

# Klasifikasi Kualitas Buah dengan Menggunakan Convolutional Neural Network (Studi Kasus: Dataset Fresh and Rotten Classification)

I Gede Diva Dwijayana<sup>a1</sup>, I Putu Fajar Tapa Mahendra<sup>a2</sup>, Ivan Luis Simarmata<sup>a3</sup>,  
Gst. Ayu Vida Matrika Giri<sup>a4</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Udayana, Bali  
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia

<sup>1</sup>dwijayanaigede@gmail.com

<sup>2</sup>ftapamahendra@gmail.com

<sup>3</sup>ivan\_luis030602@protonmail.com

<sup>4</sup>vida@unud.ac.id

## Abstract

*This research aims to develop a deep learning model for fruit quality classification using Convolutional Neural Network (CNN) with the Fresh and Rotten Classification dataset. Two CNN models are compared, with the first model serving as the baseline and the second model resulting from parameter tuning based on the first model. The results indicate that increasing the number of epochs improves the model accuracy, as evidenced by the first model achieving 91% accuracy with 10 epochs and 93% accuracy with 15 epochs. Similar patterns are observed in the second model, with 87% accuracy at 10 epochs and 90% accuracy at 15 epochs. Despite the second model involving the addition of layers and parameters, its accuracy tends to be lower compared to the first model. The research emphasizes that increasing the number of epochs enhances model performance, while adding layers does not always lead to significant improvements, depending on the model's complexity and dataset characteristics. The first model, trained with 15 epochs, demonstrates the highest accuracy, approaching results from similar previous studies. This evaluation provides valuable insights for developing a CNN-based fruit classification model on the Fresh and Rotten Classification dataset.*

**Keywords:** Fruit Classification, Rotten, Fresh, Convolutional Neural Network, Accuracy, Epochs

## 1. Pendahuluan

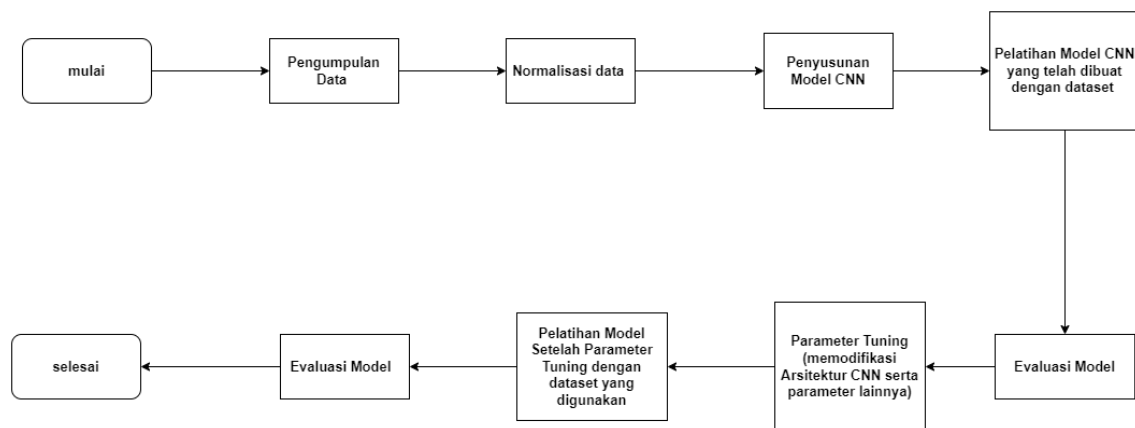
Mendeteksi kualitas buah dalam industri pangan merupakan hal yang utama, tidak hanya untuk menjaga kesehatan konsumen tetapi juga untuk mempertahankan stabilitas ekonomi dan mempertahankan praktik yang berkelanjutan. Dalam era teknologi canggih saat ini, kehadiran kecerdasan buatan, *deep learning*, dan *machine learning* dapat membantu mengatasi masalah ini. Data gambar dari buah-buahan dapat digunakan untuk membangun suatu alat yang mumpuni dalam mengklasifikasikan kualitas dari buah-buahan. Diantara arsitektur *deep learning* yang ada, *Convolutional Neural Network (CNN)* adalah metode yang terkenal dalam melaksanakan tugas yang menggunakan data gambar. *CNN* adalah arsitektur *deep learning* yang didesain untuk dapat mempelajari dan beradaptasi dengan fitur yang ada pada gambar. Alasan ini membuat metode ini sangat cocok dalam mendeteksi ciri dari busuknya suatu buah yang seringkali terlewat menggunakan pengamatan mata manusia. *CNN* dapat dilatih untuk mengenali perbedaan antara buah busuk dan buah segar, membantu mengurangi ketergantungan pada pengamatan manusia yang rentan terhadap subjektivitas dan kesalahan. Beberapa penelitian terdahulu telah menunjukkan potensi *CNN* dalam klasifikasi gambar, seperti penelitian Atul Sharma pada tahun 2021 yang mencapai 94% akurasi validasi dalam klasifikasi gambar menggunakan *dataset* cifar-10 keras[1]. Penelitian lain yang dilakukan oleh Nur Ibrahim, dkk pada tahun 2022 yang melakukan klasifikasi tingkat kematangan pucuk daun teh dengan pendekatan *deep learning* arsitektur VGG19 dan ResNet50 yang mendapatkan

akurasi terbaik dari model VGG19 yaitu sebesar 97,5% [2]. Penelitian terkait yang juga dilakukan oleh Parab pada tahun 2020 yang melakukan klasifikasi sel darah merah menggunakan metode *CNN* dimana model dapat mengklasifikasikan 9 kelas sel darah merah dengan akurasi sebesar 98,6% [3]. Dengan mempertimbangkan hasil penelitian terdahulu, penelitian ini bertujuan untuk mengembangkan model yang mampu mengklasifikasikan buah busuk dan buah segar dengan metode *CNN*. Melalui pendekatan ini, diharapkan dapat meningkatkan keamanan pangan dan memberikan kontribusi pada efisiensi operasional dalam industri pangan.

## 2. Metode Penelitian

Metode yang akan digunakan dalam penelitian ini adalah *Convolutional Neural Network* untuk mengklasifikasi kelas buah busuk ataupun buah segar dengan arsitektur *CNN* yang disusun oleh penulis.

### 2.1. Alur Penelitian

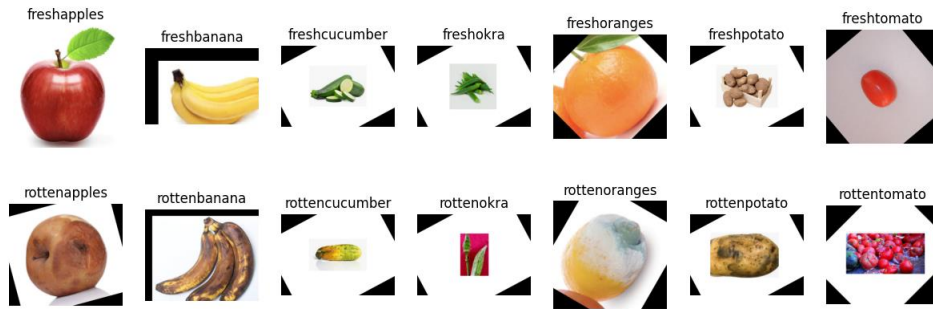


Gambar 1. Alur Penelitian

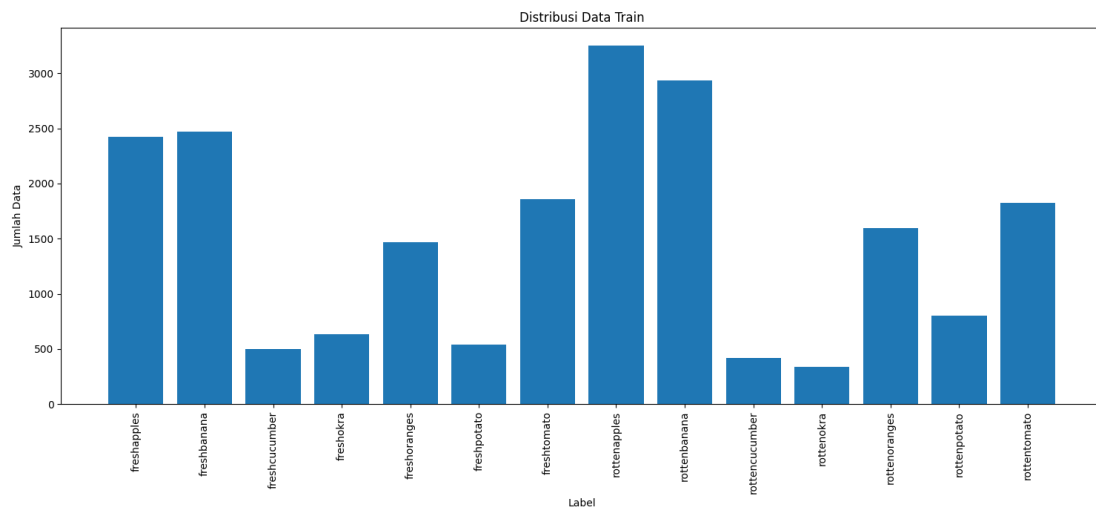
Pelaksanaan penelitian dimulai dengan pengumpulan data yaitu data gambar buah busuk dan buah segar dimana ada 7 kelas buah segar dan 7 kelas buah busuk. Selanjutnya data tersebut akan melalui pra pemrosesan untuk normalisasi data dan menyesuaikan ukuran gambar agar dapat digunakan untuk melatih model. Kemudian menyusun arsitektur *CNN* yang akan dilatih. Selanjutnya pelatihan model untuk dapat mengklasifikasikan kualitas buah dengan data gambar. Setelah model tersebut dilatih, akan dilakukan evaluasi pada model tersebut untuk mengetahui performa dari model yang telah dilatih. Selanjutnya model melalui *parameter tuning* yaitu memodifikasi *parameter* model dengan harapan model bisa menghasilkan performa yang lebih baik dari sebelumnya. Kemudian melatih kembali model yang telah melalui *parameter tuning*. Selanjutnya melakukan evaluasi model yang telah melalui proses *parameter tuning*.

### 2.2. Pengumpulan Data

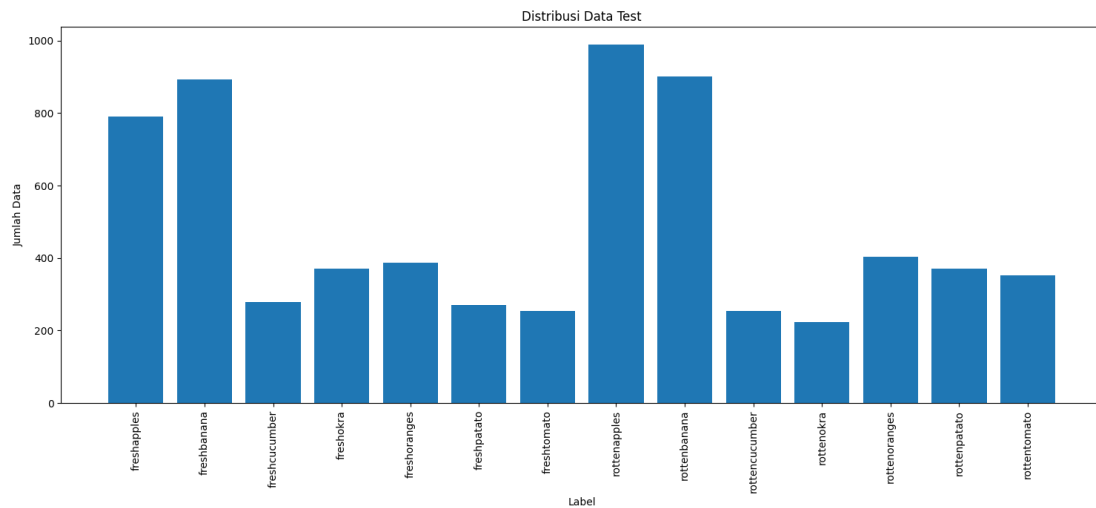
Data yang digunakan dalam penelitian ini adalah data sekunder dari *website kaggle*. Dataset ini bernama *Fresh and Rotten Classification*. Dataset berisi data buah dengan 7 jenis buah yang dibagi menjadi kelas buah busuk dan kelas buah segar, jadi total kelas yang ada pada dataset adalah 14 kelas. Dataset telah dibagi menjadi data *test* dan data *train*. Data dalam bentuk citra *RGB* berjumlah 27,782 yang dibagi menjadi data *test* dan *train*. Data *test* memiliki 6,738 data gambar, sedangkan data *train* memiliki 21,044 data gambar. Dataset dapat diakses melalui situs <https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification>.



**Gambar 2.** Contoh Data Gambar Dari Masing-Masing Kelas



**Gambar 3.** Distribusi data *train*



**Gambar 4.** Distribusi data *test*

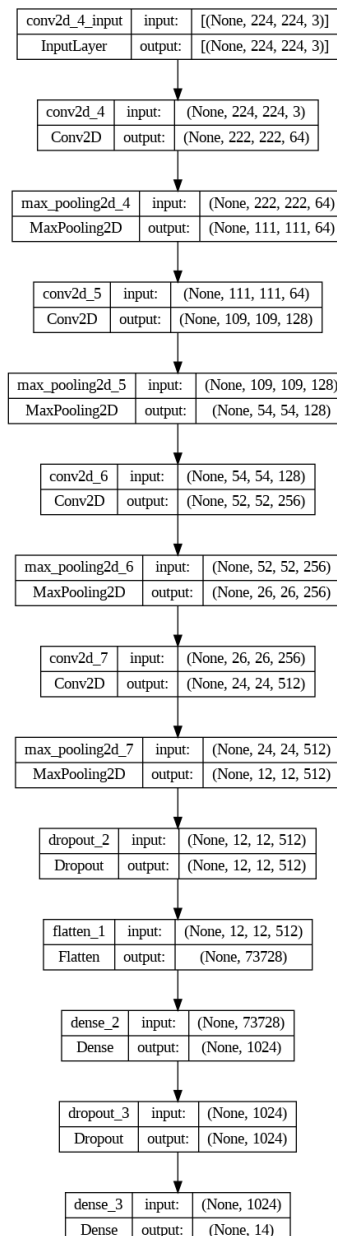
### 2.3. Pra Pemrosesan Data

Sebelum melakukan pelatihan model, setiap gambar pada data latih dan data uji harus melalui proses normalisasi dan *resize* terlebih dahulu. Normalisasi adalah proses perubahan nilai piksel dalam gambar, yang semula berkisar antara 0 hingga 255, menjadi rentang antara 0

hingga 1. Dalam penelitian ini, normalisasi dilakukan dengan membagi setiap nilai piksel di data latih dan data uji dengan 255. Setelah normalisasi, seluruh gambar yang akan masuk ke dalam model harus melewati standarisasi ukuran dengan melalui proses *rescale* menjadi gambar berukuran 224 x 224 dengan 3 *channel* warna yaitu *RGB*.

## 2.4. Penyusunan Model

Tahap selanjutnya adalah penyusunan model *CNN*. Setiap *convolutional layer* menggunakan fungsi aktivasi *ReLU* dan untuk *output layer* menggunakan fungsi aktivasi *softmax* untuk mengklasifikasikan lebih dari 2 kelas. Untuk susunan *layer* atau arsitektur model dapat dilihat pada gambar di bawah:



Gambar 5. Arsitektur Model *CNN* Pertama

## 2.5. Pelatihan Model

Model dilatih dengan tambahan *parameter* sebagai berikut:

1. Jumlah *epoch* adalah 10 dan 15
2. *Batch size* 32

## 2.6. Evaluasi Model

Evaluasi model menggunakan matriks akurasi, nilai *precision*, nilai *recall*, dan *f1-score*. Matriks akurasi adalah metrik evaluasi yang dapat dihasilkan dari data yang terdapat dalam *confusion matrix*. *Confusion matrix* adalah sebuah tabel yang dimanfaatkan untuk menilai kinerja model klasifikasi dalam konteks pengenalan pola. Tabel ini memberikan gambaran tentang sejauh mana model mampu mengklasifikasikan data secara benar dan menunjukkan seberapa sering model membuat kesalahan dalam mengklasifikasikan data. Berikut adalah rumus dari matriks akurasi, *precision*, *recall*, dan *f1-score*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

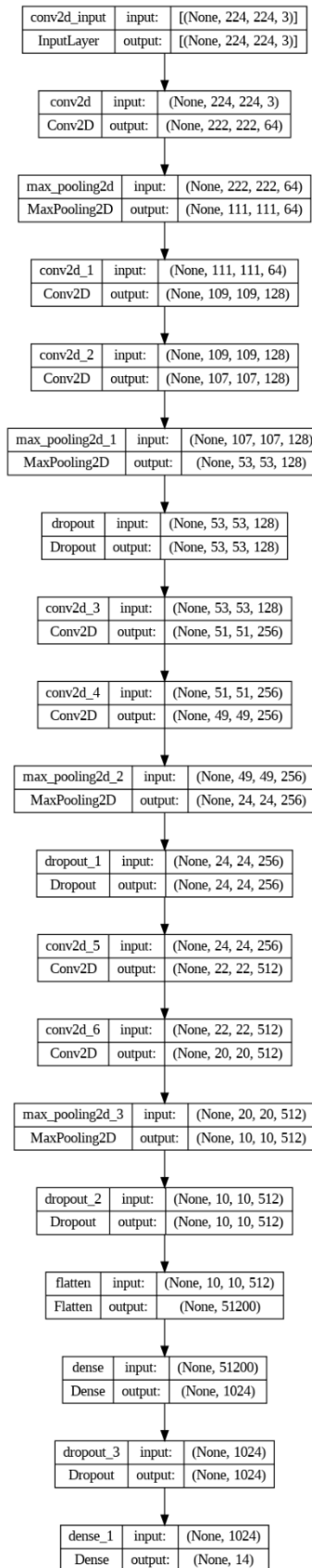
$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \cdot \frac{P \cdot R}{P + R} \quad (4)$$

## 2.7. Parameter Tuning

*Parameter tuning* adalah langkah melakukan perubahan atau memodifikasi beberapa komponen dalam model dengan maksud untuk mencapai kinerja model yang optimal. Modifikasi model pertama menghasilkan model kedua. Berikut adalah arsitektur model kedua:

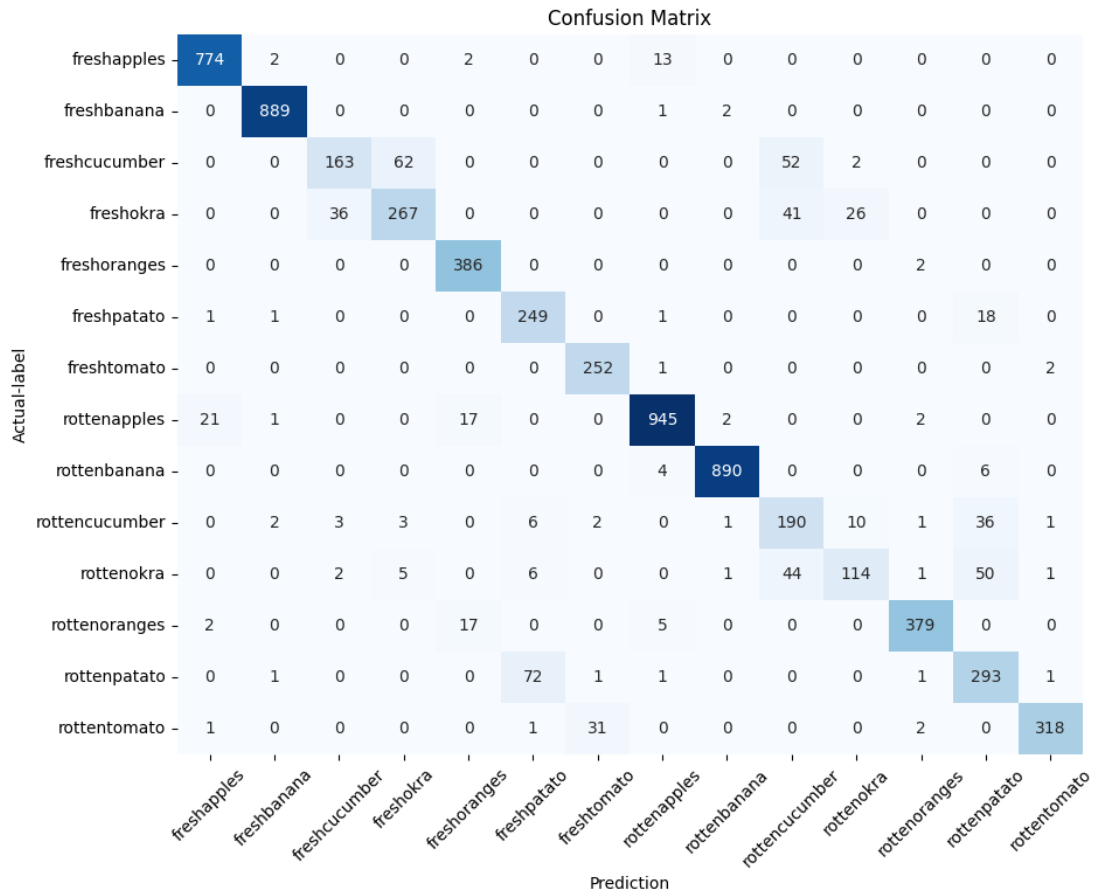


Gambar 6. Arsitektur Model CNN Kedua

### 3. Hasil dan Diskusi

#### 3.1. Hasil Pelatihan dan Evaluasi Model Pertama

Pelatihan model pertama dilakukan sebanyak 2 kali dengan *epoch* yang berbeda yaitu dengan *epoch* 10 dan *epoch* 15. Data *train* yang telah melalui proses normalisasi dan *resize* digunakan untuk pelatihan model. Hasil dari pelatihan model pertama dengan *epoch* 10 yang telah diuji menggunakan data *test* adalah model mendapat akurasi sebesar 91%. Untuk detail hasil evaluasi model pertama dengan 10 epoch menggunakan *precision*, *recall*, dan *f1-score* dapat dilihat pada gambar *confusion matrix* dan tabel berikut:



**Gambar 7.** Confusion Matrix Hasil Testing Model Pertama Epoch 10.

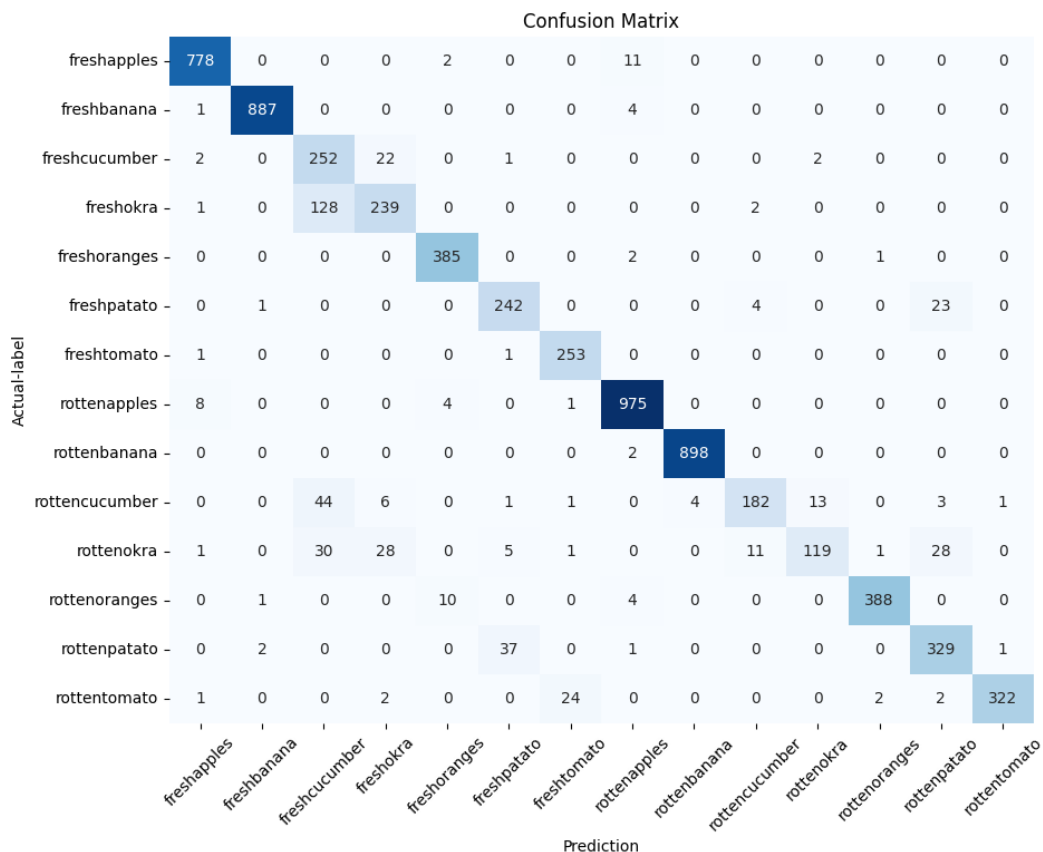
**Tabel 1.** Tabel Nilai Evaluasi Model Pertama Epoch 10

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>freshapples</i>	0,97	0,98	0,97	791
<i>freshbanana</i>	0,99	1,00	0,99	892
<i>freshcucumber</i>	0,80	0,58	0,67	279
<i>freshokra</i>	0,79	0,72	0,76	370
<i>freshoranges</i>	0,91	0,99	0,95	388
<i>freshpatato</i>	0,75	0,92	0,82	270
<i>freshtomato</i>	0,88	0,99	0,93	255
<i>rottenapples</i>	0,97	0,96	0,96	988

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>rottenbanana</i>	0,99	0,99	0,99	900
<i>rottencucumber</i>	0,58	0,75	0,65	255
<i>rottenokra</i>	0,75	0,51	0,61	224
<i>rottenoranges</i>	0,98	0,94	0,96	403
<i>rottenpatato</i>	0,73	0,79	0,76	370
<i>rottentomato</i>	0,98	0,90	0,94	353
<i>accuracy</i>		0,91		6738

Dari tabel diatas dapat dilihat pada hasil *testing* model pertama yang dilatih dengan 10 *epoch* menghasilkan akurasi sebesar 0,91 dengan total data *test* adalah 6738. Nilai *precision* tertinggi didapat pada kelas *freshbanana* dan *rottenbanana* dengan nilai *precision* 0,99. Untuk nilai *precision* terendah didapat pada kelas *rottencucumber* dengan nilai 0,58. Nilai *recall* tertinggi pada kelas *freshbanana* dengan nilai 1,00. Nilai *recall* terendah didapat pada kelas *rottenokra* dengan nilai 0,51. Nilai *f1-score* tertinggi didapat pada kelas *freshbanana* dan *rottenbanana* dengan nilai 0,99. Nilai *f1-score* terendah dimiliki kelas *rottenokra* dengan nilai 0,61.

Hasil dari pelatihan model pertama dengan *epoch* 15 yang telah diuji menggunakan data *test* adalah model mendapat akurasi sebesar 93%. Untuk detail hasil evaluasi model pertama dengan 15 *epoch* menggunakan *precision*, *recall*, dan *f1-score* dapat dilihat pada gambar *confusion matrix* dan tabel berikut:



Gambar 8. Confusion Matrix Hasil Testing Model Pertama Epoch 15.



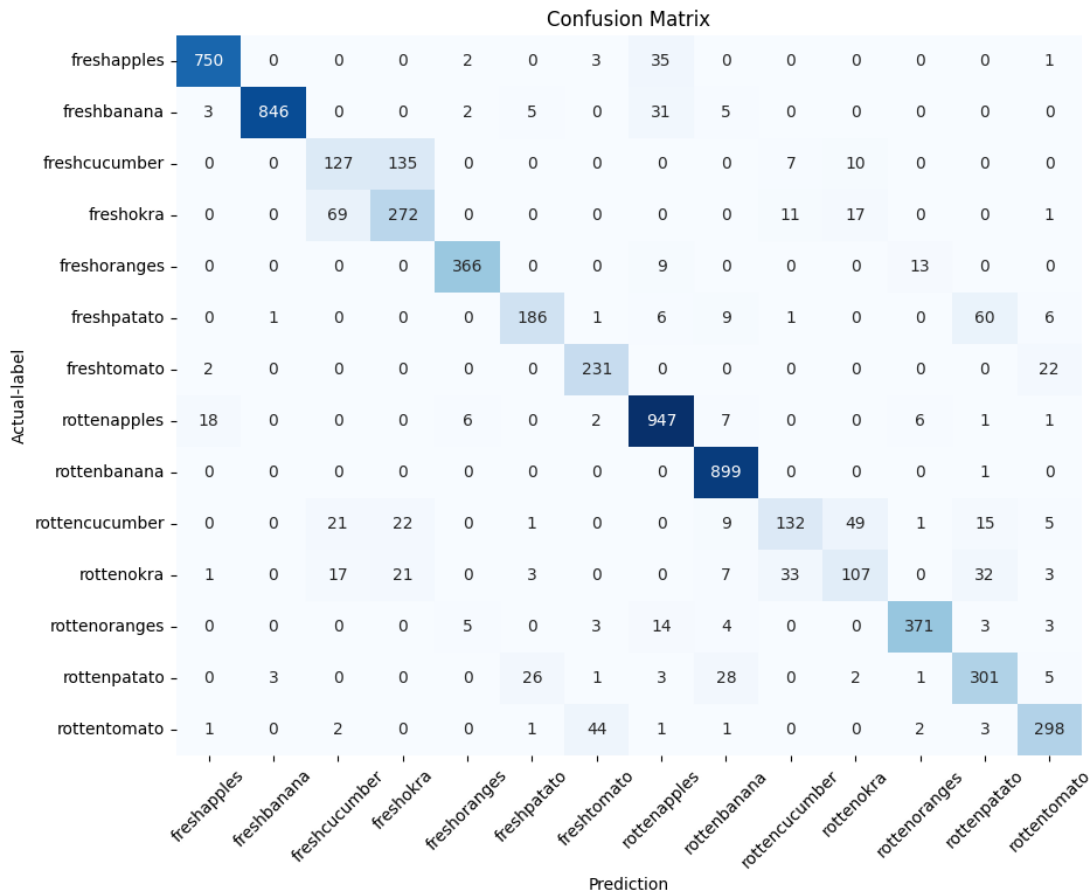
**Tabel 2.** Tabel Nilai Evaluasi Model Pertama *Epoch* 15

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>freshapples</i>	0,98	0,98	0,98	791
<i>freshbanana</i>	1,00	0,99	0,99	892
<i>freshcucumber</i>	0,56	0,90	0,69	279
<i>freshokra</i>	0,80	0,65	0,72	370
<i>freshoranges</i>	0,96	0,99	0,98	388
<i>freshpatato</i>	0,84	0,90	0,87	270
<i>freshtomato</i>	0,90	0,99	0,95	255
<i>rottenapples</i>	0,98	0,99	0,98	988
<i>rottenbanana</i>	1,00	1,00	1,00	900
<i>rottencucumber</i>	0,91	0,71	0,80	255
<i>rottenokra</i>	0,89	0,53	0,66	224
<i>rottenoranges</i>	0,99	0,96	0,98	403
<i>rottenpatato</i>	0,85	0,89	0,87	370
<i>rottentomato</i>	0,99	0,91	0,95	353
<i>accuracy</i>		0,93		6738

Dari tabel diatas dapat dilihat pada hasil *testing* model pertama yang dilatih dengan 15 *epoch* menghasilkan akurasi sebesar 0,93. Nilai *precision* tertinggi didapat pada kelas *freshbanana* dan *rottenbanana* dengan nilai *precision* 1,00. Untuk nilai *precision* terendah didapat pada kelas *freshcucumber* dengan nilai 0,56. Nilai *recall* tertinggi pada kelas *rottenbanana* dengan nilai 1,00. Nilai *recall* terendah didapat pada kelas *rottenokra* dengan nilai 0,53. Nilai *f1-score* tertinggi didapat pada kelas *rottenbanana* dengan nilai 1,00. Nilai *f1-score* terendah dimiliki kelas *rottenokra* dengan nilai 0,66.

### 3.2. Hasil Pelatihan dan Evaluasi Model Kedua

Model kedua adalah hasil modifikasi dari model pertama dengan penambahan *layer* konvolusi pada setiap blok konvolusi dan penambahan *layer dropout* setelah *maxpooling layer*. Pelatihan model kedua dilakukan sebanyak 2 kali dengan *epoch* yang berbeda yaitu dengan *epoch* 10 dan *epoch* 15. Data *train* yang telah melalui proses normalisasi dan *resize* digunakan untuk pelatihan model. Hasil dari pelatihan model kedua dengan *epoch* 10 yang telah diuji menggunakan data *test* adalah model mendapat akurasi sebesar 87%. Akurasi yang didapat lebih kecil dari akurasi model pertama dengan *epoch* 10. Untuk detail hasil evaluasi model kedua dengan *epoch* 10 menggunakan *precision*, *recall*, dan *f1-score* dapat dilihat pada gambar *confusion matrix* dan tabel berikut:



Gambar 9. Confusion Matrix Hasil Testing Model Kedua Epoch 10.

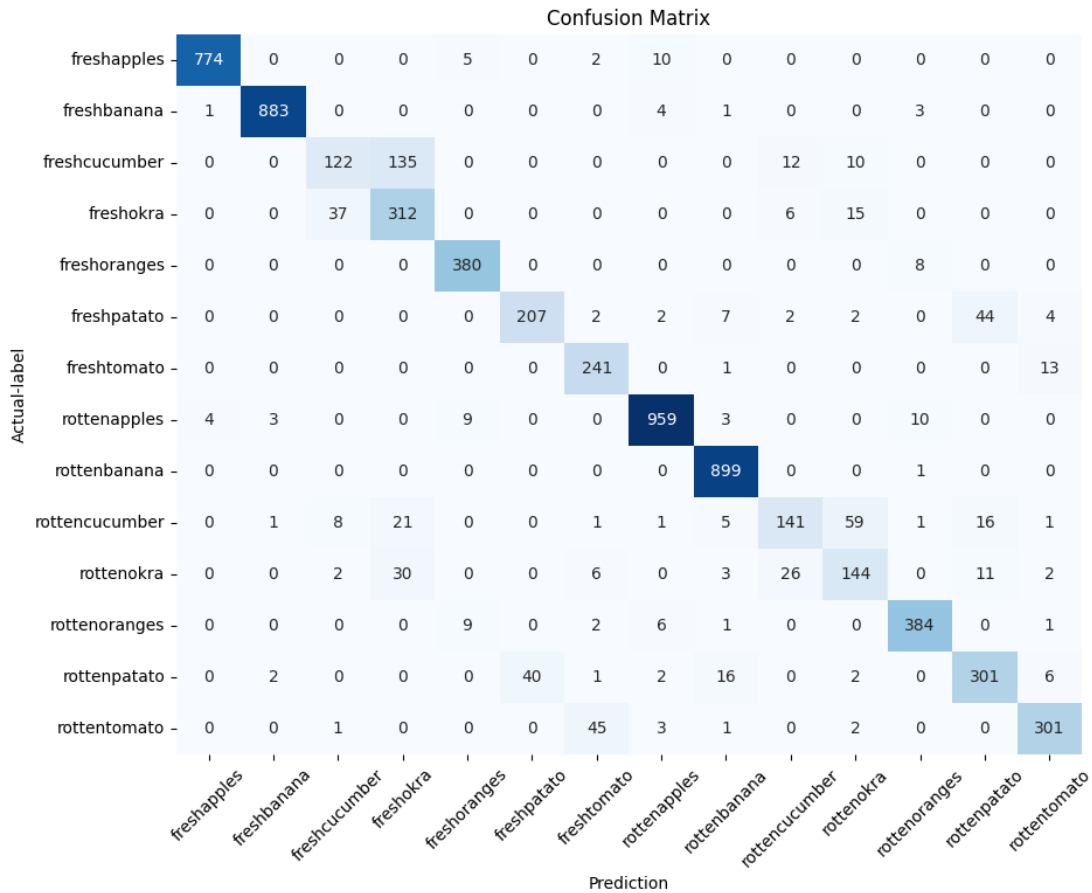
Tabel 3. Tabel Nilai Evaluasi Model Kedua Epoch 10

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>freshapples</i>	0,97	0,95	0,96	791
<i>freshbanana</i>	1,00	0,95	0,97	892
<i>freshcucumber</i>	0,54	0,46	0,49	279
<i>freshokra</i>	0,60	0,74	0,66	370
<i>freshoranges</i>	0,96	0,94	0,95	388
<i>freshpatato</i>	0,84	0,69	0,76	270
<i>freshtomato</i>	0,81	0,91	0,86	255
<i>rottenapples</i>	0,91	0,96	0,93	988
<i>rottenbanana</i>	0,93	1,00	0,96	900
<i>rottencucumber</i>	0,72	0,52	0,60	255
<i>rottenokra</i>	0,58	0,48	0,52	224
<i>rottenoranges</i>	0,94	0,92	0,93	403
<i>rottenpatato</i>	0,72	0,81	0,77	370
<i>rottentomato</i>	0,86	0,84	0,85	353
<i>accuracy</i>		0,87		6738

Dari tabel diatas dapat dilihat pada hasil *testing* model kedua yang dilatih dengan 10 *epoch* menghasilkan akurasi sebesar 0,87. Nilai *precision* tertinggi didapat pada kelas *freshbanana* dengan nilai *precision* 1,00. Untuk nilai *precision* terendah didapat pada kelas *freshcucumber* dengan nilai 0,54. Nilai *recall* tertinggi pada kelas *rottenbanana* dengan nilai 1,00. Nilai *recall* terendah didapat pada kelas *freshcucumber* dengan nilai 0,46. Nilai *f1-score* tertinggi didapat

pada kelas *freshbanana* dengan nilai 0,97. Nilai *f1-score* terendah dimiliki kelas *freshcucumber* dengan nilai 0,49.

Hasil dari pelatihan model kedua dengan *epoch* 15 yang telah diuji menggunakan data *test* adalah model mendapat akurasi sebesar 90%. Untuk detail hasil evaluasi model pertama dengan 15 *epoch* menggunakan *precision*, *recall*, dan *f1-score* dapat dilihat pada gambar *confusion matrix* dan tabel berikut:



**Gambar 10.** Confusion Matrix Hasil Testing Model Kedua Epoch 15.

**Tabel 4.** Tabel Nilai Evaluasi Model Kedua Epoch 15

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>freshapples</i>	0,99	0,98	0,99	791
<i>freshbanana</i>	0,99	0,99	0,99	892
<i>freshcucumber</i>	0,72	0,44	0,54	279
<i>freshokra</i>	0,63	0,84	0,72	370
<i>freshoranges</i>	0,94	0,98	0,96	388
<i>freshpatato</i>	0,84	0,77	0,80	270
<i>freshtomato</i>	0,80	0,95	0,87	255
<i>rottenapples</i>	0,97	0,97	0,97	988
<i>rottenbanana</i>	0,96	1,00	0,98	900
<i>rottencucumber</i>	0,75	0,55	0,64	255
<i>rottenokra</i>	0,62	0,64	0,63	224
<i>rottenoranges</i>	0,94	0,95	0,95	403
<i>rottenpatato</i>	0,81	0,81	0,81	370

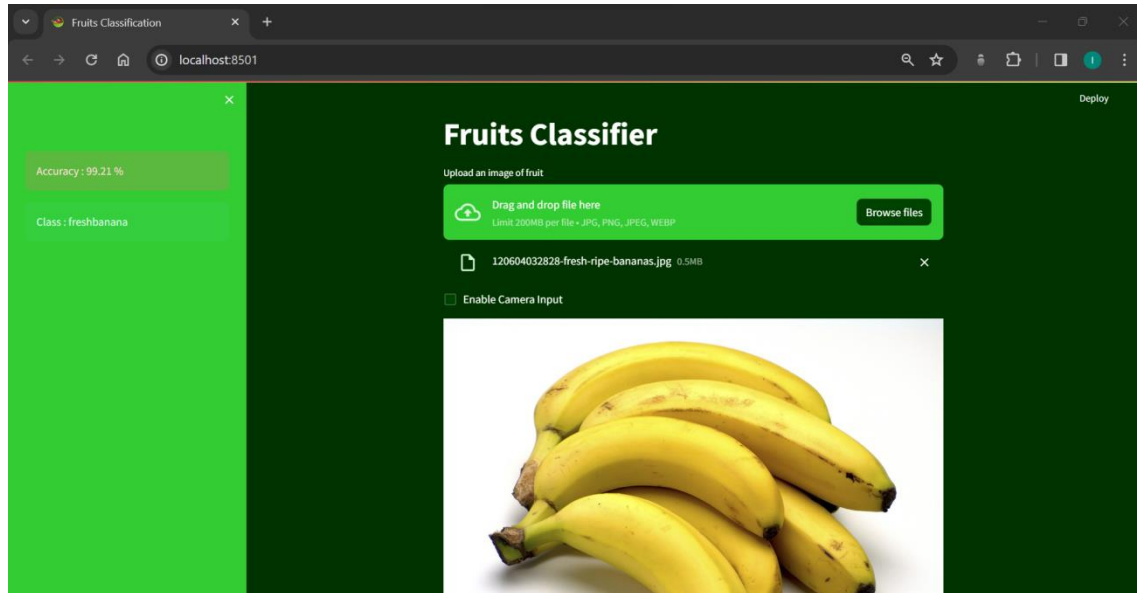
	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
<i>rotentomato</i>	0,92	0,85	0,88	353
<i>accuracy</i>			0,90	6738

Dari tabel diatas dapat dilihat pada hasil *testing* model kedua yang dilatih dengan *epoch* 15 menghasilkan akurasi sebesar 0,90. Akurasi yang didapat lebih kecil dari akurasi model pertama dengan *epoch* 15 bahkan lebih kecil dengan model pertama dengan *epoch* 10. Nilai *precision* tertinggi didapat pada kelas *freshbanana* dan *freshapples* dengan nilai *precision* 0,99. Nilai *precision* terendah didapat pada kelas *rottenokra* dengan nilai 0,62. Nilai *recall* tertinggi pada kelas *rottenbanana* dengan nilai 1,00. Nilai *recalls* terendah didapat pada kelas *freshcucumber* dengan nilai 0,44. Nilai *f1-score* tertinggi didapat pada kelas *freshbanana* dan *freshapples* dengan nilai 0,99. Nilai *f1-score* terendah dimiliki kelas *freshcucumber* dengan nilai 0,54.

Dari keempat percobaan diatas dapat diketahui bahwa banyaknya *epoch* pelatihan mempengaruhi peningkatan kinerja dari model. Berdasarkan data evaluasi yang telah didapat pada model pertama dan model kedua, pada *epoch* pelatihan 15 memiliki kinerja lebih baik dibandingkan dengan model yang dilatih pada *epoch* pelatihan 10. Berdasarkan data evaluasi diatas penambahan jumlah *layer* konvolusi belum tentu mempengaruhi peningkatan kinerja dari sebuah model. Model kedua yang memiliki lebih banyak *layer* konvolusi dan ditambah memiliki *layer dropout* menghasilkan hasil akurasi yang cenderung lebih kecil dibandingkan dengan model pertama yang memiliki *layer* lebih sedikit.

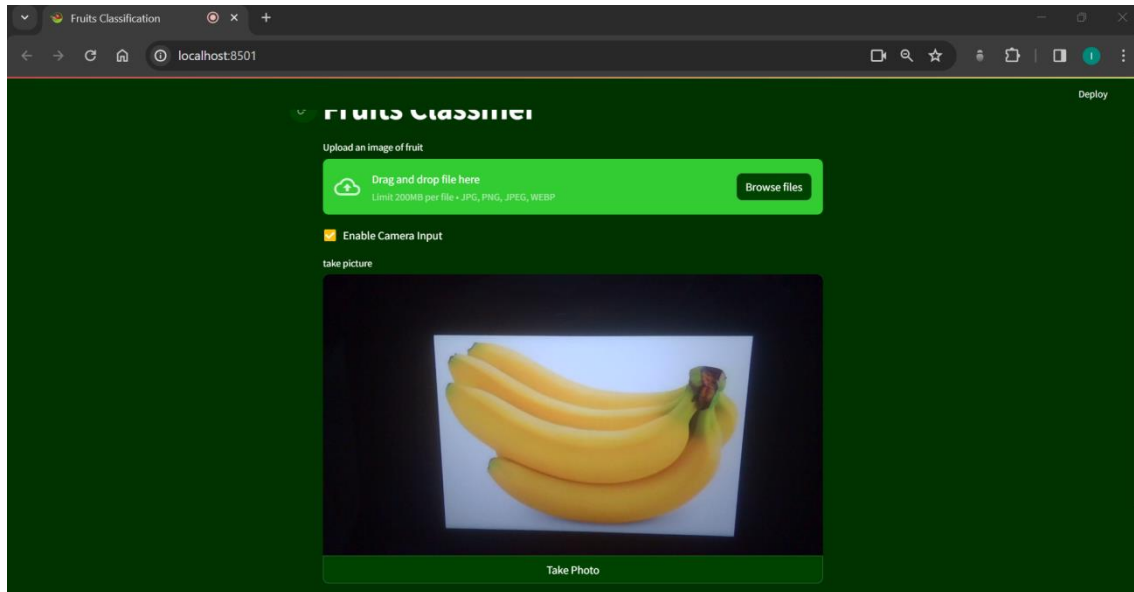
### 3.3. Integrasi Model pada Aplikasi Berbasis Web

Model yang akan diintegrasikan ke dalam aplikasi berbasis *web* adalah model pertama yang dilatih dengan *epoch* 15 yang mendapat akurasi sebesar 93%. Tampilan dari aplikasi berbasis *website* untuk integrasi model dapat dilihat pada gambar di bawah:



Gambar 11. Tampilan Website *Fruits Classifier*.

User dapat meng-*input* gambar dengan cara *drag and drop* atau dengan mencari data gambar di memori lokal. Kemudian hasil klasifikasi akan ditampilkan pada *sidebar* sebelah kiri dengan nama kelas dan juga akurasi yang didapatkan oleh model dalam mengklasifikasikan gambar tersebut. Selain *upload file* dari memori lokal, user juga bisa menggunakan fitur *camera input* dengan men-*checkboxlist* 'Enable Camera Input'. Web akan mengakses *camera input* pada *device* tersebut. Tampilan saat menggunakan fitur *camera input* dapat dilihat pada gambar di bawah:



**Gambar 12.** Tampilan Website *Fruits Classifier* Fitur *Camera Input*.

Dengan mengklik tombol '*Take Photo*' maka gambar akan langsung diproses kemudian diklasifikasikan.

#### 4. Kesimpulan

Berdasarkan hasil evaluasi pelatihan model pertama dan model kedua, dapat disimpulkan bahwa, peningkatan jumlah *epoch* cenderung meningkatkan kinerja model, terlihat dari hasil model pertama dan kedua pada *epoch* 10 dan *epoch* 15 dimana hasil evaluasi meningkat pada *epoch* 15. Model kedua dengan penambahan *layer* konvolusi dan *dropout* yang lebih banyak, tidak selalu menghasilkan performa yang lebih baik dibandingkan dengan model dengan *layer* yang lebih sedikit seperti model pertama. Penambahan *layer dropout* dapat membantu mencegah *overfitting*, namun efektifitas dari *layer dropout* tergantung pada kompleksitas model dan karakteristik *dataset*. Pada penelitian ini model yang memiliki lebih sedikit *layer dropout* cenderung memiliki akurasi lebih tinggi yaitu pada model pertama. Pada penelitian ini, didapat akurasi terbaik dari model pertama dengan *epoch* 15 dengan nilai akurasi sebesar 93%. Hasil ini membuktikan bahwa model yang telah dibangun dan dilatih telah mendekati akurasi yang baik dari penelitian sejenis sebelumnya.

#### Daftar Pustaka

- [1] A. Sharma, G. Phonsa. "Image Classification Using CNN". Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021, Available at SSRN: <https://ssrn.com/abstract=3833453> or <http://dx.doi.org/10.2139/ssrn.3833453>
- [2] N. Ibrahim, et al. "Klasifikasi Tingkat Kematangan Pucuk Daun Teh Menggunakan Metode Convolutional Neural Network". ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 10, no. 1, p. 162-176, 2022
- [3] M.A. Parab, N.D. Mehendale. "Red blood cell classification using image processing and CNN". bioRxiv, 2020, Available at: <https://doi.org/10.1101/2020.05.16.087239>
- [4] R.R. Allaam and A.T. Wibowo. "Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network (CNN)". e-Proceeding of Engineering, vol. 8, no. 2, p. 1153-1189, 2021
- [5] S.S. Nayak, "Fresh and Rotten Classification", 07 June 2023. [Online]. Available: <https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification>. [Accessed on 08 October 2023]

Halaman ini sengaja dibiarkan kosong