

## KLASIFIKASI KUALITAS BERAS MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK* BERBASIS *ANDROID*

Muhammad Yusuf<sup>1</sup>, Reinhard Ruimassa<sup>2\*</sup>, Anisa Iriani Tawainella<sup>3</sup>, dan Anisa Devi Maharani<sup>4</sup>

<sup>1,2,3,4</sup>Fakultas Teknik, Universitas Muhammadiyah Sorong, Sorong, Indonesia

Email<sup>1</sup>: [yusuf@um-sorong.ac.id](mailto:yusuf@um-sorong.ac.id)

Email<sup>2\*</sup>: [reinhardruimassa169@gmail.com](mailto:reinhardruimassa169@gmail.com)

Email<sup>3</sup>: [nisatwain12@gmail.com](mailto:nisatwain12@gmail.com)

Email<sup>4</sup>: [devimaharani6053@gmail.com](mailto:devimaharani6053@gmail.com)

### ABSTRAK

Pentingnya beras sebagai komoditas pangan di Asia, terutama di Indonesia, tidak hanya terletak pada perannya sebagai sumber karbohidrat utama, tetapi juga pada kualitasnya yang memengaruhi nilai jual, aspek gizi, dan kepuasan konsumen. Dalam konteks ini, penelitian mengenai kualitas beras menjadi esensial, dengan penekanan pada bentuk dan warna sebagai faktor utama yang memengaruhi mutu beras. Beras yang semakin putih, bersih, dan utuh cenderung memiliki kualitas yang lebih baik, membuat penilaian atas tingkat keputihan dan kebersihan beras menjadi krusial dalam industri pangan. Dengan menggunakan teknologi *deep learning* khususnya dalam tugas klasifikasi maka algoritma *convolutional neural network* dengan arsitektur VGG16 diharapkan dapat mengklasifikasikan kualitas beras bagus dan jelek. Hasil yang didapatkan dari penelitian ini berdasarkan pengujian confusion matriks mendapatkan accuracy sebesar 99% dan validation accuracy sebesar 97%. Hasil ini menunjukkan bahwa algoritma *convolutional neural network* dan VGG16 dapat dengan sangat baik mengklasifikasikan kualitas beras bagus atau jelek.

Kata kunci: *convolutional neural network*, beras, arsitektur VGG16

### ABSTRACT

The importance of rice as a food commodity in Asia, especially in Indonesia, lies not only in its role as the main source of carbohydrates, but also in its quality which affects the selling value, nutritional aspects, and consumer satisfaction. In this context, research on rice quality becomes essential, with an emphasis on shape and color as the main factors affecting rice quality. Rice that is whiter, cleaner, and intact tends to have better quality, making the assessment of the level of whiteness and cleanliness of rice crucial in the food industry. By using deep learning technology, especially in classification tasks, the convolutional neural network algorithm with the VGG16 architecture is expected to be able to classify good and bad rice quality. The results obtained from this study based on confusion matrix testing obtained an accuracy of 99% and a validation accuracy of 97%. These results indicate that the convolutional neural network and VGG16 algorithms can very well classify good or bad rice quality.

Keywords: convolutional neural network, rice, VGG16 architecture

## 1. PENDAHULUAN

Beras memiliki peran signifikan sebagai komoditas pangan, terutama di berbagai negara di Asia, termasuk Indonesia. Kualitas beras menjadi elemen kunci yang memengaruhi nilai jual, aspek gizi, dan kepuasan konsumen [1]. Kualitas beras dapat diklasifikasikan menjadi tiga kategori: berdasarkan pasar beras, berdasarkan Standar Nasional Indonesia (SNI), dan berdasarkan preferensi konsumen [2]. Akibatnya, beras yang tersedia di pasaran harus memenuhi standar dan kualitas yang baik. Beberapa karakteristik fisik beras memengaruhi mutu beras, seperti (1) bentuk dan ukuran butiran, (2) tingkat derajat sosoh, (3) kejernihan, dan (4) kebersihan dan kemurnian. Karena beras dikonsumsi dalam bentuk butiran, sifat fisik beras sangat penting untuk kualitasnya [3]. Untuk itu masyarakat perlu meningkatkan kewaspadaan dalam mengecek kualitas beras yang cocok untuk diolah. Namun, banyak dari mereka yang belum memiliki pengetahuan khusus seperti membedakan butir padi yang utuh dan warna yang baik pada beras dalam mengidentifikasi kualitas beras yang sesuai untuk keperluan memasak. Dalam mengatasi tantangan ini, pemanfaatan teknologi informasi, khususnya melalui aplikasi Android, dapat menjadi solusi yang efektif.

\*) Penulis Korespondensi

Dikirim : 03 Agustus 2024

Diterima : 11 Agustus 2024

Publikasi Online : 31 Oktober 2024

Model *Deep Learning* yang digunakan mengadopsi algoritma *Convolutional Neural Network* (CNN) dengan arsitektur VGG16 dan metode pengembangan aplikasi yaitu *Extreme Programming*.

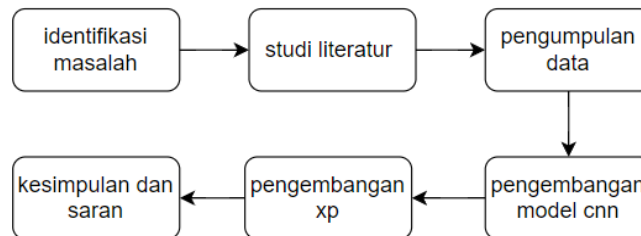
Berdasarkan penelitian sebelumnya pemanfaatan CNN mendapatkan akurasi sebesar mencapai 98% pada 4 jenis beras dengan 2500 data gambar serta menggunakan arsitektur AlexNet [4]. Penelitian ini menunjukkan bahwa CNN dapat digunakan dalam pengklasifikasian citra gambar pada beras dan juga penelitian pada tahun 2023 yang berfokus pada klasifikasi varietas beras menggunakan CNN dan arsitektur VGG16 dan VGG19 mendapatkan hasil sebesar 97% pada tes akurasi VGG19 dan 98% tes akurasi pada VGG16 [5]. Penelitian ini juga membuktikan bahwa algoritma CNN sangat baik dalam menjalankan tugas klasifikasi terutama pada citra gambar beras.

Namun meskipun penelitian terdahulu dapat mencari keunggulan dari segi algoritma dan arsitektur yang digunakan, hal ini belum memberikan dampak yang nyata pada kehidupan masyarakat sehari-hari. Oleh sebab itu pada penelitian ini, penulis bermaksud untuk mengembangkan model yang telah dibuat menggunakan CNN dan arsitektur VGG16 sampai pada tahap aplikasi Android. Hal ini bertujuan untuk memudahkan masyarakat dalam mengenali jenis beras yang baik dan buruk sesuai dengan Standar Nasional Indonesia (SNI). Penelitian ini diharapkan dapat memberikan pemahaman lebih baik untuk pembuatan model CNN dengan arsitektur VGG16.

## 2. MATERI DAN METODE

### Skema Penelitian

Pada tahapan ini akan dijelaskan skema penelitian dari penulis mulai dari mendeskripsikan masalah sampai dengan tahap kesimpulan dan saran (Gambar 1).



Gambar 1. Skema Penelitian

#### 1. Identifikasi Masalah

Setelah menentukan topik penelitian, langkah pertama yang harus dilakukan oleh peneliti adalah mengidentifikasi masalah yang akan diteliti. Tujuan dari identifikasi ini adalah untuk mengklarifikasi batas-batas permasalahan sehingga penelitian dapat tetap fokus pada tujuannya.

#### 2. Studi Literatur

Langkah kedua yang dilakukan dalam penelitian ini melibatkan pencarian dan analisis sumber-sumber penulisan dari berbagai artikel dan jurnal yang berkaitan dengan subjek penelitian yang berkaitan dengan penelitian ini. Fokus utama adalah mengklasifikasi kualitas beras, pendekatan CNN, serta perangkat lunak atau alat bantu untuk mengimplementasikan sistem pakar dalam klasifikasi kualitas beras [6][7].

#### 3. Pengumpulan Data

Langkah ketiga, peneliti mengumpulkan data-data gambar beras dengan dua kualitas yaitu kualitas bagus dan jelek.

#### 4. Pengembangan Model CNN

Perancangan sistem dalam penelitian ini mencakup beberapa tahapan yang perlu ditimbangan dalam penelitian ini yaitu:

##### a. Dataset

Data-data yang telah dikumpulkan dalam bentuk file gambar seperti jpg, jpeg, dan png akan disatukan ke dalam folder dataset pada Google Drive sebagai media penyimpanan [8].

##### b. Split Data

Pada langkah ini peneliti melakukan pembagian data-data yang telah dikumpulkan ke dalam dua label berdasarkan kualitas beras bagus dan jelek.

##### c. Pelabelan Data

Pada langkah ini data mentah yang telah displit pada tahap sebelumnya dilabelkan berdasarkan kelas yang diinginkan [9].

##### d. Preprocessing Data

Pada langkah ini peneliti melakukan pengolahan gambar yang telah diperoleh dengan melakukan *resize*, penyesuaian rotasi, dan perubahan resolusi kualitas gambar agar sesuai dengan format yang dibutuhkan oleh CNN untuk mempersiapkan data sebelum dimasukkan ke dalam model [10] [11].

Pada penelitian ini, penulis melakukan *resize* menjadi 224 x 224 piksel pada dataset, serta meningkatkan resolusi gambar menjadi lebih bagus.

e. Pembuatan Model

Pada langkah ini, dataset yang telah melalui proses preprocessing dapat digunakan untuk mengklasifikasikan kualitas beras. Penggunaan metode CNN dengan arsitektur VGG16 dimulai dengan tahap awalnya *convolutional layer* yang berfungsi sebagai lapisan awal untuk menerima input citra dan melakukan pengolahan citra. Setelah itu, dataset secara keseluruhan diubah menjadi ukuran 224 x 224 piksel, karena input pada proses CNN dengan arsitektur VGG16 adalah gambar berukuran 224 x 224 x 3. Selanjutnya, lapisan yang penuh terhubung adalah lapisan yang digunakan untuk memproses klasifikasi hasil dari tahap *pooling*. Tahap selanjutnya, data akan dibagi menjadi data uji dan data latih. Setelah model dibangun, data uji akan digunakan untuk menguji kinerja sistem [12] [13].

f. Pelatihan Model CNN

Setelah menetapkan model, maka pelatihan data dilakukan dengan beberapa proses model CNN dan juga arsitektur VGG16 agar diketahui seberapa mengenalnya sistem terhadap citra gambar kualitas beras [14].

g. Save Model

Pada titik ini, peneliti akan menyimpan model untuk digunakan pada aplikasi kemudian. Ini akan disimpan dengan ekstensi file *tflite*. Selanjutnya, data pelatihan akan diubah menjadi *tensorflow lite* untuk memudahkan pemodelan aplikasi di Android.

5. Pengembangan Aplikasi

Perancangan sistem dalam penelitian ini mencakup beberapa tahapan yang perlu ditimbangan dalam penelitian [15] yaitu:

a. *Planning* (Perencanaan)

Penelitian ini menganalisis kebutuhan fungsional dan non-fungsional sistem yang dikembangkan.

b. *Design* (Perancangan)

Pada tahap kedua, penelitian ini melakukan perancangan dengan dibuat *flowchart*, *use case*, *activity diagram* serta *user interface*, untuk menjelaskan gambaran alur penelitian yang dilakukan.

c. *Coding* (Pengkodean)

Pada tahap ketiga, penelitian ini melaksanakan perancangan aplikasi sesuai dengan desain sistem yang telah disusun sebelumnya. Aplikasi dibuat dengan menggunakan metode CNN, untuk tampilan antarmuka dibuat pada Android Studio dan untuk *back-end* digunakan java pada Android Studio.

d. *Testing* (Pengujian)

Pada tahap keempat, penelitian ini melakukan pengujian. Sistem harus melalui tahap pengujian untuk menegaskan bahwa *software* berfungsi dengan baik dan menemukan kesalahan.

### 3. HASIL DAN PEMBAHASAN

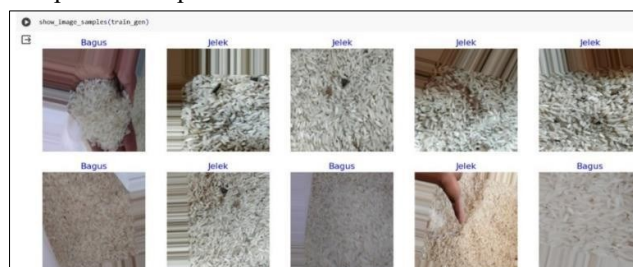
#### Pengumpulan Data

Peneliti mengumpulkan *dataset* berupa gambar yang diambil dari Gudang beras bulog di Kampung Baru Kota Sorong. *Dataset* yang diambil sebanyak 340 data gambar yang terbagi menjadi 170 data gambar untuk kualitas bagus, dan 170 data gambar untuk kelas jelek.

#### Pengembangan Model CNN

##### 1. Dataset

Data gambar yang digunakan dalam penelitian ini terdiri dari 340 data gambar yang dibagi menjadi 2 kelas, yaitu kelas kualitas bagus dan kualitas buruk. Setelah itu dataset diupload ke Google Drive yang dimana berfungsi sebagai lokasi dalam menyimpan data gambar dan data-data tersebut akan di latih pada Google Collab. Berikut contoh sampel *dataset* pada Gambar 2.



Gambar 2. Contoh Dataset

## 2. Split Data

Pada tahap ini data dibagi menjadi data *train*, data *test*, dan data validasi. Berdasarkan jumlah *dataset* yang disiapkan maka *dataset* di *split* dengan rasio 70:20:10 dengan persentase 237 data *train*, 68 data *test*, dan 35 data validasi.

## 3. Pelabelan Data

Setelah melewati tahapan *split data*, *dataset* yang telah displit kemudian dilabelkan sesuai dengan kelas yang diinginkan. Pada kasus ini pelabelan terbagi menjadi dua kelas yaitu kualitas bagus dan kualitas jelek.

## 4. Preprocessing Data

Tahapan ini adalah bagian dari proses mengatur pembagian data yang akan disebar ke model CNN ke dalam kelas yang ditentukan untuk meningkatkan data gambar dalam proses pelatihan model dengan menggunakan *hard library* dan *tensorflow*.

## 5. Pembuatan Model

Pada tahap ini peneliti menggunakan arsitektur VGG16. Hal ini dikarenakan VGG16 merupakan arsitektur terbaru yang mampu mengklasifikasikan gambar dengan tingkat akurasi yang lebih tinggi dari pada arsitektur sebelumnya [5]. Gambar 3 merupakan *code* yang digunakan pada pembuatan model.

```
▼ pembuatan model

model_name='Cendekia'
print("Building model with", base_model)
model = tf.keras.Sequential([
    # Note the input shape is the desired size of the image 128x128 with 3 bytes color
    # This is the first convolution
    base_model,
    tf.keras.layers.Conv2D(filters=32, padding='same', kernel_size=3, activation='relu', strides=1),
    tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
    tf.keras.layers.Dropout(rate=0.5),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(2, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001), loss='categorical_crossentropy', metrics='accuracy')
```

Gambar 3. Pembuatan Model

## 6. Training Model CNN

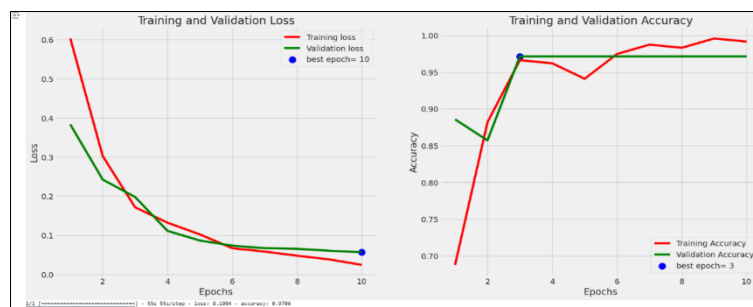
Pada tahap ini peneliti melakukan pelatihan pada data-data gambar yang telah melewati tahapan sebelumnya. Berikut ini adalah hasil *training* yang dapat dilihat pada Gambar 4 dan Gambar 5.

```
▼ train model

epochs = 10
history=model.fit(x=train_gen, epochs=epochs, validation_data=valid_gen)

Epoch 1/10
4/4 [=====] - 244s 61s/step - loss: 0.6030 - accuracy: 0.6878 - val_loss: 0.3833 - val_accuracy: 0.8857
Epoch 2/10
4/4 [=====] - 215s 49s/step - loss: 0.3027 - accuracy: 0.8819 - val_loss: 0.2424 - val_accuracy: 0.8571
Epoch 3/10
4/4 [=====] - 233s 61s/step - loss: 0.1715 - accuracy: 0.9662 - val_loss: 0.1978 - val_accuracy: 0.9714
Epoch 4/10
4/4 [=====] - 228s 59s/step - loss: 0.1322 - accuracy: 0.9620 - val_loss: 0.1113 - val_accuracy: 0.9714
Epoch 5/10
4/4 [=====] - 230s 59s/step - loss: 0.1022 - accuracy: 0.9489 - val_loss: 0.0866 - val_accuracy: 0.9714
Epoch 6/10
4/4 [=====] - 223s 57s/step - loss: 0.0670 - accuracy: 0.9747 - val_loss: 0.0735 - val_accuracy: 0.9714
Epoch 7/10
4/4 [=====] - 216s 49s/step - loss: 0.0583 - accuracy: 0.9873 - val_loss: 0.0669 - val_accuracy: 0.9714
Epoch 8/10
4/4 [=====] - 216s 49s/step - loss: 0.0478 - accuracy: 0.9831 - val_loss: 0.0655 - val_accuracy: 0.9714
Epoch 9/10
4/4 [=====] - 232s 59s/step - loss: 0.0383 - accuracy: 0.9958 - val_loss: 0.0604 - val_accuracy: 0.9714
Epoch 10/10
4/4 [=====] - 222s 51s/step - loss: 0.0242 - accuracy: 0.9916 - val_loss: 0.0569 - val_accuracy: 0.9714
```

Gambar 4. Hasil Training



Gambar 5. Grafik Hasil Training

Gambar 4 terlihat jumlah *epoch* yang digunakan yaitu 10 *epoch* untuk menentukan berapa kali dataset dilatih. Nilai *accuracy* tertinggi didapat mencapai 99% dan *validation accuracy* mencapai 97%. Pada *training loss* didapatkan angka terbaik yaitu 0.0242 dan *validation loss* didapatkan angka terbaik yaitu 0.0569. Pada Gambar 5 adalah gambar visualisasi hasil yang didapat pada Gambar 4. Dapat dilihat

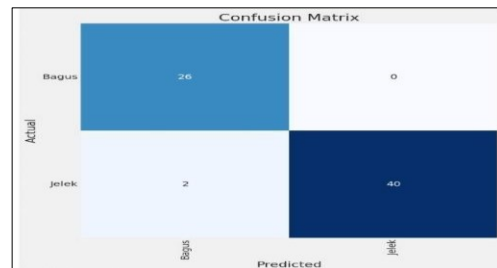
grafiknya menunjukkan hasil *train* dan *validation loss* yang menurun dan *training* dan *validation accuracy* yang memiliki grafik tidak stabil tetapi pada akhirnya menunjukkan kenaikan yang signifikan sampai pada *epoch* terakhir.

### 7. Pengujian Model

Pada tahap ini, dilakukan pengujian akurasi dari arsitektur VGG16 dan pengecekan label prediksi pada sistem yang telah dikembangkan dengan menggunakan *Confusion Matrix* dan *Classification Report* (Gambar 6 dan Gambar 7).

Classification Report:				
	precision	recall	f1-score	support
Bagus	0.93	1.00	0.96	26
Jelek	1.00	0.95	0.98	42
accuracy			0.97	68
macro avg	0.96	0.98	0.97	68
weighted avg	0.97	0.97	0.97	68

Gambar 6. Classification Report



Gambar 7. Confusion Matriks

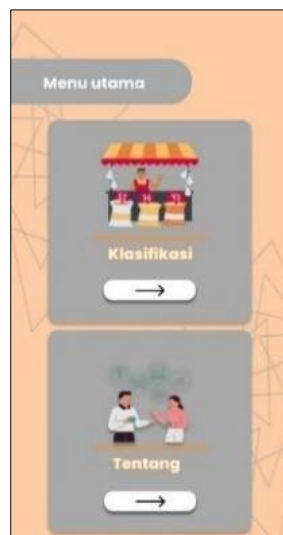
Berdasarkan Gambar 7, model yang dibuat menggunakan CNN dan VGG16 menunjukkan hasil yang bagus pada pengujian *Confusion Matriks*. Pada uji kelas Bagus mendapatkan 26 prediksi yang berhasil dan 2 yang gagal. Pada uji kelas Jelek mendapatkan 40 prediksi yang berhasil dan 0 yang gagal. Pada Gambar 6, prediksi arsitektur model dengan menggunakan *Classification Report*. Berdasarkan pengujian kelas Bagus, *precision* dapat mencapai angka 93%, *recall* 100%, *f1-score* 96%. Pada pengujian kelas Jelek, *precision* dapat mencapai 100%, *recall* 95%, *f1-score* 98%. Oleh karena itu, dengan menggunakan *VGG16* tingkat akurasi yang dicapai mencapai 97%. Hal ini menunjukkan bahwa model memiliki tingkat akurasi yang sangat bagus dikarenakan akurasi model mencapai akurasi diatas 90% pada *precision*, *recall*, dan *f1-score*.

### 8. Saving Model

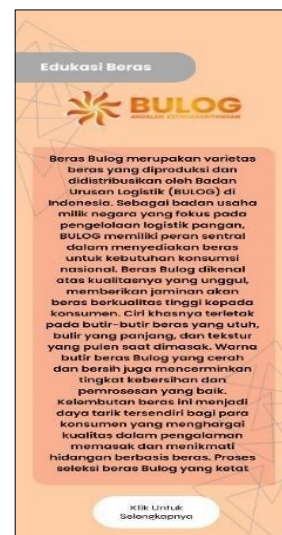
Ini adalah langkah yang diambil untuk menyimpan model yang telah dibuat. Mentransfer model dari Google Collab menjadi file tflite, yang dapat diterapkan pada aplikasi Android.

### Pengembangan Tampilan Antarmuka Aplikasi

Pada tahapan ini penulis membuat tampilan antarmuka aplikasi yang akan digunakan oleh pengguna berbasis Android.



Gambar 8. Menu Utama



Gambar 9. Menu Tentang

#### 1. Menu Utama

Menu utama adalah menu yang paling awal diakses oleh pengguna setelah memasuki aplikasi. Gambar 8 merupakan tampilan menu utama dari aplikasi yang dibangun. Dengan terdiri dari 2 fitur yaitu fitur klasifikasi untuk mengklasifikasikan kualitas beras, dan tentang yang bertujuan untuk memberikan informasi mengenai edukasi beras.

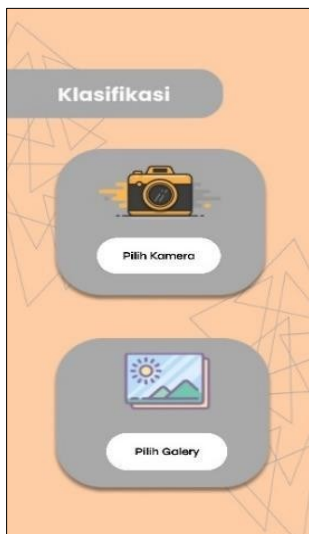


## 2. Menu Tentang

Menu tentang dapat diakses melalui menu utama setelah pengguna mengakses aplikasi. Gambar 9 merupakan Menu tentang yang mengandung informasi tentang kualitas beras. Pada gambar disamping merupakan tampilan menu tentang. Menu ini berisi informasi tentang kualitas beras dan tombol klik untuk selengkapnya yang dapat beralih ke website perusahaan bulog tentang kualitas beras.

## 3. Menu Klasifikasi

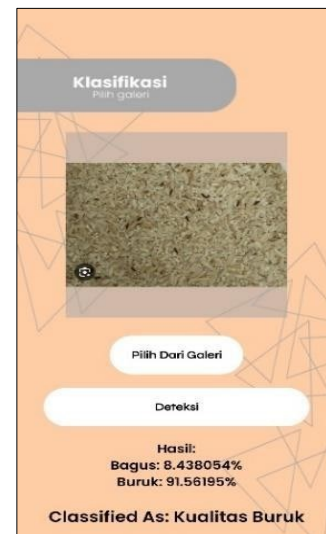
Menu tampilan klasifikasi dapat diakses melalui menu utama setelah pengguna mengakses aplikasi. Pada Gambar 10 Menu ini memiliki dua menu pilih kamera dan pilih galery untuk mendeteksi kualitas beras. Pada gambar disamping merupakan halaman klasifikasi pada aplikasi yang menyediakan dua opsi menu yakni pilih kamera dan pilih gambar. Opsi pilih kamera dapat dimanfaatkan untuk mengambil gambar objek yang akan dideteksi, sementara opsi pilih gambar memungkinkan pengguna untuk memilih gambar objek dari galeri yang akan dijalani proses deteksi.



Gambar 10. Menu Klasifikasi



Gambar 11. Menu Pilih Kamera



Gambar 12. Menu Pilih Galeri

## 4. Menu Pilih kamera

Menu pilih kamera dapat diakses setelah mengakses tampilan klasifikasi. Gambar 11, user akan memasukkan gambar yang akan dideteksi maka sistem akan menampilkan hasil klasifikasi dari gambar yang telah diinput.

## 5. Menu Pilih Galeri

Menu pilih galeri dapat diakses setelah mengakses tampilan klasifikasi. Pada Gambar 12, user akan memasukkan gambar lewat pilih galeri yang akan dideteksi maka sistem akan menampilkan hasil klasifikasi dari gambar yang telah diinput.

## 4. KESIMPULAN DAN SARAN

Berdasarkan hasil evaluasi menggunakan pengujian confusion matriks pada penelitian dapat disimpulkan bahwa penggunaan algoritma CNN sangat cocok untuk digunakan pada tugas *image recognition* terutama pada penelitian ini dengan objek citra kualitas beras dengan label kualitas bagus dan jelek. Penelitian menggunakan arsitektur VGG16 dapat meningkatkan akurasi model diatas 90 % tepatnya pada angka 97% dengan nilai *f1-score* pada label kualitas bagus dan jelek berturut-turut adalah 96% dan 98%. Oleh sebab itu, algoritma CNN dan VGG16 dapat dengan sangat baik mengklasifikasikan beras dengan kualitas bagus dan jelek. Adapun saran yang dapat diberikan adalah dari segi kompleksitas model dengan menambahkan ukuran dataset dan jumlah label pada setiap jenis beras sehingga mendapatkan model yang kompleks untuk masalah pendeteksian kualitas beras.

## DAFTAR PUSTAKA

- [1] M. R. Wathani, and N. Hidayati, "Analisis Perbandingan Fungsi Aktivasi CNN Pada Pengelompokan Jenis Beras Berdasarkan Mutu Beras," *BRAHMANA J. Penerapan Kecerdasan Buatan*, vol. 4, no. 2, pp. 144–153, 2023. [Online]. Available: <https://tunasbangsa.ac.id/pkm/index.php/brahmana/>

- [article/view/189](#).
- [2] M. Z. Altim, A. Basalamah, K. Kasman, R. A. Syamsul, and A. Yudhistira, "Implementasi *Convolutional Neural Network (CNN)* Untuk Penentuan Kualitas Beras Berdasarkan Bentuk Dan Warna," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 8, no. 2, pp. 380–386, 2023, doi: [10.24252/instek.v8i2.42968](https://doi.org/10.24252/instek.v8i2.42968).
  - [3] F. Paramudita, and M. I. Zulfa, "Aplikasi *Android* Pendeteksi Kualitas Beras Berbasis Machine Learning Menggunakan Metode *Convolutional neural network*," *J. Pendidik. dan Teknol. Indones.*, vol. 3, no. 7, pp. 297–305, 2023, doi: [10.52436/1.jpti.310](https://doi.org/10.52436/1.jpti.310).
  - [4] V. A. A. Pellokila, K. Usman, and N. K. C. Pratiwi, "Pemanfaatan *Convolutional neural network (CNN)* Untuk Klasifikasi Jenis Beras Berbasis Citra," *e-Proceedings of Engineering*, vol. 10, no. 5, pp. 4246–4252, 2023.[Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/21212/20489>
  - [5] W. I. Kusumawati, and A. Z. Noorizki, "Perbandingan Performa Algoritma *VGG16* Dan *VGG19* Melalui Metode *CNN* Untuk Klasifikasi Varietas Beras," *J. Comput. Electron. Telecommun.*, vol. 4, no. 2, 2023, doi: [10.52435/complete.v4i2.387](https://doi.org/10.52435/complete.v4i2.387).
  - [6] A. M. Lesmana, R. P. Fadhillah, and C. Rozikin, "Identifikasi Penyakit pada Citra Daun Kentang Menggunakan *Convolutional neural network (CNN)*," *J. Sains dan Inform.*, vol. 8, no. 1, pp. 21–30, 2022, doi: [10.34128/jsi.v8i1.377](https://doi.org/10.34128/jsi.v8i1.377).
  - [7] B. Julianto, I. N. Farida, and M. A. D. W. Dara, "Implementasi Metode *CNN* Pada Aplikasi *Android* Untuk Deteksi Penyakit Pada Daun Padi Penulis Korespondensi," *Inotek*, vol. 7, pp. 2549–7952, 2023, [Online]. Available: <https://proceeding.unpkediri.ac.id/index.php/inotek/>
  - [8] L. Arisandi, and B. Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma *Convolutional neural network*," *J. Sist. Cerdas*, vol. 5, no. 3, pp. 135–146, 2022, doi: [10.37396/jsc.v5i3.262](https://doi.org/10.37396/jsc.v5i3.262).
  - [9] R. P. Pratama, R. Hidayat, N. F. Fano, A. Akbar, and N. A. Rakhmawati, "Implementasi Deep Learning untuk Entity Matching pada Dataset Obat (Studi Kasus K24 dan Farmaku)," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 6, no. 3, pp. 130–138, 2021, doi: [10.14421/jiska.2021.6.3.130-138](https://doi.org/10.14421/jiska.2021.6.3.130-138).
  - [10] A. Rinardi, and V. A. Herlinda, "Implementasi Model *Convolutional neural network* Untuk Klasifikasi Daun Sirih Dan Daun Pandan," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, pp. 2–6, 2022.
  - [11] E. L. P. Ristanti, "Analisis Dan Perbandingan Arsitektur *VGG16* dan *MobileNetV2* Untuk Klasifikasi Dan Identifikasi Penyakit Daun Pada Tanaman Cabai Menggunakan *CNN*," *J. Ilm. Sain dan Teknol.*, vol. 2, no. 9, pp. 216–226, 2024. [Online]. Available: <http://jurnal.kolibi.org/index.php/scientica/article/view/2381>.
  - [12] M. D. Meitanyta, C. A. Sari, E. H. Rachmawanto, and R. R. Ali, "VGG16 Architecture On CNN For American Sign Language," *JUTIF*, vol. 5, no. 4, pp. 1165–1171, 2024.
  - [13] M. Defriani, and I. J. Jaelani, "Recognition of Regional Traditional House in Indonesia Using *Convolutional neural network (CNN)* Method," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 4, no. 2, pp. 104–115, 2022, doi: [10.47709/cnahpc.v4i2.1562](https://doi.org/10.47709/cnahpc.v4i2.1562).
  - [14] M. A. Hasan, Y. Riyanto, and D. Riana, "Grape leaf image disease classification using *CNN-VGG16* model," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 4, pp. 218–223, 2021, doi: [10.14710/jtsiskom.2021.14013](https://doi.org/10.14710/jtsiskom.2021.14013).
  - [15] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, "Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis *Android*," *INOVTEK Polbeng - Seri Inform.*, vol. 5, no. 2, p. 297, 2020, doi: [10.35314/isi.v5i2.1654](https://doi.org/10.35314/isi.v5i2.1654).