

PERBANDINGAN KLASIFIKASI KACANG KERING DENGAN ALGORITMA KNN DAN SVM LINEAR

Alfian Indrajaya

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail:

ABSTRAK

Kacang merupakan sumber pangan yang sering digunakan sebagai bahan makanan, dengan adanya kemajuan teknologi yang pesat kacang-kacangan mendapatkan jenis jenis baru. Untuk mengklasifikasi jenis jenis kacang semakin susah. Namun untuk mengklasifikasi jenis kacang kering dapat dipermudah dengan menggunakan algoritma Machine learning seperti Support Vector Machine dan K-Nearest Neighbor. Tujuan dari penelitian ini adalah untuk mengetahui dan membedakan jenis jenis kacang dan menampilkan hasil akurasi, precision, recall, f1-score yang terbaik dari algoritma Support Vector Machine dan K-Nearest Neighbor. Dataset ini bersumber dari website Kaggle yang terdapat 16 fitur dan 1 kelas untuk klasifikasi. Hasil pengujian yang telah dilakukan mendapatkan hasil bahwa dari setiap pembagian data latih dan data uji algoritma K-Nearest Neighbor mendapatkan hasil yang lebih tinggi dibandingkan Support Vector Machine. Sehingga pada penelitian ini K-Nearest Neighbor menghasilkan nilai klasifikasi yang lebih baik.

Kata kunci : 3-5 Klasifikasi; K-Nearest Neighbor; Support Vector Machine; Polinomial

ABSTRACT

Advancements in technology have led to the creation of new types of beans, making their classification more difficult. However, the classification of dry beans can be simplified using machine learning algorithms such as Support Vector Machine and K-Nearest Neighbor. The objective of this study was to identify and differentiate types of beans and determine the algorithms that yield the best accuracy, precision, recall, and f1-score. The dataset used for this research was obtained from the Kaggle website and consists of 16 features and 1 class for classification. The findings reveal that the K-Nearest Neighbor algorithm outperformed the Support Vector Machine algorithm in terms of classification accuracy, regardless of the division between training and test data. Thus, K-Nearest Neighbor provides better results for the classification of beans in this study.

Key words: Classification; K-Nearest Neighbour; Support Vector Machine; Polinomial

1. PENDAHULUAN

1.1 Latar Belakang Permasalahan

Pada zaman sekarang teknologi sudah sangat maju, dengan adanya kemajuan pada teknologi manusia dapat menciptakan tumbuhan baru dengan menggunakan rekayasa genetik. Rekayasa genetik yang dilakukan dapat menciptakan jenis tanaman yang baru, seperti jenis-jenis kacang yang ada. Kacang merupakan tumbuhan yang bermula dari biji-bijian. Kacang mengandung serat yang besar, kalori yang rendah sehingga tidak mengganggu kesehatan saat mengonsumsi kacang-kacangan. Jenis kacang dapat dibedakan dengan cara melihat ukuran, panjang serta warna dari kacang tersebut [1][2]. Terdapat senyawa yang ada pada kacang-kacangan yaitu senyawa fenolik, senyawa ini berperan untuk membantu metabolisme pada tubuh manusia [3]. Selain itu jenis kacang-kacangan yang terdapat pada dunia banyak salah satunya adalah kacang kering.

Kacang kering merupakan Tumbuhan yang memiliki kandungan nutrisi yang sangat banyak seperti serat, dan vitamin yang tinggi, [4][5]. Kacang kering juga kerap sekali digunakan untuk metode diet rata-rata orang untuk memenuhi asupan kalori yang tidak berlebihan [6] selain itu dengan banyak mengkonsumsi kacang kering dapat mencegah terserangnya penyakit hati, mengatur gula dalam tubuh dan menjaga kesehatan tubuh.[7][8] Namun kualitas pada Kacang kering juga harus dilihat dikarenakan banyak sekali kacang kering yang terkena jamur jamur yang dapat menimbulkan penyakit.[9]

1.2 Tujuan

Penelitian ini bertujuan untuk mengklasifikasi jenis kacang kering yang ada dengan menggunakan metode *K-Nearest Neighbor* dan *Support vector machine* dengan *kernel Polynomial*.

1.3 Manfaat

Dengan adanya klasifikasi kacang kering dapat memberikan manfaat kepada pihak rumah makan dan ibu rumah tangga karena dapat terbantu dengan membedakan jenis kacang kering yang cocok untuk diolah pada masakan.

1.4 Metode yang digunakan

Metode algoritma yang digunakan untuk mengklasifikasi kacang kering adalah *K-Nearest neighbor* dan *Support Vector Machine*

2. METODE PENELITIAN

2.1 Dataset

Data yang digunakan dalam klasifikasi kacang kering terdapat pada situs web kaggle yang dapat diakses melalui <https://www.kaggle.com/datasets/nimapourmoradi/dry-bean-dataset-classification>. Dataset ini berupa file csv yang terdiri dari 17 variabel , 16 diantaranya merupakan data numerik dan yang 1 berupa data kelas yang masing masing memiliki 13611 data.

2.2 Pra-pemrosesan data

Setelah terkumpul datanya, data akan diperiksa apakah ada missing value pada dataset tersebut, jika tidak ada missing value maka selanjutnya memilih attribut yang akan digunakan untuk diuji. Setelah selesai mengisi missing value dan memilih attribut yang akan digunakan selanjutnya adalah memecah jumlah data uji dan data latih dengan menggunakan *train_test_split* library dari Scikit-learn pada python. Tujuan dari pemecahan jumlah data uji dan data latih adalah untuk mendapatkan akurasi , dan tingkat performa dari model yang diteliti.

2.3 K-Nearest Neighbor

K-Nearest Neighbor atau yang disingkat KNN merupakan algoritma yang mempunyai kinerja tanpa parameter tetapi didasarkan pada nilai tetangga k yang paling dekat. [10][11] K-Nearest Neighbor disebut jengan Lazy Learning dimana metode K-Nearest Neighbor hanya menghitung dari jarak antara data uji dan data latih [12] Rumus yang diggunakan pada K-Nearest Neighbor juga dapat dihitung dengan rumus Euclidian distance pada persamaan (1)

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1)$$

Keterangan :

d = jarak antar data

xk = data latih

yk = data uji

n = jumlah fitur yang ada pada data

selain itu K-Nearest Neighbor juga mempunyai kelebihan dan kekurangan. Kelebihannya adalah metode K-Nearest Neighbor memiliki Tingkat akurasi yang tinggi terhadap data ekstrim [13]

2.4 Support Vector Machine

Support Vector Machine merupakan algoritma nonlinear yang tujuannya adalah menemukan nilai dari hyperplane (garis pembatas) dengan tujuan untuk memisahkan dua kelas data. Support Vector Machine memiliki kernel contohnya seperti kernel polinomial. Kernel ini mempunyai perhitungan pada persam [1]aan (2)[14][15]

$$K(x_i, x_j) = (x_i^T x_j + r)^d \quad (2)$$

Keterangan :

K = Kernel Polinomial

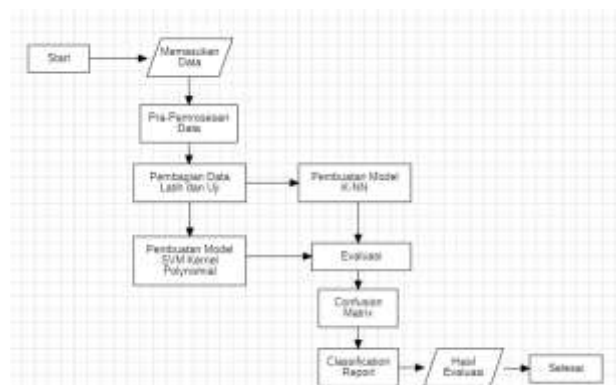
x_i, x_j = Pembanding fitur pada dataset

r = Parameter bias

d = Parameter pangkat

2.5 Alur Penelitian

Alur penelitian tentang kacang kering terdapat pada Gambar 1 dibawah ini



Gambar 1 flowchart alur penelitian

Tahapan pertama pada Alur adalah memasukkan data yang telah didapatkan, setelah data dimasukkan maka akan melakukan pra-pemrosesan data seperti mengisi missing values, setelah missing value terisi data uji dan data latih akan dibagi, setelah pembagian data latih dan data uji untuk pelatihan model *K-Nearest Neighbor* dan *Support Vector Machine*, lalu melakukan evaluasi dalam bentuk *confusion matrix* untuk mendapatkan hasil dari *classification report*.

2.6 Metode Evaluasi

Metode evaluasi yang diggunakan adalah *confusion matrix* dan *classification report*. *Confusion matrix* digunakan untuk mendapatkan nilai dari akurasi, *precision*, *recall* dan *f1-score*. Contoh *confusion matrix* terdapat pada Tabel 1 dibawah [16,17].

Tabel 1 Confusion Matrix

Kelas Prediksi	Kelas Sebenarnya		
		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

Keterangan:

TP = True Positive

FP = False Positive

FN = False Negative

TN = True Negative

Confusion matriks memiliki nilai yang dapat dihitung yang akan masuk ke dalam *classification report* yaitu akurasi, *precision*, *recall*, *f1-score* [18]

Untuk mendapatkan nilai Akurasi rumusnya ada pada persamaan (2) dibawah.

$$akurasi = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

Untuk mendapatkan nilai *precision* rumusnya ada pada persamaan (3) dibawah.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Untuk mendapatkan nilai *recall* rumusnya ada pada persamaan (4) dibawah.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Untuk mendapatkan nilai *f1-score* rumusnya ada pada persamaan (4) dibawah.

$$f1 - score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (6)$$

2.7 Machine Learning

Machine learning merupakan sebuah metode yang dilakukan dengan pembelajaran dari sebuah dataset untuk dilakukan klasifikasi atau prediksi.[19] salah satu metode dari *machine learning* adalah *Supervised learning* merupakan pelatihan sebuah mesin yang diawasi dan diberikan data secara berkala untuk dilatih.[20]

3. HASIL DAN PEMBAHASAN

Pada pengujian dataset kacang kering dengan jumlah data 13611 data dengan tipe numerik dan kelas yang terdiri dari Area, Perimeter, *MajorAxisLength*, *MinorAxisLength*, *AspectRatio*, *Eccentricity*, *ConvexArea*, *EquivDiameter*, *Extent*, *Solidity*, *Roundness*, *Compactness*, *ShapeFactor1*, *ShapeFactor2*, *ShapeFactor3*, *ShapeFactor4*, dan *class*. Area luas dari zona kacang kering dan ukuran pixelnya (Area), keliling dari kacang kering (Perimeter), panjang dari kacang kering (*MajorAxisLength*), lebar dari kacang kering (*MinorAxisLength*), hubungan dari panjang dan lebar kacang kering (*AspectRatio*), bentuk dari kacang kering saat diukur bebrbentuk melengkungnya atau tidak (*Eccentricity*), tingkat kekompleksan bentuk kacang kering (*ConvexArea*), diameter lingkaran dari kacang kering (*EquivDiameter*), daerah yang ditempati oleh kacang kering dalam ratio pixel (*Extent*), Seberapa cembung bentuk kacang kering (*Solidity*), Seberapa bulat bentuk kacang kering (*Roundness*), seberapa padat bentuk kacang kering (*Compactness*), faktor dari bentuk-bentuk kacang kering (*ShapeFactor1-4*), Jenis dari kacang kering (*Class*)

Setelah mengetahui data-data yang akan digunakan dan dilakukan untuk mengklasifikasi proses pertama adalah pra-pemrosesan data dimana akan mengecek apakah ada *missing value* dari dataset. Setelah itu membagi data latih, data uji, *random state* dan kebutuhan penelitian seperti parameter untuk *Support Vector Machine* dan seperti gambar 2 dibawah.

```
test_sizes = [0.1, 0.2, 0.25]
random_states = [24, 48, 53, 54, 55]
kernels = [1.0, 10.0]

for test_size in test_sizes:
    for random_state in random_states:
        for kernel_param in kernels:
            print(f"Test Size: {test_size}, Random State: {random_state} Kernel Parameter: {kernel_param}")

# Bagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)
```

Gambar 2 Pembagian data latih, data uji, random state dan kernel pada metode *Support Vector Machine*

Untuk metode *K-Nearest Neighbor* seperti pada Gambar 3

```
test_sizes = [0.1,0.2,0.25]
random_states = [24,40,53,54,55]
n_neighbors = [3,5,7,9]

for test_size in test_sizes:
    for n_neighbor in n_neighbors:
        for random_state in random_states:
            print(f"Test Size: {test_size}, Random State: {random_state} , Neighbor: {n_neighbor}")

            # Bagi data menjadi data latih dan data uji
            X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)
```

Gambar 3 Pembagian data latih, data uji, random state dan tetangga pada metode *K-Nearest-Neighbor*

Selanjutnya akan membuat model untuk algoritma penelitian yaitu *Support Vector Machine* dan *K-Nearest Neighbor* untuk model *Support Vector Machine* menggunakan *library* yang sudah disediakan *scikit-learn* yaitu *SVC* dengan menggunakan kernel polynomial dan parameter yang sudah ditentukan seperti pada Gambar 4.

```
svm_poly = SVC(kernel='poly', C=kernel_param)
svm_poly.fit(X_train, y_train)
y_pred_poly = svm_poly.predict(X_test)
accuracy_poly = accuracy_score(y_test, y_pred_poly)
print('Akurasi model dengan kernel Polinomial: {0:0.4f}'.format(accuracy_poly))
print('Hasil Classification report:')
print(classification_report(y_test, y_pred_poly))
print('Confusion Matrix dengan metode SVM:')
nama_kelas = y_test.unique().tolist()
cm = confusion_matrix(y_test,y_pred_poly)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels= nama_kelas,
            yticklabels=nama_kelas)
plt.ylabel('Kelas Hasil Prediksi',fontsize=13)
plt.xlabel('Kelas Sebenarnya',fontsize=13)
plt.title('Confusion Matrix SVM Poly',fontsize=17)
plt.show()
```

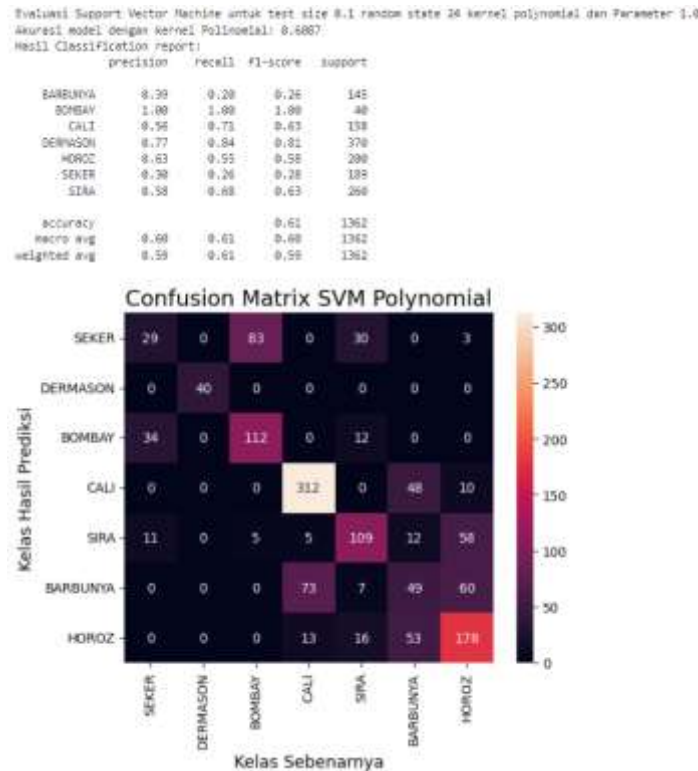
Gambar 4 Model Algoritma Support Vector Machine dengan kernel polinomial

Pada Model *K-Nearest Neighbor* seperti pada Gambar 5

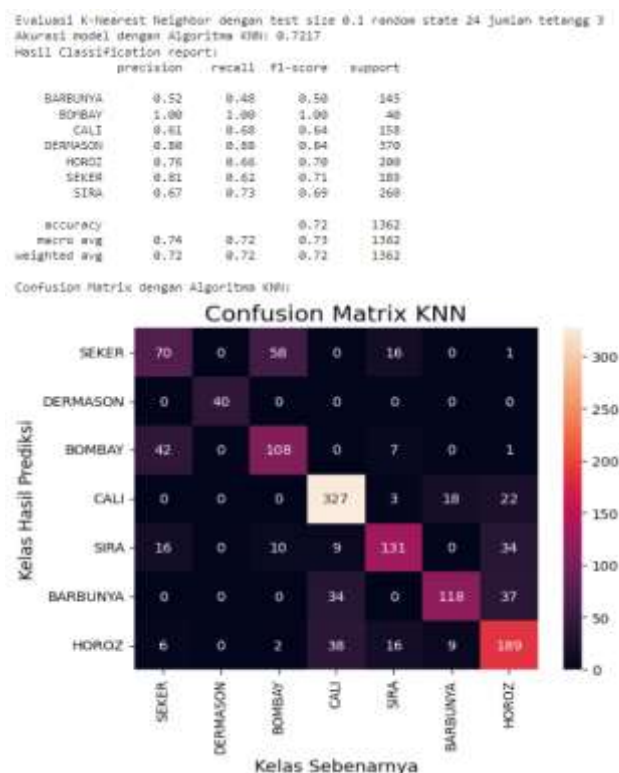
```
knn = KNeighborsClassifier(n_neighbors=n_neighbor)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy_knn = accuracy_score(y_test, y_pred)
print("Akurasi model dengan Algoritma KNN: {0:0.4f}".format(accuracy_knn))
print("Hasil Classification report:")
print(classification_report(y_test, y_pred))
print('Confusion Matrix dengan Algoritma KNN:')
nama_kelas = y_test.unique().tolist()
cm = confusion_matrix(y_test,y_pred)
sns.heatmap(cm,
            annot=True,
            fmt='g',
            xticklabels= nama_kelas,
            yticklabels=nama_kelas)
plt.ylabel('Kelas Hasil Prediksi',fontsize=13)
plt.xlabel('Kelas Sebenarnya',fontsize=13)
plt.title('Confusion Matrix KNN',fontsize=17)
plt.show()
```

Gambar 5 Model algoritma *K-Nearest Neighbor*

Untuk mengetahui performa dari setiap algoritma diperlukanya metode evaluasi yaitu dengan cara menggunakan *confusion matrix* dan *classification* untuk hasil performa setiap algoritma terdapat pada Gambar 6 dan Gambar 7 dibawah.



Gambar 6 Evaluasi Classification Report dan Confusion matrix metode Support Vector Machine



Gambar 7 Evaluasi Classification Report dan Confusion matrix metode K-Nearest Neighbor

Dari Penelitian yang telah dilakukan perbandingan kinerja antara K-Nearest Neighbor dan Support Vector Machine dapat dilihat pada Tabel 2, Tabel 3 dan Tabel 4

Tabel 2. Hasil Rata-Rata pada data uji 10% data latih 90%

Algoritma	Akurasi	Precision	Recall	F1-score
Support Vector Machine kernel Polinomial (Parameter 1.0)	0.632	0.622	0.624	0.614
Support Vector Machine kernel Polinomial (Parameter 10.0)	0.636	0.634	0.634	0.63
K-Nearest Neighbor (tetangga 3)	0.74	0.75	0.734	0.74
K-Nearest Nighbor (tetangga 5)	0.738	0.744	0.73	0.738

Tabel 3. Hasil Rata-Rata pada data uji 20% data latih 80%

Algoritma	Akurasi	Precision	Recall	F1-score
Support Vector Machine kernel Polinomial (Parameter 1.0)	0.626	0.626	0.624	0.616
Support Vector Machine kernel Polinomial (Parameter 10.0)	0.634	0.636	0.632	0.63
K-Nearest Neighbor (tetangga 3)	0.732	0.742	0.726	0.732
K-Nearest Nighbor (tetangga 5)	0.726	0.738	0.724	0.728

Tabel 4. Hasil Rata-Rata pada data uji 25% data latih 75%

Algoritma	Akurasi	Precision	Recall	F1-score
Support Vector Machine kernel Polinomial (Parameter 1.0)	0.626	0.624	0.624	0.618
Support Vector Machine kernel Polinomial (Parameter 10.0)	0.634	0.632	0.63	0.63
K-Nearest Neighbor (tetangga 3)	0.722	0.734	0.718	0.724
K-Nearest Nighbor (tetangga 5)	0.72	0.73	0.716	0.72

Penelitian diatas menunjukan bahwa performa yang bagus dilakukan menggunakan metode metode K-Nearest Neighbor dengan tetangga sebanyak $k = 3$. pada data uji 10% akurasi 0.74 , precision 0.75, recall 0.734, f1-score 0.74. pada data uji 20% akurasi 0.732, precision 0.742, recall 0.726, f1-score 0.732. pada data uji 25% akurasi 0.722, precision 0.734 , recall 0.718, f1-score 0.724

4. KESIMPULAN

Dari penelitian klasifikasi jenis kacang kering dapat disimpulkan bahwa untuk membedakan jenis kacang kering seker, dermason, bombay, cali, sira, barubunya, horaz dapat menggunakan teknik *machine learning*. Hasil yang didapatkan pada penelitian ini juga tidak berbeda jauh tetapi jika ingin mendapatkan hasil yang baik maka membutuhkan nilai evaluasi model yang tinggi. Pada penelitian klasifikasi kacang kering dapat disimpulkan bahwa metode *K-Nearest Neighbor* dengan jumlah tetangga 3 memperoleh nilai evaluasi yang tinggi, tidak hanya akurasi melainkan *precision*, *recall*, *f1-score* yang tinggi dibandingkan dengan metode *Support Vector Machine*

DAFTAR PUSTAKA

- [1] R. R. Sani, Y. A. Pratiwi, S. Winarna, E. D. Udayanti and F. A. Zami, "Analisis Perbandingan Algoritma Naive Bayes Classifier dan Support Vector Machine untuk Klasifikasi Hoax Pada Berita Online Indonesia," *Jurnal Masyarakat Informatika*, vol. 13, pp. 85-98, 2022.
- [2] M. E. Al Rivian, N. Rachmat and M. R. Ayustin, "Klasifikasi Jenis Kacang-Kacangan Berdasarkan Tekstur Menggunakan Jaringan Syaraf Tiruan," *Jurnal Komputer Terapan*, vol. 6 (1), pp. 89-98, 1 mei 2020.
- [3] J. A. Wiesinger, K. A. Cichy, D. S. Hooper, J. J. Hart and R. P. Glahn, "Processing White or yellow dry beans (*Phaseolus vulgaris* L.) into a heat treated flour enhances the iron bioavailability of bean-based pastas," *Journal of Functional Foods*, vol. 71, August 2020.
- [4] N. Diniyah and S. H. Lee, "Komposisi Senyawa Fenol dan Potensi Antioksidan dari Kacang-Kacangan," *Jurnal Agroteknologi*, vol. 14 (01), pp. 91-102, 15 September 2020.
- [5] M. A. Uebersax, A. K. Cichy, E. F. Gomez, G. T. Porch, J. Heitholt, M. J. Osorno , K. Kamfwa, s. S. Snapp and S. Bales, "Dry Beans (*Phaseolus vulgaris* L.) as a vital component of sustainable agriculture and food security," *Legume Science*, vol. 5.1, p. e155, 30 May 2022.
- [6] H. G. Dogan, M. Kan and A. Kan, "Projection of Dry Beans Cultivation Area For Turkey," *Journal of Global Innovations*, vol. 8(4), pp. 195-201, 2020.
- [7] A. P. Mullins and B. H. Arjmandi, "Health Benefits of Plant-Based Nutrition : Focus on Beans Cardiometabolic disease," *Nutrients*, vol. 13.2, p. 519, 5 Febuary 2021.
- [8] T. Lin, O. S. Keefe, S. Duncan and F. C. Fraguas, "Retention of primary bile salts by dry beans (*Phaseolus vulgaris* L.) during in vitro digestion : Role of bean components and effect of food processing," *Food Research International*, vol. 137, no. 109337, November 2020.
- [9] W. Wang, E. M. Wright, M. A. Uebersax and K. Cichy , "A pilot-scale dry bean canning and evaluation protocol," *Journal of Food Processing and Preservation*, vol. 49, no. 9, 24 November 2021.
- [10] E. Elfatimi, R. Erygit and L. Elfatimi, "Beans Leaf Disease Classification Using Mobilenet Models," *IEEE Access* 10, vol. 10, pp. 9471-9482, 13 January 2022.
- [11] A. R. Isnain, J. Supriyanto and M. P. Kharisma, "Implementation of K-Nearest Neighbor (K-NN) Algorithm For Public Sentiment Analysis of Online Learning," *Indonesian Journal of Computing and Cybernetics System*, vol. 15(2), pp. 121-130, April 2021.
- [12] S. R. Cholil, T. Handayani, R. Prathivi and T. Ardianita, "Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) untuk Klasifikasi Seleksi Penerima Beasiswa," *Indonesian Journal on Computer and Information Technology*, vol. 6 (2), pp. 118-127, 13 juli 2021.
- [13] S. Rahayu, Y. MZ, J. E. Bororing and R. Hadiyat, "Implementasi Metode K-Nearest Neighbor (K-NN) Untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP," *Jurnal Pendidikan Informatika Edumatic*, vol. 6 no 1, pp. 98-106, 1 Juni 2022.
- [14] A. Setianingrum, A. Hindayanti, D. M. Cahya and D. S. Purnia, "Perbandingan Metode Algoritma K-NN & Metode Algoritma C4.5 Pada Analisa Kredit Macet (Studi Kasus PT Tugmung Textile Bintan)," *Evolusi : Jurnal Sains dan Manajemen*, vol. 9 No 2, 2 September 2021.
- [15] S. S. Fadli and M. Ashari, "Optimization of Support Vector Machine Method Using Feature Selection to Improve Classification Result," *JISA (Jurnal Informatika dan Sains)*, vol. 04 no 1, pp. 22-27, 1 June 2021.
- [16] F. P. Anindya, D. E. Herwindiati and N. J. Perdana, "Pengenalan Suara Manusia Menggunakan Support Vector Classifier (SVC) Untuk Proses Otentikasi," *Computatio: Journal of Computer Science and Information System*, vol. 7.1, pp. 28-36, 2023.

- [17] D. Krstinic, M. Braovic, L. Seric and D. B. Stulic, "Multi-Label classifier performance evaluation with confusion matix," *Computer Science & Information Technology*, vol. 1, pp. 1-14, 2020.
- [18] Z. D. Vunjovic, "Classification Model Evaluation Metrics," *(IJACSA) International Journal of Advanced Computer Science and Application*, vol. 12, no. 6, pp. 599-606, 2021.
- [19] A. Roihan, P. A. Sunarya and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang," *IJICT (Indonesian Journal on Computer and Information Technology)*, vol. 5 (1), pp. 75-82, 22 April 2020.
- [20] P. Santoso, H. Abijono and N. L. Angggreini, "Algoritma Supervised Learning Dan Unsupervised Learning Dalam Pengolahan Data," *G-Tech: (Jurnal Teknologi Terapan)*, vol. 4 No 2, pp. 315-318, April 2021.