

Rancang Bangun Smart Engine Untuk Mendeteksi Jenis Biji Kopi Dengan Menerapkan Model Convolutional Neural Network

Geraldo Julius Halim

Program Studi Teknik Informatika, Matana University

Korespondensi penulis : gerald.julius@student.matanauniversity.ac.id

Abstract. *Coffee is an agricultural product that not only functions as a fresh drink, but also comes from an annual plant. Indonesia is known as one of the largest countries in coffee production in the world after Brazil, Vietnam and Colombia in 2017. There are two of the most commonly known varieties of coffee, namely arabica coffee (Coffea arabica) and robusta coffee (Coffea canephora). Due to the similarities between the two types of coffee beans, many people, especially those who are not experienced in the world of coffee, have difficulty telling the difference. Therefore, we need a tool that can help overcome this problem, such as smartengine which can classify arabica and robusta coffee beans. The development of this smartengine follows the Software Developer Life Cycle method using a spiral approach which involves several cycle stages. The first stage involves creating a deep learning model using the Deep Learning Life Cycle method which consists of several steps. In the second stage, deep learning models are provided as a service that can be used by other applications through application programming interfaces (APIs). For smartengine implementation, Google Colab with Keras API and TensorFlow backend is used. This smartengine has the ability to detect coffee beans and also allows the retrain process if needed. Testing is carried out using the blackbox method, where the feature functionality of the smartengine is tested. This research succeeded in developing a smartengine that can detect Arabica and Robusta coffee beans.*

Keywords: *Coffee, Smart Engine, Spiral, API*

Abstrak. Kopi adalah produk pertanian yang tidak hanya berfungsi sebagai minuman segar, tetapi juga berasal dari tanaman tahunan. Indonesia dikenal sebagai salah satu negara terbesar dalam produksi kopi di dunia setelah Brasil, Vietnam, dan Kolombia pada tahun 2017. Terdapat dua varietas kopi yang paling umum dikenal, yaitu kopi arabika (*Coffea arabica*) dan kopi robusta (*Coffea canephora*). Karena kemiripan antara kedua jenis biji kopi tersebut, banyak orang, terutama yang tidak berpengalaman dalam dunia kopi, mengalami kesulitan dalam membedakannya. Oleh karena itu, diperlukan sebuah alat yang dapat membantu dalam mengatasi permasalahan ini, seperti smartengine yang dapat melakukan klasifikasi biji kopi arabika dan robusta. Pengembangan smartengine ini mengikuti metode Software Developer Life Cycle dengan menggunakan pendekatan spiral yang melibatkan beberapa tahap siklus. Tahap pertama melibatkan pembuatan model deep learning dengan menggunakan metode Deep Learning Life Cycle yang terdiri dari beberapa langkah. Pada tahap kedua, model deep learning disediakan sebagai layanan yang dapat digunakan oleh aplikasi lain melalui antarmuka pemrograman aplikasi (API). Untuk implementasi smartengine, digunakanlah Google Colab dengan Keras API dan backend TensorFlow. Smartengine ini memiliki kemampuan untuk mendeteksi biji kopi dan juga memungkinkan proses retrain jika diperlukan. Pengujian dilakukan menggunakan metode blackbox, di mana fungsionalitas fitur dari smartengine diuji. Penelitian ini berhasil mengembangkan sebuah smartengine yang dapat mendeteksi biji kopi arabika dan robusta.

Kata Kunci : Kopi , Smart Engine , Spiral , API

Received Mei 30, 2023; Revised Juni 15, 2023; Accepted Juli 14, 2023

* Geraldo Julius Halim, gerald.julius@student.matanauniversity.ac.id

PENDAHULUAN

Produk pertanian yang umum ditanam di berbagai belahan dunia adalah kopi. Terdapat dua varietas kopi yang umum dikenal, yaitu kopi arabika (*Coffea arabica*) dan kopi robusta (*Coffea canephora*). Namun, kemiripan antara biji kopi ini sering membuat sulit untuk membedakannya. Masalah yang sering muncul adalah ketidakpastian konsumen dalam mengenali jenis kopi yang dijual di kedai kopi..(Prabowo, 2019).

Perkembangan teknologi, seperti kecerdasan buatan (Artificial Intelligence), telah menghadirkan kemajuan yang pesat dalam zaman yang terus berubah. Salah satu bidang yang signifikan dalam kecerdasan buatan adalah Machine Learning, yang berfokus pada pengembangan sistem yang mampu belajar secara otomatis dan akurat dari data yang tersedia. Di dalam ranah Machine Learning, terdapat sub-bidang yang dikenal sebagai Deep Learning, yang mengadopsi algoritma yang terinspirasi oleh struktur otak manusia. Salah satu algoritma populer dalam Deep Learning adalah Convolutional Neural Network (CNN), yang sering digunakan untuk memproses dan menganalisis data gambar, termasuk dalam deteksi objek pada gambar. Dalam konteks ini, penelitian telah dilakukan menggunakan Deep Learning dengan metode Convolutional Neural Network untuk mengidentifikasi jenis tumbuhan berdasarkan citra daun. Dengan menggunakan pendekatan ini, sistem mampu mempelajari fitur-fitur yang terkait dengan citra daun dan mengenali pola-pola yang berkaitan dengan jenis tumbuhan tertentu. Hal ini membuka potensi pengembangan aplikasi yang dapat membantu dalam pengenalan dan klasifikasi tumbuhan berdasarkan citra daun dengan tingkat akurasi yang tinggi.(Ilahiyah & Nilogiri, n.d.). maka dari itu *Deep Learning* dapat digunakan untuk melakukan klasifikasi atau deteksi jenis biji kopi menggunakan metode *Convolutional Neural Network* (CNN). Permasalahan tersebut salah satunya dapat diselesaikan dengan membangun aplikasi *client-server*. Dengan mengintegrasikan model *deep learning* kedalam *server* yang akan dihubungkan ke *client* menggunakan Restful-API. Tujuan penulis dalam pernyataan tersebut adalah untuk mengembangkan sebuah engine deep learning yang mengintegrasikan model Convolutional Neural Network (CNN) dan akan dijalankan di dalam server. Dalam hal ini, aplikasi *client* akan terhubung dengan engine tersebut melalui Restful-API. Salah satu fitur yang diinginkan adalah kemampuan server untuk menghasilkan model deep learning dan melakukan re-train model dengan dataset baru yang diperoleh dari *client* dalam bentuk file zip yang berisi gambar-gambar dengan format PNG.

METODE PENGEMBANGAN

1. Data Collection

Pada penelitian ini, data yang digunakan adalah data primer yang diperoleh dengan cara mengambil foto biji kopi arabika dan robusta. Sebanyak 100 biji kopi arabika dan 100 biji kopi robusta diambil foto menggunakan lensa smartphone Iphone 11. Untuk menciptakan kondisi pencahayaan yang optimal, peneliti menggunakan lampu dari atas serta flash dari smartphone. Latar belakang yang digunakan adalah latar belakang putih.

2. Data Preprocessing

1. Membagi Data

Pertama, dilakukan pembagian data untuk menyamakan jumlah dataset antara kelas biji kopi arabika dan robusta. Setelah itu, dilakukan pembagian dataset ulang dengan rasio 80% sebagai data latih dan 20% sebagai data uji. Selain itu, akan dibuat pula perbandingan dengan pembagian 70% sebagai data latih dan 30% sebagai data uji.

2. Melakukan Grayscale

Proses konversi citra menjadi grayscale dilakukan pada saat pengujian dan sebelum pengujian dilakukan terhadap gambar-gambar yang digunakan.

3. Melakukan Rescale

Data yang digunakan akan diubah ukurannya menjadi 240x240 sehingga semua gambar menjadi ukuran yang sama.



3. Memilih Model

```
[14] from keras import regularizers

cnn_model = tf.keras.models.Sequential([

    tf.keras.layers.Conv2D(32,3, activation='relu', padding='same', input_shape=(240,240,1)),
    tf.keras.layers.MaxPooling2D(pool_size=2),

    tf.keras.layers.Conv2D(64, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=2),

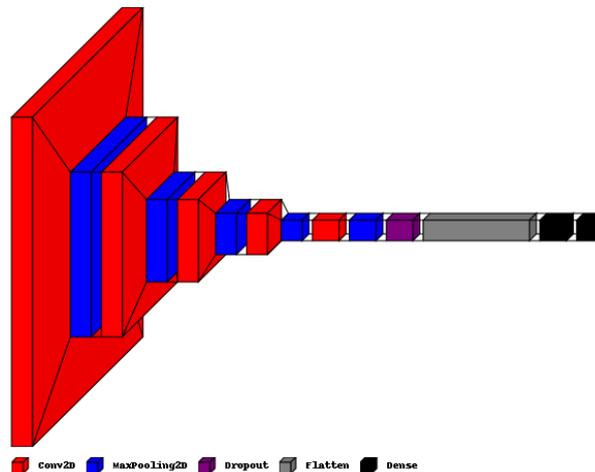
    tf.keras.layers.Conv2D(64, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=2),

    tf.keras.layers.Conv2D(128, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=2),

    tf.keras.layers.Conv2D(256, 3, activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(pool_size=2),

    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')

])
```

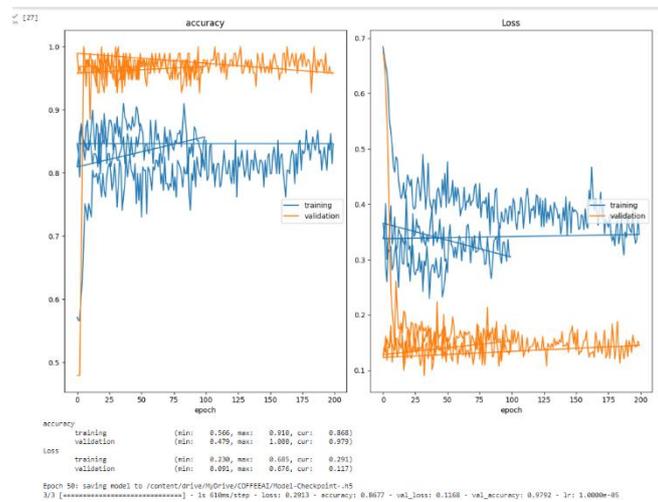


Model yang digunakan dalam penelitian ini merupakan hasil dari percobaan dengan berbagai kombinasi parameter. Jumlah layer yang digunakan telah disesuaikan berdasarkan percobaan untuk mencapai hasil terbaik. Model ini dirancang untuk menerima gambar berukuran 240x240 piksel dengan skala abu-abu. Setiap lapisan konvolusi memiliki ukuran kernel, fungsi aktivasi, dan padding yang sama, tetapi memiliki ukuran filter yang berbeda. Beberapa lapisan konvolusi menggunakan filter yang sama, seperti pada lapisan ketiga yang mengambil hasil maxpooling dari lapisan kedua dan menggunakan 64 filter. Tujuannya adalah untuk terus mengekstraksi fitur yang semakin kompleks. Proses konvolusi berlanjut hingga mencapai lapisan ke-256. Setelah itu, hasil maxpooling dari lapisan kelima masuk ke dalam lapisan dropout dengan tingkat dropout sebesar 0.3. Lapisan dropout ini berfungsi untuk secara acak menghilangkan sebagian unit input selama pelatihan guna mengurangi overfitting pada model. Keluaran dari lapisan dropout kemudian dilipat menjadi bentuk satu dimensi oleh lapisan flatten untuk diproses oleh lapisan selanjutnya. Terdapat dua lapisan Dense, di mana lapisan pertama memiliki 256 unit dengan fungsi aktivasi ReLU, sedangkan lapisan kedua

memiliki 2 unit yang sesuai dengan jumlah kelas yang akan diprediksi. Fungsi aktivasi softmax digunakan pada lapisan output untuk menghasilkan probabilitas prediksi kelas.

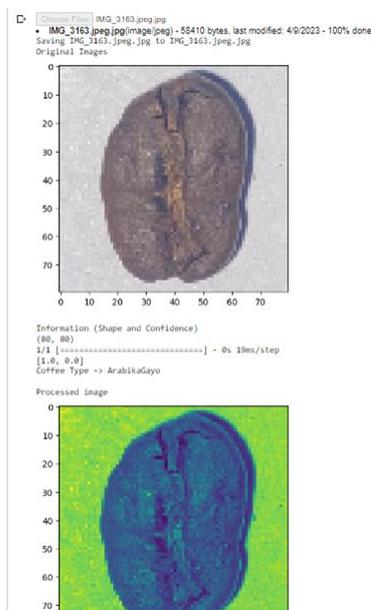
4. Model Training

Setelah arsitektur jaringan saraf tiruan dibentuk, langkah selanjutnya adalah melatih model. Dalam melatih model, beberapa optimisasi dapat diterapkan untuk mencari hasil yang terbaik. dengan epoch sebanyak 50,100 ,dan 200 untuk melihat perbandingan akurasi yang didapatkan.

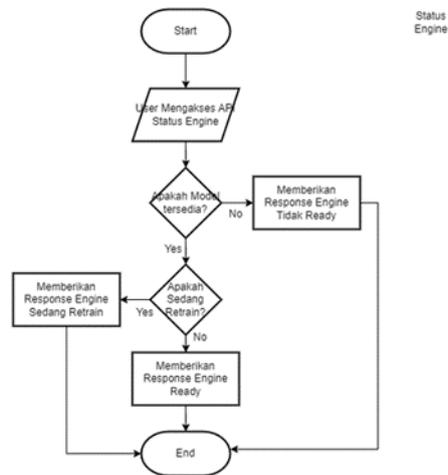
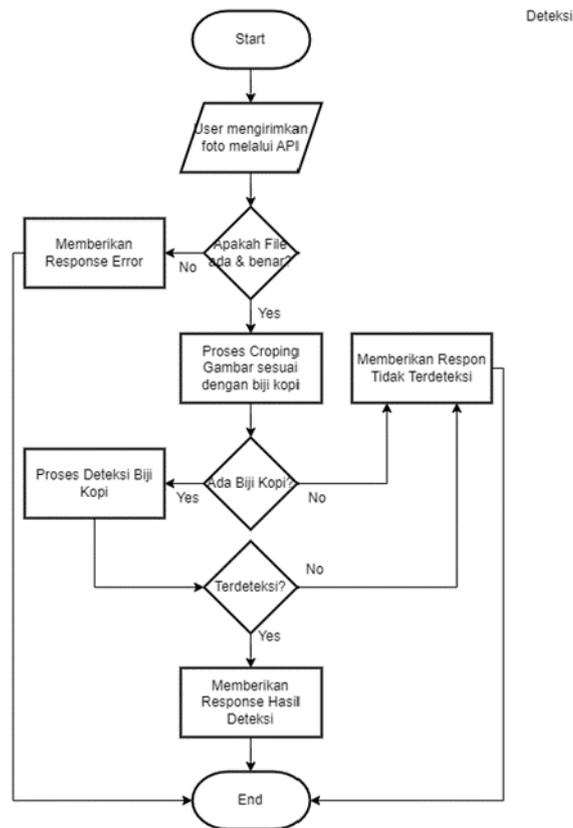


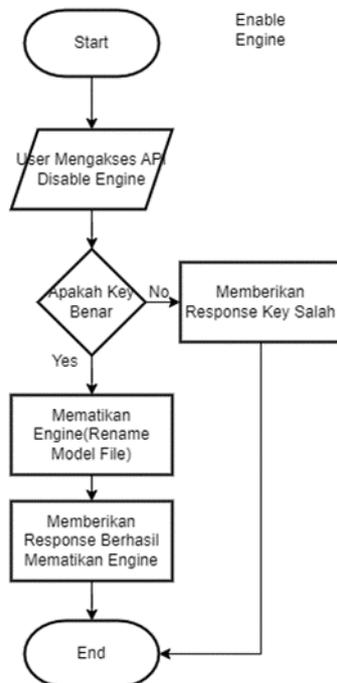
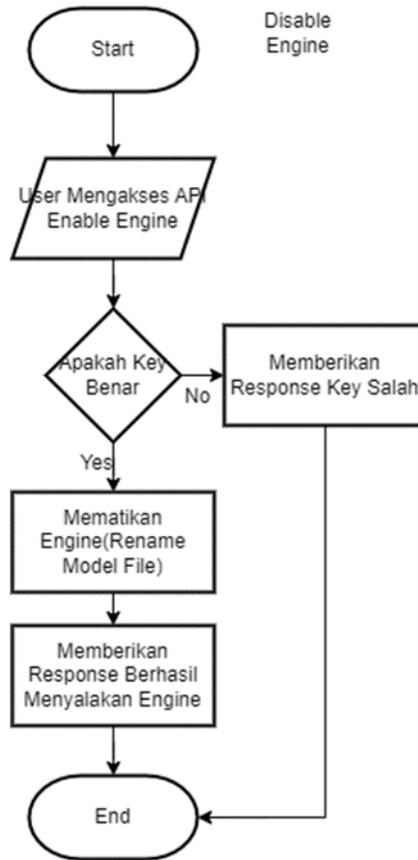
5. Model Evaluation

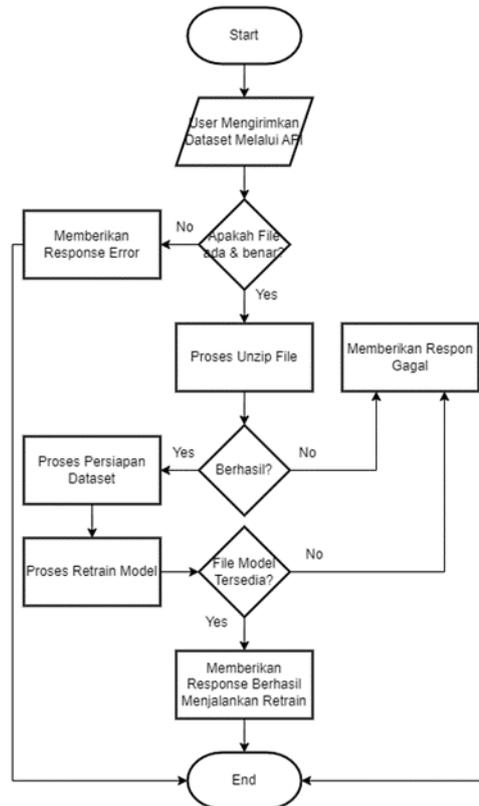
Pengujian adalah salah satu langkah penting dalam proses pengembangan model machine learning. Setelah melakukan pelatihan, model harus diuji untuk mengetahui seberapa baik performanya dalam menangani data baru.



6. Model Serving



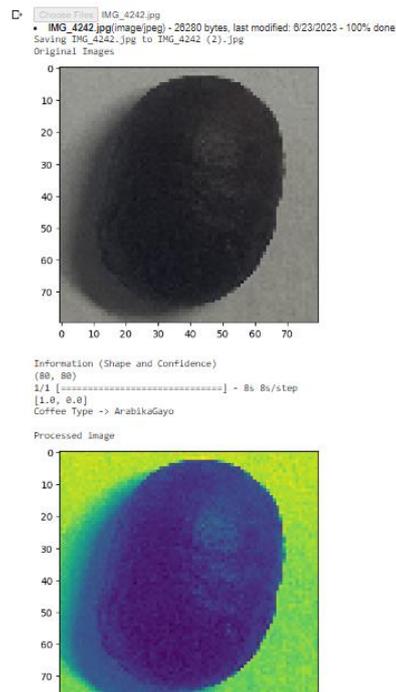




7. Prediction & Inference

Pada tahap ini model yang sudah di-deploy kedalam sebuah engine dapat digunakan untuk mengenali jenis biji kopi berdasarkan citra yang dikirimkan oleh aplikasi client melalui restful-api.

HASIL DAN SARAN



Check Engine Status

- Endpoint : `/api/check/`
- HTTP Method : `GET`
- Request Body :

```
{  
  "key": "secretKey"  
}
```

- Response Body (Success):

```
{  
  "error": false,  
  "message": "Engine Ready",  
  "status": "Online"  
}
```

Disable Engine

- Endpoint : `/api/check/disable`
- HTTP Method : `POST`
- Request Body :

```
{  
  "key": "secretKey"  
}
```

- Response Body (Success):

```
{  
  "error": false,  
  "message": "Engine turned off",  
  "status": null  
}
```

Enable Engine

- Endpoint : `/api/check/enable`
- HTTP Method : `POST`
- Request Body :

```
{  
  "key": "secretKey"  
}
```

- Response Body (Success):

```
{  
  "error": false,  
  "message": "Engine turned on",  
  "status": null  
}
```

Detection

- Endpoint : `/api/detection`
- HTTP Method : `POST`
- Request Body :
 - `image as file`
- Response Body (Success):

```
{  
  "coffeeType": "Robusta Gayo",  
  "error": false,  
  "message": "Success"  
}
```

- Response Body (Fail):

```
{  
  "coffeeType": null,  
  "error": true,  
  "message": "Can't get Image"  
}
```

Retrain

- Endpoint : `/api/retrain`
- HTTP Method : `POST`
- Request Body :
 - `dataset as file`
- Response Body (Success):

```
{
  "error": false,
  "message": "Retrain requested successfully"
}
```

- Response Body (Fail):

```
{
  "error": true,
  "message": "Can't get Datasets file"
}
```

Pembahasan mengenai smartengine yang dibuat dimana pada pembuatan smartengine ini dilakukan beberapa percobaan seperti :

1. Penggunaan perbandingan dataset yang berbeda
2. *Training* epoch yang berbeda
3. *Biji kopi lain*

a. Perbandingan dataset training dan testing

Selama pembuatan smartengine ini, dilakukan perbandingan yang berbeda dalam pembagian dataset antara data latih (*training*) dan data uji (*testing*). Pembagian dataset dilakukan dengan perbandingan 70:30 (70% data training dan 30% data testing) dan 80:20 (80% data training dan 20% data testing).

b. Training Epoch

Pada saat melakukan *training* pada *smartengine* yang dibangun juga dicoba dengan beberapa percobaan pada epoch yang digunakan untuk mendapatkan hasil yang terbaik digunakan beberapa settingan pada epoch yaitu 50 , 100 , dan 200.

No	Komposisi	Epoch	Akurasi
1	70:30	50	0.8143
2	70:30	100	0.8429
3	70:30	200	0.8643
4	80:20	50	0.8677
5	80:20	100	0.8571
6	80:20	200	0.84366

KESIMPULAN

Dalam penelitian ini, telah berhasil dikembangkan sebuah smartengine yang memiliki kemampuan untuk mendeteksi biji kopi. Perancangan smartengine ini mengikuti metode Software Development Life Cycle (SDLC) menggunakan metode spiral. Metode spiral ini terdiri dari beberapa fase yang meliputi pembuatan model deep learning menggunakan metode Deep Learning Life Cycle. Fase pertama melibatkan tahap-tahap dalam pembuatan model deep learning yang meliputi pengumpulan data, preprocessing, pembuatan arsitektur model, pelatihan model, dan evaluasi kinerja model. Fase kedua dari perancangan smartengine ini melibatkan penyediaan model deep learning melalui penggunaan API untuk digunakan oleh aplikasi lain. Smartengine ini dibuat menggunakan Google Colab dengan menggunakan Keras API dan backend TensorFlow. Selain itu, smartengine ini juga memungkinkan dilakukannya retrain untuk meningkatkan performa model. Pengujian pada pembuatan smartengine ini dilakukan menggunakan metode blackbox, di mana fungsionalitas dari fitur-fitur smartengine diuji tanpa melihat implementasi internalnya. Pengujian ini bertujuan untuk memastikan bahwa smartengine berfungsi sesuai dengan yang diharapkan. Dengan adanya smartengine ini, pengguna dapat dengan mudah mendeteksi biji kopi secara otomatis. Smartengine ini dapat digunakan untuk berbagai aplikasi, seperti dalam industri pengolahan kopi atau dalam penelitian tentang biji kopi.

DAFTAR PUSTAKA

- Ilahiyah, S., & Nilogiri, A. (n.d.). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. Retrieved December 20, 2022, from <http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/2254>
- Prabowo, H. (2019). IDENTIFIKASI JENIS KOPI ARABIKA DAN ROBUSTA MENGGUNAKAN DETEKSI TEPI CANNY DENGAN KLASIFIKASI K-NEAREST NEIGHBOR [Universitas Pembangunan Nasional Veteran]. <https://repository.upnvj.ac.id/1704/>