

Pendekatan Deep Learning Untuk Klasifikasi Kematangan Tempe Mendoan Menggunakan *Convolutional Neural Network*

Nuke L Chusna¹⁾, Ahmad Rizqi Sampoerna²⁾, Ninuk Wiliani³⁾

¹⁾²⁾ Teknik Informatika, Fakultas Teknik, Universitas Krisnadwipayana, Jakarta

¹⁾ nukelchusna@unkris.ac.id

²⁾ arsampoerno@gmail.com

³⁾ Teknik Informatika, Fakultas Teknik, Universitas Pancasila, Jakarta

³⁾ ninuk.wiliani@univpancasila.ac.id

Abstrak

Tempe mendoan dikenal dengan makanan yang memiliki kematangan yang berbeda dalam tiap jenisnya. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi tingkat kematangan tempe mendoan menggunakan algoritma *Convolutional Neural Network* (CNN). Dataset yang digunakan terdiri dari 400 data citra tempe mendoan yang dikategorikan ke dalam empat level kematangan: Level 1 (6 jam pertama), Level 2 (12 jam), Level 3 (18 jam), dan Level 4 (24 jam). Berbagai arsitektur CNN diuji dalam penelitian ini, dan hasil terbaik diperoleh menggunakan arsitektur VGG16 dengan nilai AUC sebesar 0,94 atau 95%, menunjukkan kemampuan klasifikasi yang sangat baik. Sistem ini dirancang untuk membantu produsen, seperti karyawan dan penjual tempe mendoan, dalam menentukan tingkat kematangan tempe secara tepat. Dengan sistem ini, tempe yang dihasilkan memiliki kualitas kematangan optimal, sehingga dapat meningkatkan daya tarik produk dan minat konsumen. Penelitian ini memberikan kontribusi pada penerapan teknologi berbasis *deep learning* untuk meningkatkan kualitas produksi dalam industri makanan tradisional.

Kata kunci: *Convolutional Neural Network*, Klasifikasi Kematangan, Tempe Mendoan, *Deep Learning*, Pengolahan Citra

Abstract

Tempe mendoan is known as a food that has different levels of doneness in each type. This study seeks to establish a classification system for the ripeness of mendoan tempe via the Convolutional Neural Network method. (CNN). The dataset comprises 400 photos of mendoan tempe classified into four ripeness levels: Level 1 (initial 6 hours), Level 2 (12 hours), Level 3 (18 hours), and Level 4 (24 hours). This study evaluated multiple CNN architectures, with the VGG16 architecture yielding the highest performance, achieving an AUC value of 0.95, indicating exceptional classification abilities. This system is intended to aid producers, including employees and vendors of tempe mendoan, in precisely assessing the ripeness level of tempe. This technique yields tempeh of ideal ripeness quality, consequently augmenting product appeal and consumer interest. This research enhances the application of deep learning technology to elevate production quality in the conventional food business.

Keywords: *Convolutional Neural Network, Maturity Classification, Tempe Mendoan, Deep Learning, Image Processing.*

1. PENDAHULUAN

Tempe mendoan merupakan salah satu makanan tradisional Indonesia yang memiliki tingkat kematangan beragam sesuai preferensi konsumen [1]. Klasifikasi tingkat kematangan ini penting untuk menjaga konsistensi kualitas produk, terutama dalam industri kuliner dan manufaktur makanan [2]. Teknologi *image processing* menawarkan solusi otomatis untuk mengatasi tantangan mengklasifikasikan kematangan tempe mendoan [3]. Masalah yang ditemukan adalah masih ditemukannya kesulitan dalam menentukan atau membedakan tingkatan kematangan tempe mendoan dimana terdapat kesamaan bentuk dari kedele yang melewati proses fermentasi tingkat kematangan tempe [4].

Dengan dukungan algoritma *deep learning*, seperti *Convolutional Neural Network* (CNN), proses analisis dan klasifikasi menjadi lebih efisien serta akurat [5]. CNN dikenal memiliki performa tinggi dalam mengenali pola dari data visual. Penelitian ini bertujuan mengembangkan model CNN yang dapat mengklasifikasikan tingkat kematangan tempe mendoan berdasarkan citra digital [6]. Tujuan dari penelitian ini adalah melakukan proses klasifikasi pada tingkat kematangan tempe mendoan dengan menggunakan arsitektur terbaik dari Algoritma *Convolution Neural Network* (CNN) [7].

2. TINJAUAN PUSTAKA

Klasifikasi tingkat kematangan tempe adalah cara untuk menilai sejauh mana tempe telah mencapai tingkat kesiapan atau kualitas tertentu [1]. Kematangan tempe umumnya dilihat dari beberapa aspek, termasuk tekstur, rasa, aroma, dan kandungan nutrisi [2]. Beberapa cara umum untuk mengklasifikasikan tingkat kematangan tempe melibatkan:

1. Ketebalan Tempe: Tempe yang lebih tipis mungkin belum mencapai tingkat kematangan yang ideal, sementara tempe yang lebih tebal bisa dianggap lebih matang. Ketebalan tempe ini sering dikaitkan dengan kandungan udara di dalam tempe.
2. Warna Tempe: Tempe yang sudah matang biasanya memiliki warna yang lebih merata dan cerah. Warna yang tidak merata atau warna yang masih terlalu pucat bisa mengindikasikan kematangan yang kurang.

Penelitian sebelumnya menggunakan algoritma *deep learning* untuk klasifikasi citra menunjukkan potensi besar CNN dalam menghasilkan akurasi tinggi [8]. Penelitian ini membandingkan SVM, KNN, dan CNN untuk klasifikasi citra cuaca. Hasil penelitian menunjukkan CNN memiliki performa terbaik dengan akurasi mencapai 94,2% setelah 50 epoch, meskipun membutuhkan waktu eksekusi yang lebih lama dibandingkan algoritma lain. Perbedaan utama antara penelitian tersebut dan penelitian ini adalah jenis dataset yang digunakan, di mana penelitian sebelumnya menggunakan dataset cuaca, sedangkan penelitian ini berfokus pada klasifikasi tingkat kematangan tempe mendoan. Penelitian kedua membandingkan algoritma CNN dan MLP pada dataset American Sign Language [7]. Hasil penelitian menunjukkan bahwa CNN menghasilkan akurasi terbaik sebesar 96,97% dengan *preprocessing* menggunakan *Gaussian LPF*, dibandingkan dengan MLP yang mencapai akurasi maksimal sebesar 74,79%. Perbedaan mendasar dengan penelitian ini terletak pada sumber data, di mana penelitian sebelumnya menggunakan dataset yang tersedia di Kaggle.com, sedangkan penelitian ini menggunakan dataset yang dikumpulkan langsung melalui kamera *smartphone*, sehingga menimbulkan tantangan tersendiri terkait variasi kualitas citra. Penelitian selanjutnya [9] melakukan komparasi algoritma KNN, SVM, dan CNN untuk klasifikasi citra alfabet Sistem Isyarat Bahasa Indonesia (SIBI). Penelitian tersebut menunjukkan bahwa CNN dengan arsitektur *Xception* menghasilkan F1 Score terbaik sebesar 99,57%, namun dengan waktu komputasi yang cukup lama. Dalam penelitian ini, fokus utamanya bukan hanya pada performa model, tetapi juga pada hasil klasifikasi yang dapat membantu pengguna menentukan tingkat kematangan tempe mendoan yang diinginkan.

Ketiga penelitian tersebut memberikan wawasan mengenai keunggulan CNN dalam klasifikasi citra, namun berbeda dalam konteks dataset dan tujuan akhir. Penelitian ini menawarkan kontribusi dengan fokus pada CNN untuk klasifikasi tingkat kematangan tempe mendoan, yang belum banyak dieksplorasi sebelumnya

3. METODE PENELITIAN

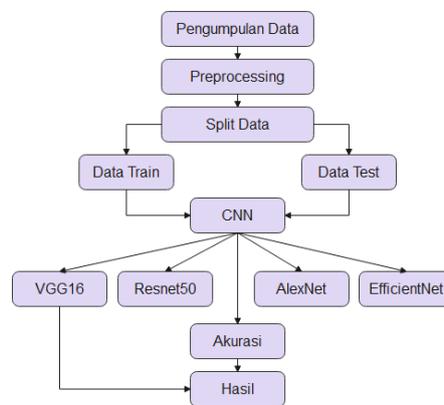
Data yang didapatkan melalui pengambilan data secara langsung oleh penulis yang melakukan penelitian dan observasi langsung pada proses fermentasi tempe mendoan dan Data yang digunakan untuk penelitian ini adalah data sekunder yang diperoleh secara langsung menggunakan kamera *smartphone* dengan proses yang telah dijelaskan sebelumnya.



Gambar 1. Dataset tempe dari level 1 sampai level 4

Gambar 1 menggambarkan tingkat kematangan tempe mendoan yang digunakan sebagai dataset dalam penelitian ini:

1. Level 1 (sebelum 6 jam): Kacang kedelai masih utuh tanpa jamur yang tumbuh. Suhu fermentasi optimal pada 30–40°C, dengan kadar air yang menyusut. Pada tahap ini, tempe belum mengalami perubahan signifikan dan belum siap untuk diolah.
2. Level 2 (12 jam): Benang-benang halus (hifa) mulai tumbuh di permukaan kedelai, membentuk struktur awal tempe. Zat adhesif yang dihasilkan oleh kapang membantu mengikat kedelai menjadi lebih padat, meskipun hifa belum terlihat jelas secara fisik.
3. Level 3 (18 jam): Jamur mulai menyelimuti kedelai, membentuk lapisan tipis yang dapat diamati. Namun, lapisan ini masih rentan sobek jika tempe digoreng. Tahap ini menunjukkan tempe yang hampir matang namun belum optimal.
4. Level 4 (24 jam): Jamur atau kapang telah sepenuhnya menutupi permukaan kedelai, menghasilkan enzim yang meningkatkan rasa, tekstur, dan nilai gizi tempe. Fermentasi menghasilkan rasa asam khas, dan tempe pada tahap ini sudah matang dan siap untuk diolah atau disajikan.



Gambar 2. Rancangan penelitian

Gambar 2 menjelaskan proses jalannya penelitian, mulai dari awal pengambilan gambar yang akan digunakan untuk dataset sampai proses pengklasifikasian menggunakan algoritma CNN [5]. Sebelum menerapkan algoritma CNN pada model, data dilakukan sebuah pre processing yaitu : merubah gambar RGB ke grayscale, resize gambar menjadi ukuran yang sama, yaitu 40x40 dan merubah gambar menjadi array, proses ini ditampilkan di dalam Gambar 3.

```
[6] # perulangan pembacaan image
for image_path in image_paths:
    # resize gambar menjadi berukuran 40x40 pixel
    image = cv2.imread(image_path)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) #Gambar diubah dari RGB ke Grayscale
    image = cv2.resize(image,(40,40))
    image = img_to_array(image)
    data.append(image)

    # mengekstrak label dari gambar
    label = image_path.split(os.path.sep)[-2]
    label=all_class.index(label)
    labels.append(label)

    # update proses pada progress bar
    pbar.update(1)

toc = time.time()
# menampilkan jumlah total waktu yang diperlukan untuk ekstraksi fitur
print("The total time taken: {:.2f}sec".format(toc-tic))
print("Done")

100% ██████████ 400/400 [04:01<00:00, 1.83it/s]
The total time taken: 241.68sec
```

Gambar 3. Preprocessing data

Setelah dilakukan preprocessing data, dilakukan proses pembagian data, yaitu sebesar 90:10 untuk data latih sebesar 90% dan data testing sebesar 10%. Banyaknya data yang akan digunakan sebanyak 360 data dan data testing sebanyak 40 data. Proses pembagian data ditunjukkan pada Gambar 4.

```
[9] # memisahkan data train dan data test dengan komposisi 0.1 data test sehingga data train nya 0.9
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.10, random_state=42)

print(X_train.shape,"X_train")
print(y_train.shape,"y_train")
print(X_test.shape, "X_test")
print(y_test.shape, "y_test")

(360, 40, 40, 3) X_train
(360,) y_train
(40, 40, 40, 3) X_test
(40,) y_test
```

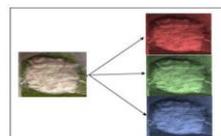
Gambar 4. Proses pembagian data

Setelah itu dilanjutkan skenario penerapan CNN model VGG-16 terdapat pada Gambar 5, dengan memasukkan gambar ke dalam proses CNN. Ukuran gambar yang dimasukkan adalah 40 x 40 x 3. Angka 3 yang dimaksud adalah angka dari RGB (*red, green, blue*) seperti pada Gambar 6.

```
[13] # menampilkan rangkuman dari model
cnn.summary()

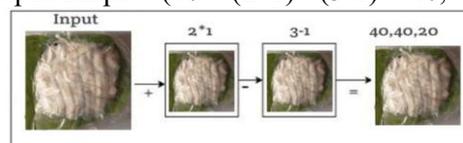
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)             (None, 40, 40, 20)       1520
activation (Activation)     (None, 40, 40, 20)       0
max_pooling2d (MaxPooling2D) (None, 20, 20, 20)       0
conv2d_1 (Conv2D)           (None, 20, 20, 50)       25050
activation_1 (Activation)   (None, 20, 20, 50)       0
max_pooling2d_1 (MaxPooling2D) (None, 10, 10, 50)       0
flatten (Flatten)           (None, 5000)              0
dense (Dense)                (None, 500)               2500500
activation_2 (Activation)   (None, 500)               0
dense_1 (Dense)              (None, 4)                 2004
activation_3 (Activation)   (None, 4)                 0
-----
Total params: 2529074 (9.65 MB)
Trainable params: 2529074 (9.65 MB)
Non-trainable params: 0 (0.00 Byte)
```

Gambar 5. Skenario algoritma *convolution neural network*



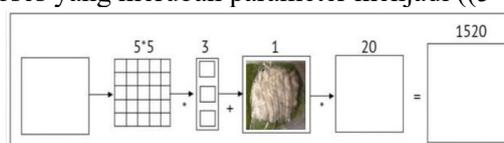
Gambar 6. RGB image

Gambar 7 menjelaskan proses gambar yang dimasukkan kedalam proses CNN sistem akan melakukan proses VGG-16 dimana pada conv2d (CONV2D) diperoleh nilai param sebesar 1520 dengan cara perhitungan Output Shape = $(40 + (2*1) - (3-1)) = 40,40,20$.



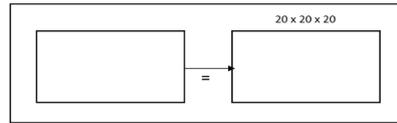
Gambar 7. Perhitungan output

Gambar 8 menjelaskan proses yang merubah parameter menjadi $((5*5*3)+1)*20 = 1520$;



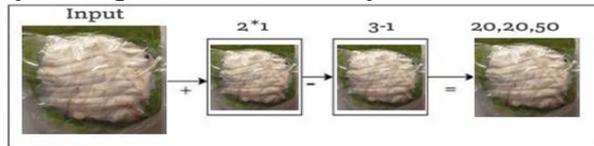
Gambar 8. Perubahan parameter

Gambar 9. merupakan menggunakan *maxpooling* dan memperoleh hasil $20 \times 20 \times 20$.



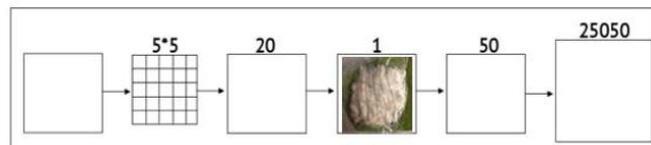
Gambar 9. Maxpooling

Gambar 10 merupakan perhitungan kembali dan didapatkan hasil sebesar 25.050



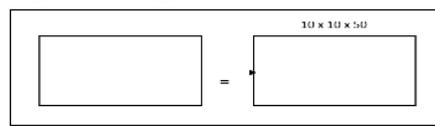
Gambar 10. Layer ke dua

Gambar 11 merupakan langkah untuk merubah kembali *pixel* dalam image



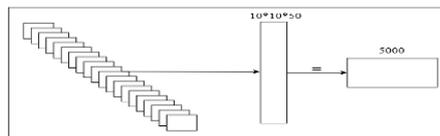
Gambar 11. Perubahan *pixel image*

Gambar 12 dilakukan *maxpoling* dan menghasilkan $10 \times 10 \times 50$.



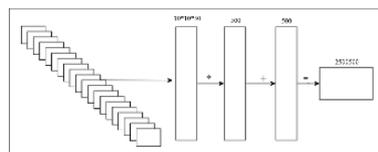
Gambar 12. Hasil $10 \times 10 \times 50$

Gambar 13 dilakukan *dense* sebesar 5000, dengan cara perhitungan $10 \times 10 \times 50 = 5000$.



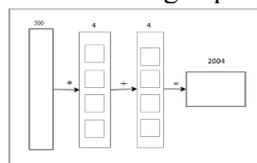
Gambar 13. Langkah 6 CNN VGG-16

Gambar 14 dilakukan *dense* dengan cara perhitungan $(5000 \times 500) + 500 = 2500500$.



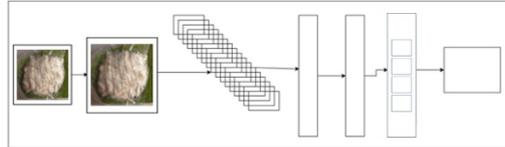
Gambar 14. Dense

Gambar 15 kemudian diberikan *layer* kembali dengan penghitungan: $(500 \times 4) + 4 = 2004$.



Gambar 15. Hasil Layer ketiga

Pada Gambar 16, empat kotak merupakan banyaknya kelas yaitu *level 1* (6 jam pertama), *level 2* (12 jam), *level 3* (18 jam), dan *level 4* (24 jam). Gambar 14 mempresentasikan hasil yang sudah dihitung secara keseluruhan pada langkah sebelumnya dengan perhitungan: $1520 + 25050 + 2500500 + 2004 = 2529074$.



Gambar 16. Hasil akhir dari klasifikasi

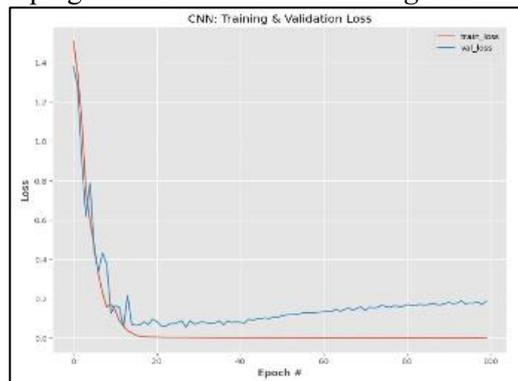
4. HASIL DAN PEMBAHASAN

Setelah selesai membuat pemodelan arsitektur untuk mengklasifikasi dengan metode CNN dan dilanjut dengan mengkompilasi model yang sudah dibuat dan melatih dataset seperti pada Gambar 17.

```
# plot validasi training dan validation loss
N = np.arange(0, 100)
plt.style.use("ggplot")
plt.figure(figsize = [10,8])
plt.plot(N, model.history["loss"], label="train_loss")
plt.plot(N, model.history["val_loss"], label="val_loss")
plt.title("CNN: Training & Validation Loss")
plt.xlabel("Epoch #", weight="bold")
plt.ylabel("Loss", weight="bold")
plt.legend()
plt.show()
```

Gambar 17. Proses validasi training dan validation loss

Gambar 18 menghasilkan pergerakan besaran nilai *training* dan *validation*.

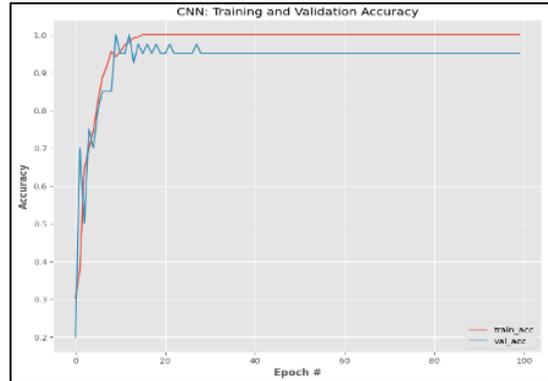


Gambar 18. Hasil proses validasi training dan validation Loss

```
# plot the training and validation accuracy
N = np.arange(0, 100)
plt.style.use("ggplot")
plt.figure(figsize = [10,8])
plt.plot(N, model.history["accuracy"], label="train_acc")
plt.plot(N, model.history["val_accuracy"], label="val_acc")
plt.title("CNN: Training and Validation Accuracy")
plt.xlabel("Epoch #", weight="bold")
plt.ylabel("Accuracy", weight="bold")
plt.legend()
plt.show()
```

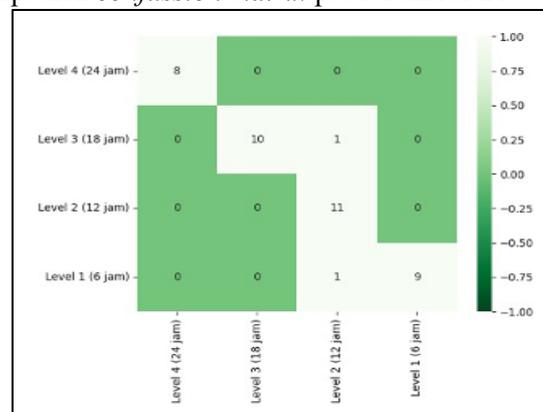
Gambar 19. Proses validasi training dan validation accuracy

Gambar 19 merupakan proses untuk melihat pergerakan besaran nilai *training* dan *validation*.



Gambar 20. Hasil proses *validasi training* dan *validation accuracy*

Berdasarkan hasil training, maka diperoleh model, dan setelah dilakukan *testing* dan validasi maka diketahui atau didapatkan *confussion matrix* pada Gambar 21.



Gambar 21. *Confussion Matrix* vgg16

Berdasarkan hasil dari Gambar 21 dapat dianalisis untuk mendapatkan nilai *Area Under Curve* (AUC) yang di uji. Dalam pengujian *Confussion matrix* ini menggunakan data training sebanyak 360 data dan data testing sebanyak 40 data, dengan penjelasan sebagai berikut:

1. Tingkatan *level 1* (6 jam) dapat diprediksi benar sebanyak 9 kali dalam 9 kali percobaan.
2. Tingkatan *level 2* (12 jam) dapat diprediksi benar sebanyak 11 kali dalam 13 kali percobaan.
3. Tingkatan *level 3* (18 jam) dapat diprediksi benar sebanyak 10 kali dalam 10 kali percobaan
4. Tingkatan *level 4* (24 jam) dapat diprediksi benar sebanyak 8 kali dalam 8 kali percobaan.

Hasil akurasi dari klasifikasi menggunakan algoritma CNN dapat dilihat pada gambar berikut ini:

```
[18] #Plot classification report precision, recall, f1-score dan support
print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

     0         1.00      1.00      1.00         8
     1         1.00      0.91      0.95        11
     2         0.85      1.00      0.92        11
     3         1.00      0.90      0.95        10

 accuracy          0.96
 macro avg          0.96
 weighted avg          0.96

[19] #menghitung akurasi antara data test dengan data prediksi
cnn_accuracy = accuracy_score(y_test, y_pred)
print(f"The Model Accuracy = {cnn_accuracy*100:.2f}%")

The Model Accuracy = 95.00%
```

Gambar 22. Performance dan hasil akurasi

- *Precision* adalah tingkat ketepatan antara informasi dengan jawaban yang diberikan oleh sistem.
- *Recall* adalah tingkat keberhasilan sistem dalam menemukan sebuah informasi.
- *F1-Score* adalah menghitung rata – rata dari ketepatan dan keberhasilan dari sistem.

- *Support* adalah jumlah sampel pada kelas tersebut.
- *Accuracy* adalah jumlah data yang benar pada keseluruhan data.
- *Macto AVG* adalah menghitung rata rata pada setiap kelas.
- *Weighted AVG* adalah menghitung nilai bobot dari setiap kelas.

Berdasarkan hasil klasifikasi menggunakan confusion matrix, model VGG16 mencapai tingkat akurasi sebesar 0.95 atau 95%. Nilai akurasi ini mencerminkan sejauh mana model berhasil mengklasifikasikan data dengan benar *Confusion matrix* akan di sederhanakan ke dalam bentuk Tabel 1.

Tabel 1. *Actual* dan *prediction* dari *confusion matrix*

Class	Actual	Predicted			
		Level 1 (6 Jam)	Level 2 (12 Jam)	Level 3 (18 jam)	Level 4 (24 Jam)
Level 1 (6 jam)	8	8	0	0	0
Level 2 (12 Jam)	11	0	10	1	0
Level 3 (18 Jam)	11	0	0	11	0
Level 4 (24 Jam)	10	0	0	1	9
Total			40		

Dengan menganalisa Tabel 1 nilai kinerja model dalam pengidentifikasian contoh secara akurat untuk setiap level tingkat kematangan. Ini memberikan pengetahuan tentang seberapa baik kinerja model untuk kelas yang berbeda dan dapat membantu mengevaluasi model keefektifan dalam mendeteksi tingkat kematangan tempe mendoan. Dalam penilaian kinerja model prediksi, penting untuk menggunakan metrik evaluasi seperti *Area Under the Curve* (AUC) untuk mengevaluasi sejauh mana model tersebut dapat membedakan antara kelas positif dan kelas negatif. Agar dapat memahami hasil analisa dari confusion matrix dapat di sederhanakan menggunakan elemen-elemen yang menggambarkan hasil klasifikasi yang benar sebagai berikut pada Tabel 2.

Tabel 2. Elemen dari *confusion motrix*

Class	True	False	False	True
	Positive (TP)	Positive (FP)	Negative (FN)	Negatif (TN)
Level 1 (6 Jam)	9	1	1	30
Level 2 (12 Jam)	11	0	0	27
Level 3 (18 Jam)	10	1	1	29
level 4 (24 Jam)	8	1	1	30

Pada Tabel 2 terdapat data tentang prediksi model terhadap 4 (empat tingkat kematangan). Setiap tingkat kematangan memiliki kolom *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). AUC yang digunakan untuk mengukur kemampuan model dalam membedakan antara tempe mendoan yang sudah matang (kelas positif) dan tempe mendoan yang belum matang (kelas negatif). Dalam hal ini, AUC dapat menghitung untuk setiap jenis kerusakan jalan menggunakan data dari tabel. Untuk menghitung AUC dibutuhkan nilai *Recall* dan *Specificity*. Nilai *recall* dan *Specificity* ditambahkan kemudian dibagi empat untuk mendapatkan nilai AUC.

$$Recall = \frac{0,9+1+0,9+0,8}{4} = \frac{3,6}{4} = 0,9$$

$$Specifisitas = \frac{0,97+1+0,9+0,97}{4} = \frac{3,91}{4} = 0,97$$

$$Nilai AUC = \frac{Nilai Recall+Nilai Specificity}{2} = \frac{0,9 + 0,97}{2} = 0,935$$

Nilai AUC ini memiliki kinerja rata-rata yang seluruh kelas dalam tugas klasifikasi. Nilai AUC yang lebih tinggi menunjukkan model yang lebih baik, karena menandakan peningkatan kemampuan membedakan antara instansi positif dan negatif. Nilai AUC yang dihasilkan dari *confusion matrix* ini terhitung sebesar 0,935 dibulatkan menjadi 0,94. Setelah dilakukan beberapa penelitian terhadap gambar tingkat kematangan tempe mendoan, maka dapat disimpulkan beberapa hal berikut:

Tabel 3. Hasil implementasi

No	Gambar Tempe Mendoan	Hasil Uji
1	 tingkatan level 3	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 95%
2	 tingkatan level 1	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 97%
3	 tingkatan level 4	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 93%
4	 tingkatan level 1	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 95%
5	 tingkatan level 3	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 96%
6	 tingkatan level 2	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 90%
7	 tingkatan level 4	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 94%
8	 tingkatan level 1	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 95%
9	 tingkatan level 2	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem gagal memprediksi secara benar, gambar tersebut adalah tingkatan Level 2 (12 jam) sistem memprediksi sebagai tingkatan Level 3 (18 jam) dengan akurasi sekitar 0,89 atau 89%
10	 tingkatan level 3	Pada saat uji coba sistem dengan menggunakan gambar disamping, sistem berhasil memprediksi secara benar dengan akurasi sekitar 93%

Pada tabel 3 terlihat dari 10 gambar yang dilakukan uji coba, ada 9 gambar berhasil di prediksi dengan baik. Dengan begitu dapat dihasilkan akurasi sebesar 0,9 atau 90%.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan untuk klasifikasi tingkat kematangan tempe mendoan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Algoritma CNN telah berhasil digunakan untuk proses klasifikasi tingkat kematangan tempe mendoan dengan menggunakan data sebanyak 400 data yang terbagi menjadi 4 (empat) jenis tingkatan yaitu *Level 1* (6 jam pertama), *Level 2* (12 jam), *Level 3* (18 jam), dan *Level 4* (24 jam). Dari hasil keempat arsitektur yang telah diuji di indikasikan bahwa model arsitektur VGG16 dengan hasil nilai AUC yaitu 0,5 sehingga memiliki kemampuan membedakan yang cukup baik, serta performa akurasi yang telah didapatkan yaitu sebesar 0,95 atau 95%.
2. Sistem tersebut dapat dibilang cukup membantu sebagai bahan pertimbangan para produsen yaitu karyawan dan penjual tempe mendoan yang masih tidak bisa atau bingung dalam menentukan tingkat kematangan tempe mendoan yang tepat, sehingga tempe yang dihasilkan merupakan tempe dengan tingkat kematangan terbaik dan memudahkan mempercantik produk sehingga meningkatkan minat pembeli dalam penjualan tempe mendoan.

DAFTAR PUSTAKA

- [1] D. Aulia, N. Mistriani, B. Destian Nugraha, S. Wati Bunawolo, and U. Faruq, "Strategi Increasing Revenue and Product Promotion Of Tempe Home Industri Erowari Semarang," 2024.
- [2] I. Istiadi, A. Y. Rahman, and A. D. R. Wisnu, "Identification of tempe fermentation maturity using principal component analysis and k-nearest neighbor," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 7, no. 1, pp. 286–294, 2023.
- [3] N. Wiliani, R. Hesnananda, N. S. Rahmawati, and E. H. Prianggara, "Application Of Machine Learning For Bitcoin Exchange Rate Prediction Againts US Dollar," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 7, no. 2, pp. 67–74, Feb. 2022, doi: 10.33480/jitk.v7i2.2880.
- [4] M. Irfak and A. Y. Rahman, "Support Vector Machine Application for Classification of Tempe Fermentation Maturity with Information Gain Selection Feature," *Edutran Computer Science and Information Technology*, vol. 1, no. 2, pp. 1–9, 2023.
- [5] A. Hosseini *et al.*, "A mobile application based on efficient lightweight CNN model for classification of B-ALL cancer from non-cancerous cells: A design and implementation study," *Inform Med Unlocked*, vol. 39, p. 101244, 2023.
- [6] T. W. Qurâ, "Implementasi Metode Convolutional Neural Network (CNN) untuk Klasifikasi Motif pada Citra Sasirangan," *Madani: Jurnal Ilmiah Multidisiplin*, vol. 1, no. 7, 2023.
- [7] M. F. Naufal *et al.*, "Analisis Perbandingan Algoritma Klasifikasi MLP dan CNN pada Dataset American Sign Language," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 3, pp. 489–495, Jun. 2021, doi: 10.29207/resti.v5i3.3009.
- [8] "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 2, pp. 311–318, Mar. 2021, doi: 10.25126/jtiik.2021824553.
- [9] M. Farid Naufal, S. Ferdiana Kusuma, and P. Korespondensi, "Analisis Perbandingan Algoritma Machine Learning dan Deep Learning Untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 10, Jul. 2023, doi: 10.25126/jtiik.2023106828.

Biodata Penulis

Nuke L Chusna, merupakan dosen tetap di Universitas Krisnadwipayana, program studi Teknik Informatika, Jakarta

Ahmad Rizqi Sampoerna, merupakan mahasiswa Universitas Krisnadwipayana yang mengambil program studi Teknik Informatika yang tertarik dengan bidang ilmu computer vision.

Ninuk Wiliani, merupakan dosen tetap di Universitas Pancasila, program studi Teknik Informatika, Jakarta