* untuk Prediksi Luas Lahan Panen Tanaman Padi di Kabupaten Indramayu. Algoritma *CART* memberikan hasil prediksi yang lebih akurat dibandingkan dengan *KNN*. Algoritma *machine learning CART* dan *KNN*. Perbandingan kinerja dua algoritma dalam memprediksi luas lahan panen padi[10], Akurasi Estimasi Kadar Sukrosa pada Penentuan Tingkat Kematangan Pepaya Menggunakan Nilai RGB Berbasis Aplikasi Mobile. Estimasi kadar sukrosa menggunakan nilai RGB dari aplikasi mobile menunjukkan hasil yang akurat. *Aplikasi mobile* dengan analisis nilai *RGB*. Menggunakan aplikasi mobile untuk menangkap gambar pepaya dan menganalisis kadar sukrosa berdasarkan nilai *RGB* [12].

Penelitian ini bertujuan untuk membandingkan dua metode, *KNN* dan *Regresi Linear*, untuk menentukan metode mana yang paling akurat dalam menentukan masa tanam. Dengan memanfaatkan teknologi yang berkembang pesat, penelitian ini akan mengkaji keefektifan kedua algoritma tersebut dalam memprediksi waktu panen tanaman melalui berbagai indikator pertumbuhan seperti *pH* air dan nutrisi. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam peningkatan efisiensi dan produktivitas pertanian, khususnya dalam metode penanaman hidroponik[13].

## 2. METODE PENELITIAN

Pada tahap-tahap penelitian ini dapat dilihat pada gambar di bawah, yang menampilkan langkah-langkah yang dilakukan dalam penelitian ini. Tahap pertama adalah tinjauan pustaka, diikuti oleh pengumpulan data, prediksi menggunakan *KNN*, prediksi menggunakan *Regresi Linear*, evaluasi hasil prediksi, dan diakhiri dengan publikasi.

### 2.1 Algoritma KNN

Berikut ini adalah gambar tahapan-tahapan dalam melakukan proses prediksi menggunakan KNN, yang terdiri dari input data, *KNN* data *formatting*, *training*, prediksi, dan pengujian[5][6][15].

1. Langkah 1: Input Data  
Mulai dengan dataset yang terdiri dari titik-titik data dengan fitur-fitur yang relevan dan label atau nilai target yang sesuai untuk setiap titik data.
2. Langkah 2: Pilih Nilai K  
Tentukan nilai K, yaitu jumlah tetangga terdekat yang akan digunakan untuk memprediksi label atau nilai target dari titik data baru.
3. Langkah 3: Hitung Jarak  
Hitung jarak antara titik data baru (yang ingin diprediksi) dengan setiap titik data dalam dataset menggunakan metrik jarak seperti Euclidean distance atau manhattan distance. Ini dapat dihitung sebagai:

$$dis(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2} \quad (1)$$

Di sini,  $x_{1i}$  dan  $x_{2i}$  adalah nilai fitur dari titik data yang sedang diprediksi dan titik data lain dalam dataset.

4. Langkah 4: Identifikasi Tetangga Terdekat  
Identifikasi K titik data dengan jarak terdekat ke titik data baru. Ini akan menjadi K tetangga terdekat berdasarkan nilai jarak yang telah dihitung.
5. Langkah 5: Prediksi  
Klasifikasi: Untuk masalah klasifikasi, tentukan label dari titik data baru dengan menggunakan majority voting dari K tetangga terdekat. Artinya, label yang paling sering muncul di antara K tetangga terdekat akan menjadi prediksi untuk titik data baru.  
Regresi: Untuk masalah regresi, prediksi nilai target untuk titik data baru dapat dihitung sebagai rata-rata (mean) atau median dari nilai target K tetangga terdekat.
6. Langkah 6: Output  
Outputkan label atau nilai prediksi untuk titik data baru berdasarkan proses yang telah dijelaskan.

### 2.2 Linear Regression

Berikut ini adalah gambar tahapan-tahapan dalam melakukan proses prediksi menggunakan *Linear Regression*, yang terdiri dari input data, *Linear Regression* data preparation, training, prediksi kemudian pengujian[16] [17]

1. Langkah 1: Input Data  
Mulai dengan dataset yang terdiri dari pasangan nilai fitur ( $x$ ) dan nilai target ( $y$ ). Setiap sampel dalam dataset memiliki satu atau lebih nilai fitur dan nilai target yang sesuai.
2. Langkah 2: Persiapkan Data

Pisahkan dataset menjadi data pelatihan (training data) dan data pengujian (test data) jika diperlukan untuk evaluasi model. Lakukan normalisasi atau standarisasi data fitur jika diperlukan untuk memperbaiki kinerja algoritma.

3. Langkah 3: Inisialisasi Model  
Inisialisasi model Linear Regression dengan menetapkan parameter awal seperti koefisien ( $\theta$ ) dan intersep ( $b$ ).
4. Langkah 4: Latih Model  
Latih model menggunakan data pelatihan. Proses ini melibatkan menyesuaikan parameter model (koefisien dan intersep) agar sesuai dengan data pelatihan. Ini dilakukan dengan meminimalkan fungsi biaya (misalnya, Mean Squared Error atau MSE) antara nilai prediksi model dan nilai target yang sebenarnya dalam data pelatihan.
5. Langkah 5: Prediksi  
Setelah model dilatih, gunakan model untuk membuat prediksi terhadap data baru atau data pengujian. Prediksi dilakukan dengan mengalikan nilai fitur dengan koefisien yang sesuai dan menambahkan intersep.
6. Langkah 6: Evaluasi Model  
Evaluasi kinerja model menggunakan metrik evaluasi seperti Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), dan koefisien determinasi (R-squared). Analisis residual untuk memverifikasi asumsi model dan mengevaluasi kesesuaian model dengan data.
7. Langkah 7: Output  
Outputkan hasil prediksi dan hasil evaluasi model seperti metrik evaluasi yang telah dihitung.

### 3. HASIL DAN PEMBAHASAN

Hasil pengolahan data menggunakan algoritma *K-Nearest Neighbors (KNN)* dan *Linear Regression* untuk memprediksi percepatan masa tanam sawi berdasarkan suhu air dan kadar nutrisi menunjukkan perbedaan yang signifikan dalam kinerja kedua model tersebut. Berikut adalah hasil dari pengolahan data.

ID	Suhu_Air (°C)	Nutrisi (ppm)	Waktu_Tumbuh (hari)
1	18	400	35
2	20	450	32
3	22	500	30
4	24	550	28
5	26	600	26
6	28	650	25
7	30	700	24
8	32	750	23
9	34	800	22
10	36	850	21
11	18	420	34
12	20	470	31
13	22	520	29
14	24	570	27
15	26	620	25
16	28	670	24
17	30	720	23
18	32	770	22
19	34	820	21
20	36	870	20
21	18	440	33
22	20	490	30
23	22	540	28

Gambar 1 Potongan Dataset

**Keterangan Dataset**

Deskripsi: Dataset ini berisi 100 sampel data mengenai waktu tumbuh sawi (dalam hari) berdasarkan variasi suhu air (dalam derajat Celsius) dan kadar nutrisi (dalam ppm). Dataset ini dapat digunakan untuk melakukan analisis atau melatih model *machine learning* dalam memprediksi waktu tumbuh sawi berdasarkan suhu air dan kadar nutrisi yang diberikan.

**Kolom:**

1. ID: Nomor identifikasi untuk setiap sampel data (1 hingga 100).
2. Suhu\_Air (°C): Suhu air yang digunakan untuk menanam sawi, diukur dalam derajat Celsius. Nilai bervariasi dari 18°C hingga 36°C.
3. Nutrisi (ppm): Kadar nutrisi yang diberikan dalam larutan penanaman, diukur dalam ppm (parts per million). Nilai bervariasi dari 400 ppm hingga 1030 ppm.
4. Waktu Tumbuh (hari): Waktu yang dibutuhkan untuk menumbuhkan sawi hingga siap panen, diukur dalam hari. Nilai bervariasi dari 12 hari hingga 35 hari

Untuk mengolah dataset menggunakan algoritma *K-Nearest Neighbors (KNN)* di *Google Colab*, kita dapat mengikuti langkah-langkah berikut. Di bawah ini saya berikan kode yang dapat Anda salin dan tempel ke dalam *Google Colab* untuk menjalankan analisis menggunakan *KNN*.

### 3.1. Langkah-langkah Pengolahan Data

1. Unggah dataset ke *Google Colab*
2. Impor pustaka yang diperlukan
3. Memuat dataset
4. Mempersiapkan data untuk pelatihan dan pengujian
5. Melatih model *KNN*
6. Mengevaluasi model

```

# Memuat dataset
file_path = '/content/dataset_percepatan_masa_tanam_sawi.xlsx'
df = pd.read_excel(file_path)

# Mempersiapkan data untuk pelatihan dan pengujian
X = df[['Suhu_Air (°C)', 'Nutrisi (ppm)']]
y = df['Waktu_Tumbuh (hari)']

# Membagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menstandarkan fitur
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Melatih model KNN
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(X_train, y_train)

# Mengevaluasi model
y_pred_knn = knn.predict(X_test)
mse_knn = mean_squared_error(y_test, y_pred_knn)
rmse_knn = mse_knn ** 0.5
mae_knn = mean_absolute_error(y_test, y_pred_knn)
r2_knn = r2_score(y_test, y_pred_knn)

print(f"KNN Mean Squared Error: {mse_knn}")
print(f"KNN Root Mean Squared Error: {rmse_knn}")
print(f"KNN Mean Absolute Error: {mae_knn}")
print(f"KNN R^2 Score: {r2_knn}")

# Menampilkan beberapa prediksi vs nilai sebenarnya
comparison_knn = pd.DataFrame({'Actual': y_test, 'Predicted_KNN': y_pred_knn})
print(comparison_knn.head())

```

Gambar 2 Pengolahan data Algoritma KNN

#### Penjelasan Kode:

- a. Langkah 1: Mengunggah file Excel ke *Google Colab*.
- b. Langkah 2: Mengimpor pustaka yang diperlukan seperti *pandas*, *scikit-learn*, dll.
- c. Langkah 3: Memuat dataset dari file Excel ke dalam *DataFrame*.
- d. Langkah 4: Memisahkan dataset menjadi fitur (*X*) dan target (*y*), kemudian membagi dataset menjadi set pelatihan dan pengujian.
- e. Langkah 5: Melakukan standarisasi fitur dan melatih model *KNN*.
- f. Langkah 6: Mengevaluasi model dengan menghitung *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)*, serta menampilkan beberapa prediksi dibandingkan dengan nilai sebenarnya.

### 3.2. Ringkasan Hasil Pengolahan Data KNN

1. *Mean Squared Error (MSE)*: 3.69625
2. *Root Mean Squared Error (RMSE)*: 1.92156
3. *Mean Absolute Error (MAE)*: 1.49625
4. *R<sup>2</sup> Score*: 0.9656

Hasil Pengolahan Data dengan *Linear Regression*

```

# Import pustaka yang diperlukan
from sklearn.linear_model import LinearRegression

# Melatih model Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)

# Mengevaluasi model
y_pred_lr = lr.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = mse_lr ** 0.5
mae_lr = mean_absolute_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print(f"Linear Regression Mean Squared Error: {mse_lr}")
print(f"Linear Regression Root Mean Squared Error: {rmse_lr}")
print(f"Linear Regression Mean Absolute Error: {mae_lr}")
print(f"Linear Regression R^2 Score: {r2_lr}")

# Menampilkan beberapa prediksi vs nilai sebenarnya
comparison_lr = pd.DataFrame({'Actual': y_test, 'Predicted_LR': y_pred_lr})
print(comparison_lr.head())

```

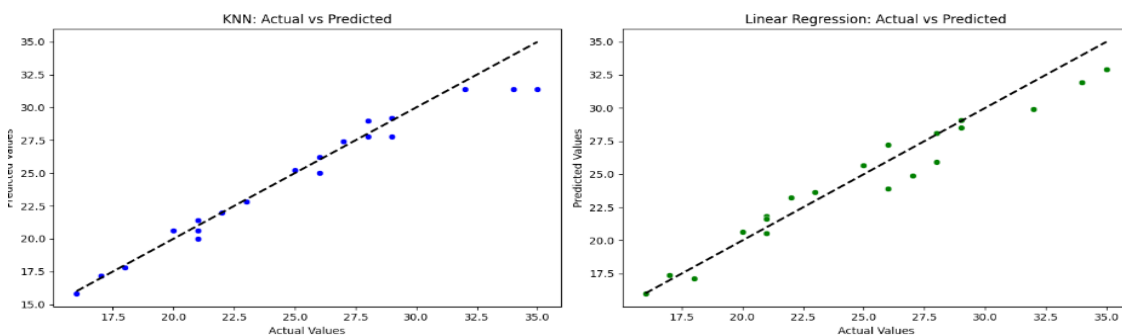
Gambar 3 Pengolahan data Algoritma Linier Regresi

## Penjelasan Kode:

- Langkah 1: Mengunggah file Excel ke Google Colab.
- Langkah 2: Mengimpor pustaka yang diperlukan seperti pandas, scikit-learn, dll.
- Langkah 3: Memuat dataset dari file Excel ke dalam DataFrame.
- Langkah 4: Memisahkan dataset menjadi fitur (X) dan target (y), kemudian membagi dataset menjadi set pelatihan dan pengujian.
- Langkah 5: Melakukan standarisasi fitur dan melatih model Linear Regression.
- Langkah 6: Mengevaluasi model dengan menghitung Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), dan R<sup>2</sup> Score, serta menampilkan beberapa prediksi dibandingkan dengan nilai sebenarnya.

## Linear Regression

- Mean Squared Error (MSE): 4.33562
- Root Mean Squared Error (RMSE): 2.08121
- Mean Absolute Error (MAE): 1.61738
- R<sup>2</sup> Score: 0.95932



Gambar 4 Kinerja dan KNN dan LR

- *KNN Scatter Plot*: Titik-titik dari model KNN lebih dekat ke garis ideal ( $Actual = Predicted$ ) dibandingkan dengan model Linear Regression, menunjukkan bahwa KNN memiliki prediksi yang lebih akurat.
- *Linear Regression Scatter Plot*: Meskipun prediksi *Linear Regression* juga cukup akurat, titik-titiknya cenderung lebih tersebar dibandingkan dengan *KNN*.

Dari hasil *scatter plot* dan metrik evaluasi, kita dapat menyimpulkan bahwa model KNN lebih baik dalam memprediksi waktu tumbuh sawi berdasarkan suhu air dan kadar nutrisi dibandingkan dengan model *Linear Regression*.

1. *Mean Squared Error (MSE)*: Model KNN (3.69625) memiliki *MSE* lebih rendah dibandingkan *Linear Regression* (4.33562), menunjukkan KNN lebih baik dalam meminimalkan kesalahan kuadrat rata-rata.
2. *Root Mean Squared Error (RMSE)*: KNN (1.92156) memiliki *RMSE* lebih rendah dibandingkan *Linear Regression* (2.08121), menandakan KNN memiliki kesalahan prediksi yang lebih kecil. *Mean Absolute Error (MAE)*: KNN (1.49625) memiliki *MAE* lebih rendah dibandingkan *Linear Regression* (1.61738), menunjukkan rata-rata kesalahan absolut dari prediksi KNN lebih kecil.
3. *R<sup>2</sup> Score*: KNN (0.96567) memiliki *R<sup>2</sup> Score* sedikit lebih tinggi dibandingkan *Linear Regression* (0.95932), menandakan bahwa model KNN dapat menjelaskan lebih banyak variasi dalam data.

Secara keseluruhan, KNN menunjukkan kinerja yang lebih baik dalam semua metrik evaluasi yang disajikan.

#### 4. KESIMPULAN

Secara keseluruhan, dari hasil perbandingan metrik evaluasi antara model KNN dan *Linear Regression*:

1. *KNN* menunjukkan kinerja yang lebih baik dalam meminimalkan kesalahan prediksi, dengan *MSE*, *RMSE*, dan *MAE* yang lebih rendah dibandingkan *Linear Regression*. Hal ini menunjukkan bahwa KNN memberikan prediksi yang lebih akurat dan mendekati nilai sebenarnya.
2. *R<sup>2</sup> Score* dari model *KNN* juga sedikit lebih tinggi, menandakan bahwa model KNN dapat menjelaskan variasi dalam data dengan lebih baik daripada *Linear Regression*. Ini mengindikasikan bahwa *KNN* lebih cocok untuk menjelaskan hubungan antara variabel input dan output dalam dataset tersebut.

Dengan demikian, berdasarkan semua metrik evaluasi yang diberikan, *KNN* secara konsisten menunjukkan performa yang lebih baik dibandingkan *Linear Regression* dalam konteks dataset yang dievaluasi.

#### 5. SARAN

Penelitian selanjutnya disarankan untuk memperluas dataset dengan lebih banyak variabel lingkungan seperti *pH* air, kelembaban udara, dan intensitas cahaya, yang mungkin juga mempengaruhi pertumbuhan sawi. Selain itu, penggunaan algoritma *machine learning* yang lebih canggih seperti *Random Forest* atau *Gradient Boosting* dapat dieksplorasi untuk melihat apakah dapat meningkatkan akurasi prediksi.



#### DAFTAR PUSTAKA

- [1] A. A. Ilham, N. Harun, and Arwansyah, "Simulasi Penerapan Teknologi Data Mining Untuk Menghasilkan Model Pola Tanam Berkelanjutan," *Pros. Konf. Nas. Ilmu Komput. 2013*, pp. 124–132, 2013, [Online]. Available: <http://repository.unhas.ac.id/handle/123456789/9384>
- [2] T. Wahyudi *et al.*, "Penerapan Data Mining Pada Transaksi Penjualan Pakaian Dengan Menggunakan Algoritma Apriori," *Jupiter*, vol. 14, no. 2, pp. 473–482, 2022.
- [3] A. Ardiansyah, J. Triloka, and Indera, "Evaluasi Kinerja Model YOLOv8 dalam Deteksi Kesegaran Buah," *JUPITER J. Penelit. Ilmu Dan Teknol. Komput.*, pp. 357–368, 2024.
- [4] E. F. Yogatama and L. Rifani, "Aplikasi Data Mining Untuk Klasifikasi Kesuburan Tanaman Selada Hidroponik Dengan Metode K Nearest Network," *J. Ilmu Komput. dan Bisnis*, vol. 13, no. 2, pp. 33–43, 2022, doi: 10.47927/jikb.v13i2.308.
- [5] S. C. Wohon, D. Hatidja, and N. Nainggolan, "Penentuan Model Regresi Terbaik Dengan Menggunakan Metode Stepwise ( Studi Kasus : Impor Beras Di Sulawesi Utara ) Determining the Best Regression Model Using Stepwise Method ( Case Study : Rice Imports in North Sulawesi )," *J. Ilm. Sains*, vol. 17, no. 2, p. 81, 2017.
- [6] M. F. Aziz, S. Defiyanti, and B. N. Sari, "Perbandingan Algoritma Cart Dan K-Nearest Neighbor Untuk Prediksi Luas Lahan Panen Tanaman Padi Di Kabupaten Karawang," *J. TAM (Technology Accept. Model.*, vol. 9, no. 2, pp. 74–78, 2018.
- [7] A. P. Wibowo, "Komparasi Algoritma Klasifikasi untuk Penentuan Jenis Spesies Tanaman Hutan," *J. Cakrawala Inf.*, vol. 1, no. 1, pp. 12–19, 2021, doi: 10.54066/jci.v1i1.209.
- [8] H. Panen, P. Di, and P. Jawa, "PERBANDINGAN METODE LINEAR REGRESSION , RANDOM FOREST & K-NEAREST NEIGHBOR UNTUK PREDIKSI PRODUKSI," vol. 8, no. 3, pp. 3895–3900, 2024.
- [9] R. Andrianto and F. Irawan, "Implementasi Metode Regresi Linear Berganda Pada Sistem Prediksi Jumlah Tonase Kelapa Sawit di PT . Paluta Inti Sawit," *J. Pendidik. Tambusai*, vol. 7, no. 1, pp. 2926–2934, 2023.
- [10] A. Bahtiar, "Jurnal Informatika Terpadu PREDIKSI HASIL PANEN PADI TAHUN 2023 MENGGUNAKAN METODE REGRESI LINIER DI KABUPATEN INDRAMAYU," *J. Inform. Terpadu*, vol. 9, no. 1, pp. 18–23, 2023, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JIT>
- [11] H. Fitriyah, "Pengukuran Panjang-Berat Ikan dan Sayuran pada Budikdamber (Budi Daya Ikan dalam Ember) Menggunakan Visi Komputer dan Regresi Linier," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 8–14, 2020, doi: 10.47970/siskom-kb.v4i1.166.
- [12] M. L. Firdhaus, F. Romadlon, and F. M. Wibowo, "Akurasi Estimasi Brix pada Tingkat Kematangan Pepaya Menggunakan Nilai RGB Berbasis Aplikasi Mobile," *Ind. J. Teknol. dan Manaj. Agroindustri*, vol. 8, no. 2, pp. 79–86, 2019, doi: 10.21776/ub.industria.2019.008.02.1.
- [13] A. Bravo, T. Tursina, and H. Sastypratiwi, "Penerapan Metode Naive Bayes Untuk Penentuan Bibit Kelapa Sawit Berdasarkan Kondisi Daerah Tanam dan Perawatan Tanaman," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 1, p. 101, 2023, doi: 10.26418/justin.v11i1.52277.
- [14] U. Hasdiana, "Title," *Anal. Biochem.*, vol. 11, no. 1, pp. 1–5, 2018, [Online]. Available: <http://link.springer.com/10.1007/978-3-319-59379-1%0Ahttp://dx.doi.org/10.1016/B978-0-12-420070-8.00002-7%0Ahttp://dx.doi.org/10.1016/j.ab.2015.03.024%0Ahttps://doi.org/10.1080/07352689.2018.1441103%0Ahttp://www.chile.bmw-motorrad.cl/sync/showroom/lam/es/>
- [15] M. S. E. ,M. K. M. Y. S. K. ,M. K. Kandi Sri, "Penerapan Data Mining untuk Meprediksi Hasil Panen Pertanian Tanaman Padi Daerah Serdang Bedagai Menggunakan Metode

C4.5 Pada Dinas Tanaman Pangan Dan Holtikultura Sumut,” *J. Sains Manaj. Inform. dan Komput.*, vol. 1, no. 1, pp. 01–07, 2020.