

## KLASIFIKASI SAMPAH ORGANIK DAN ANORGANIK MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Grace Aprisia Bahagia, Mutaqin Akbar

Teknik Informatika, Mercu buana Yogyakarta  
Gejayan Jembatan Merah Yogyakarta, Indonesia  
graceaprisia@gmail.com

### ABSTRAK

Pengelolaan sampah merupakan isu global yang memerlukan solusi inovatif untuk mengoptimalkan proses pemilahan sampah. Salah satu aspek penting dalam pengelolaan sampah adalah pengelompokan sampah menjadi sampah organik dan anorganik. Klasifikasi yang efektif ini memungkinkan penerapan praktik daur ulang yang lebih efisien, meminimalkan dampak negatif terhadap lingkungan, dan memberikan dasar bagi pengembangan kebijakan pengelolaan limbah berkelanjutan. Salah satu pendekatan yang dapat digunakan adalah penggunaan Convolutional Neural Network (CNN) untuk mengklasifikasikan sampah menjadi kategori organik dan anorganik. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi sampah menggunakan metode CNN dengan library TensorFlow. Metode CNN dipilih karena kemampuannya dalam mengekstraksi fitur-fitur penting dari gambar, yang sangat relevan dalam tugas klasifikasi gambar seperti pengenalan sampah. TensorFlow, sebagai salah satu library deep learning yang populer, digunakan untuk membangun dan melatih model CNN. Penelitian ini melibatkan beberapa tahapan, termasuk pengumpulan dataset sampah yang mencakup berbagai jenis organik dan anorganik, pengembangan arsitektur CNN dan pelatihan model menggunakan TensorFlow. Evaluasi dilakukan untuk mengukur akurasi dan kinerja sistem dalam mengklasifikasikan sampah. Hasil penelitian yang telah dilakukan, diperoleh beberapa kesimpulan diantaranya yaitu Dataset yang digunakan dalam pembuatan model CNN adalah data train, dan data validasi. Data train berjumlah 18052 gambar dan data uji berjumlah 4512 gambar. Model CNN dalam penelitian ini menggunakan input shape berukuran 224x224 piksel dan dilakukan augmentasi data didalamnya. Selain itu, menggunakan hyperparameter seperti nilai batch size 32, jumlah epochs 10, learning rate 0,001, dan optimizer Adam. Sehingga memperoleh akurasi sebesar 95% dan loss sebesar 35%.

**Kata kunci :** *Convolutional Neural Network, Tensorflow, Sampah Organik dan Anorganik*

### 1. PENDAHULUAN

Salah satu aspek penting dalam pengelolaan sampah adalah pengelompokan sampah menjadi sampah organik dan anorganik. Klasifikasi yang efektif ini memungkinkan penerapan praktik daur ulang yang lebih efisien, meminimalkan dampak negatif terhadap lingkungan, dan memberikan dasar bagi pengembangan kebijakan pengelolaan limbah berkelanjutan. Dalam konteks ini, teknik kecerdasan buatan, khususnya *Convolutional Neural Network* (CNN) merupakan suatu jenis algoritma dalam *deep learning* yang dapat menerima gambar sebagai sebuah masukan (input) dan mengolahnya dengan tujuan untuk mengidentifikasi objek [1], CNN dapat digunakan untuk mengklasifikasikan citra atau gambar dengan menggunakan pendekatan *Deep Learning*. CNN terbentuk dari beberapa lapisan/layer yaitu lapisan *pooling*, *convolution*, dan *fully connected* [2] oleh karenanya CNN berperan penting pada otomatisasi klasifikasi sampah. Memilih *TensorFlow* yang merupakan sebuah *open source library* untuk mengembangkan dan melatih model *machine learning* atau *deep learning* [3] sebagai perpustakaan utama untuk membuat model *machine learning*, *tensorflow* memanfaatkan data *graphs* [4] dan mengimplementasikan CNN merupakan langkah yang strategis. Dengan menggabungkan kekuatan CNN dan *TensorFlow*, dengan menggunakan

pemrograman *python* dapat digunakan dalam pengembangan *Machine Learning* [5] dengan menggunakan *Google Colaboratory* atau Google Colab merupakan sebuah IDE dalam pemrograman *python* [6] sistem klasifikasi sampah ini diharapkan dapat berkontribusi positif terhadap upaya pengelolaan sampah di masyarakat. Oleh karena itu, latar belakang ini memberikan landasan yang kuat untuk mengembangkan proyek klasifikasi sampah organik dan anorganik berbasis *convolutional neural network* (CNN) menggunakan perpustakaan *TensorFlow*.

### 2. TINJAUAN PUSTAKA

Pada penelitian ini digunakan "Supervised Learning" untuk membantu mengenali jenis tanaman berdasarkan daunnya. Hasil pengujian menunjukkan algoritma CNN memiliki tingkat akurasi yang baik dalam pengidentifikasian piksel dan dapat mengenali setiap jenis daun yang ada. Pengujian ini menghasilkan tingkat akurasi 76%. Dari hasil pengujian dapat dinyatakan bahwa pada penelitian ini CNN adalah *classifier* terbaik [7].

Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network". Penelitian ini melakukan klasifikasi terhadap citra chest x-ray pada pasien yang terindikasi Covid-19. Proses augmentasi

data dalam penelitian ini berupa *height shift range*, *rotation range*, *width shift range*, *brightness range*, *zoom range*, *shear range*, *vertical flip*, *horizontal flip*, *channel shift range*, dan *rescale*. Dalam pelatihan model, penelitian ini menggunakan 2 lapis *layer convolution*, *max pooling*, dan *fully connected layer*. Selain itu, parameter *batch size* diatur sebesar 32, 50 *epochs*, 0.001 *learning rate*, dan Adam sehingga model memperoleh akurasi lebih dari 96% [8].

Penerapan Metode *Convolutional Neural Netork* pada aplikasi deteksi wajah dengan tujuan untuk melakukan monitoring orang yang menjadi DPO dengan cara mentraining citra wajah dan digunakan salah satu metode *deep neural networks* yaitu *Convolutional Neural Networks* (CNN) sebagai pengenalan wajah secara realtime yang sudah terbukti akurat dalam mengenali citra wajah dan di uji keakuratan wajah yang terdeteksi selama 5 menit didapatkan hasil keakuratan benar 77 % dan salah 22 %. Dari total 101 citra yang terdeteksi [9].

Penelitian ini melakukan klasifikasi antara wajah yang memakai masker dengan wajah yang tidak memakai masker dengan menggunakan Metode *Convolutional Neural Network*. Pada penelitian ini, *dataset* diperoleh dari Kaggle yang terdiri dari citra *with mask* dan citra *without mask* dengan jumlah citra sebanyak 4.326 citra. Dalam pembangunan model, ukuran citra di targetkan menjadi 150 x 150 *pixel* dengan besaran *epoch* 20 *epochs*. Proses *training* model dilakukan secara *trial and error* yang mana pada proses pertama menggunakan 3 (tiga) *layer* sehingga memperoleh akurasi mencapai 99.20% pada data *training* dan 70.59% pada data *validation*. Kemudian, proses kedua menggunakan 5 (lima) *layer* sehingga memperoleh akurasi sebesar 98.20% pada data *training* dan 82.35% pada data *validation* [10].

Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode *Convolutional Neural Network* menggunakan Arsitektur VGGNET19 dan ResNet50. Hasil pengujian sistem memperoleh nilai akurasi terbaik sebesar 97.5% dengan menggunakan arsitektur VGGNET19, Optimizer RMSprop, *learning rate* 0.01, *batch size* 32 dan *epoch* 100 [11].

Berdasarkan pemaparan di atas dan didukung oleh beberapa penelitian yang telah dilakukan, maka penelitian ini melakukan klasifikasi terhadap sampah dengan dataset dari *Waste Classification* data.

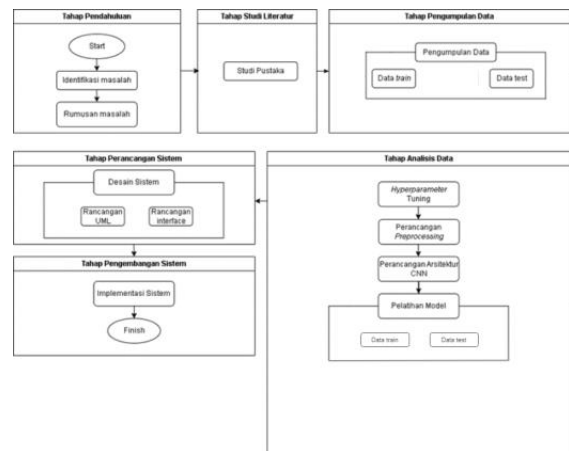
### 3. METODE PENELITIAN

#### 3.1. Jenis Dan Sumber Data

Penelitian ini terdiri dari data primer dan data sekunder, data primer diperoleh dari kamera peneliti dan Google *Photo* yang berhubungan dengan sampah organik dan anorganik, sedangkan data sekunder gambar sampah yang digunakan dalam penelitian ini diperoleh dari Kaggle pada *dataset Waste Classification Data*. *Dataset* ini terdiri dari berbagai jenis citra sampah organik dan anorganik

#### 3.2. Tahapan Penelitian

Tahapan penelitian adalah urutan langkah dari penelitian. Gambaran mengenai langkah-langkah dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, alur dari tahapan penelitian dijelaskan pada sub bab dibawah ini

#### 3.3. Tahap Pendahuluan

Penelitian ini terdiri dari data primer dan data sekunder

- Identifikasi masalah yang berhubungan dengan penelitian yang ingin diteliti. Dalam tahapan identifikasi masalah, peneliti melakukan analisis dan evaluasi terhadap permasalahan tentang sistem aplikasi klasifikasi sampah organik dan anorganik.
- Perumusan masalah yang bertujuan bahwa masalah yang akan diteliti jelas didasari dengan bukti nyata lapangan serta fakta yang ada.

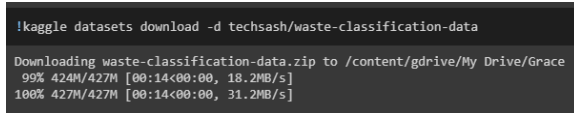
#### 3.4. Tahap Studi Literatur

Pada tahap ini akan dilakukan studi *literatur* atau studi pustaka yang berhubungan dengan penelitian. Studi pustaka dilakukan dengan tujuan untuk memberikan pemahaman materi terkait penelitian ini. Dalam hal ini, studi pustaka yang digunakan meliputi jurnal, buku, artikel ilmiah yang membahas mengenai *Convolutional Neural Network* (CNN), implementasi CNN dengan menggunakan Tensorflow dalam klasifikasi objek, penggunaan *library* Tensorflow *Lite*, serta studi pustaka lainnya.

#### 3.5. Tahap Pengumpulan Data

Proses pengumpulan gambar pada data primer terkait gambar sampah dilakukan melalui galeri *smartphone* peneliti dan kamera *smartphone*. Pada data sekunder, proses pengumpulan data sampah organik dan anorganik diperoleh dari situs Kaggle pada *dataset* yang berjudul *Waste Classification data*. *Dataset* ini terdiri dari *directory train* dan *directory Test*. *Dataset* ini berisi kumpulan gambar sampah organik dan anorganik. Proses pengunduhan *dataset* dilakukan di Google Colaboratory dengan cara

mengunduh API .json pada *account* Kaggle terlebih dahulu. API .json yang telah terunduh harus diimport terlebih dahulu ke dalam Google Drive yang telah disambungkan pada Google colab. Kemudian dilakukan proses pengunduhan *Waste Classification data* sebagaimana dijelaskan pada Gambar 2.



Gambar 2. Download dataset

Setelah proses *download dataset* selesai akan dilakukan proses *unzip dataset* sebelum dilakukan proses pengolahan data. Detail gambar sampah dari data primer dan sekunder dijelaskan dalam Tabel 1.

Tabel 1. Dataset sampah

| No    | Class     | Data Train | Data Test |
|-------|-----------|------------|-----------|
| 1     | Organik   | 1401       | 12600     |
| 2     | Anorganik | 1112       | 9999      |
| Total |           | 2513       | 22599     |

Berdasarkan Tabel 1, diketahui terdapat 2 *class* dalam *dataset*. Setiap *class* mewakili jenis sampah. *Class-class* tersebut digunakan sebagai label dalam *dataset* untuk mengelompokkan data ke dalam *class* yang sesuai

- Data train diperoleh dari *directory train* pada *dataset Waste Classification Data* yang telah diunduh sebelumnya.
- Data test atau data uji merupakan bagian penting dalam proses evaluasi performa model yang telah dilatih menggunakan data *train*. Data uji digunakan untuk menguji sejauh mana model yang telah dilatih sebelumnya dapat menggeneralisasi dan memprediksi dengan baik pada data yang belum pernah dilihat pada saat *training* atau pelatihan model.

### 3.6. Hyperparameter Tuning

Dalam pengembangan model CNN perlu dilakukan tahap *hyperparameter tuning* untuk mencapai performa model yang optimal.

Tabel 2. Hyperparameter tuning

| Parameter     | Nilai |
|---------------|-------|
| Optimizer     | Adam, |
| Learning rate | 0.001 |
| Batch size    | 32    |
| Epochs        | 10    |

*Hyperparameter tuning* yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.

### 3.7. Perancangan Preprocessing

Perancangan *preprocessing* merupakan sebuah tahap yang harus dilakukan sebelum tahap perancangan arsitektur CNN. Dalam tahap ini, dilakukan augmentasi data. Augmentasi data mampu

meningkatkan jumlah data dengan cara memodifikasi data. Proses ini dapat mencegah terjadinya *overfitting* atau *underfitting* pada model terlebih pada data dengan jumlah yang sedikit.

Dalam penelitian ini menggunakan data augmentasi yang terdiri dari melakukan rotasi pada gambar sebesar 10 derajat, *zoom* pada gambar sebesar 4%. Proses augmentasi data dengan menggunakan *ImageDataGenerator library* dari keras adalah dengan melakukan proses *thresholding*, yaitu mengubah nilai citra dari 0-255 menjadi 0-1.

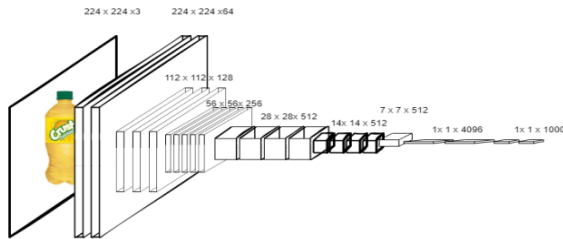
### 3.8. Perancangan Arsitektur CNN

Perancangan arsitektur adalah tahapan yang sangat penting dalam sebuah penelitian yang berfokus pada pengolahan citra atau gambar. *Convolutional Neural Network* merupakan sebuah metode *deep learning* yang efektif dalam mempelajari fitur-fitur pada sebuah citra atau gambar. Dalam perancangan arsitektur CNN, terdapat beberapa aspek yang harus diperhatikan yaitu jumlah *layer*, ukuran kernel, jumlah filter, dan fungsi aktivasi yang digunakan pada tiap *layer*. Selain itu, penggunaan serta pemilihan teknik regulasi seperti *dropout* juga harus dipertimbangkan untuk mencegah adanya *overfitting* pada sebuah model.

Dalam pengembangan arsitektur CNN terdiri dari dua langkah diantaranya adalah tahap *feature learning* dan tahap *classification*. Penelitian ini menggunakan model *Sequential*. Untuk gambar input akan menggunakan citra yang berukuran 224x224x3. Ukuran 224x224x3 pada gambar menunjukkan bahwa gambar tersebut memiliki lebar (*width*) dan tinggi (*height*) masing-masing sebesar 224 piksel dengan tiga *channel* warna, yaitu RGB (*Red, Green, Blue*).

Pada tahap *Feature Learning*, akan dilakukan proses *convolution* dan *pooling* dalam citra input. Penelitian ini menggunakan proses *convolution* yang terdiri dari 4 *layer convolution*. Setiap *layer convolution* mempunyai filter dan kernel dengan jumlah dan ukuran yang berbeda. Masing-masing *convolution* diikuti oleh *max pooling* dengan parameter yang sama pada setiap lapisan atau *layer*.

Setelah tahap *Feature Learning* selesai, kemudian dilanjutkan dengan tahap *Classification* dengan menggunakan lapisan *dense (Fully Connected)*. Pada tahap ini, arsitektur CNN akan mengambil fitur-fitur penting yang dihasilkan pada tahap *Feature Learning* dan kemudian akan digunakan untuk mengklasifikasikan citra *input* kedalam *class* atau label yang sesuai. Tahap *Classification* terdiri dari 2 lapisan *dense*. Setiap lapisan *dense* terdiri dari jumlah *neuron* dan jenis aktivasi yang berbeda. Pada lapisan *dense* pertama digunakan teknik *dropout* yang berfungsi untuk mengurangi kemungkinan *overfitting* pada model. Sedangkan pada lapisan *dense* kedua jumlah *neuron* tergantung pada jumlah *class* pada *dataset*.



Gambar 3. Arsitektur CNN

Dalam penelitian ini ada 2 tahap dalam arsitektur CNN, yaitu tahap *Feature Learning* dan tahap *Classification*. Konvolusi (*Convolution*) pertama dalam *Feature Learning* menggunakan 32 filter dan 3x3 kernel matriks dengan aktivasi ReLu. Setelah itu, terdapat proses *Max pooling2D* dengan ukuran filter 2x2. Setelah itu, dilakukan proses konvolusi kedua dengan 64 filter dan 3x3 kernel matriks dengan aktivasi ReLu. Kemudian, dilakukan kembali proses *Max pooling2D* dengan ukuran filter 2x2, dan dilanjutkan dengan proses konvolusi ketiga dan keempat, masing-masing dengan jumlah filter 128 dan 256, kernel matriks 3x3 dengan aktivasi ReLu serta *Max pooling2D* dengan ukuran filter 2x2. Setelah tahap *Feature Learning* selesai, kemudian akan dilakukan proses *Flatten* untuk mengubah *output* dari proses *convolution* yang berupa *matrix* menjadi vektor dan kemudian dilanjutkan dengan tahap *Classification*.

Pada tahap *Classification*, lapisan *dense* pertama terdiri dari 128 *neuron* dengan aktivasi ReLu dan dilanjutkan dengan teknik *dropout* sebesar 0.5. Lapisan *dense* kedua terdiri dari 12 *neuron* dengan aktivasi *binary* tanpa menggunakan teknik *dropout*.

Arsitektur CNN yang telah dirancang akan dikompilasi. Proses kompilasi CNN dapat dilihat pada Gambar 4 berikut.

```
# Model Compile

OPT = tf.keras.optimizers.Adam(lr=0.001)

model.compile(loss='binary_crossentropy',
              metrics=[tf.keras.metrics.AUC(name='auc')],
              optimizer=OPT)
```

Gambar 4. Compile model

| Model: "sequential"                         |                   |          |
|---------------------------------------------|-------------------|----------|
| Layer (type)                                | Output Shape      | Param #  |
| vgg16 (Functional)                          | (None, 7, 7, 512) | 14714688 |
| dropout (Dropout)                           | (None, 7, 7, 512) | 0        |
| flatten (Flatten)                           | (None, 25088)     | 0        |
| batch_normalization (Batch Normalization)   | (None, 25088)     | 100352   |
| dense (Dense)                               | (None, 1024)      | 25691136 |
| batch_normalization_1 (Batch Normalization) | (None, 1024)      | 4096     |
| activation_1 (Activation)                   | (None, 1024)      | 0        |
| dropout_1 (Dropout)                         | (None, 1024)      | 0        |
| dense_1 (Dense)                             | (None, 1024)      | 1049600  |
| batch_normalization_2 (Batch Normalization) | (None, 1024)      | 4096     |
| activation_2 (Activation)                   | (None, 1024)      | 0        |
| dropout_2 (Dropout)                         | (None, 1024)      | 0        |
| dense_2 (Dense)                             | (None, 1)         | 1025     |
| Total params: 41564993 (158.56 MB)          |                   |          |
| Trainable params: 41510721 (158.25 MB)      |                   |          |
| Non-trainable params: 54272 (212.00 KB)     |                   |          |

Gambar 5. Keseluruhan jaringan

Selain melakukan proses kompilasi juga dapat melihat keseluruhan lapisan jaringan dengan menggunakan `model.summary()`, hasil dari lapisan jaringan dapat dilihat pada Gambar 5.

### 3.9. Pelatihan Model

*Dataset* yang telah diunduh pada tahap sebelumnya akan digunakan dalam pelatihan. Pelatihan model CNN bertujuan untuk melatih model dalam mengenali dan mengklasifikasikan sebuah objek pada gambar yaitu gambar sampah organik dan anorganik. Data yang digunakan dalam proses pelatihan model yaitu data latih (data train), dan data validasi pada *dataset*.

Proses pelatihan model dilakukan dalam beberapa kali *literasi* atau *epochs* dan akan berhenti secara otomatis jika *epochs* sudah mencapai batas.

```
# Model Fitting

model_history=model.fit(train_dataset,
                       validation_data=valid_dataset,
                       epochs = 10,
                       callbacks = callback_list,
                       verbose = 1)
```

Gambar 6. Pelatihan model

terdapat *Callback* yang digunakan untuk menyimpan model dengan akurasi tinggi. Jika model mengalami penurunan akurasi, maka model pada pelatihan tidak akan disimpan.

```
# Defining Callbacks

filepath = './best_weights.hdf5'

earlystopping = EarlyStopping(monitor = 'val_auc',
                              mode = 'max',
                              patience = 5,
                              verbose = 1)

checkpoint = ModelCheckpoint(filepath,
                             monitor = 'val_auc',
                             mode='max',
                             save_best_only=True,
                             verbose = 1)

callback_list = [earlystopping, checkpoint]
```

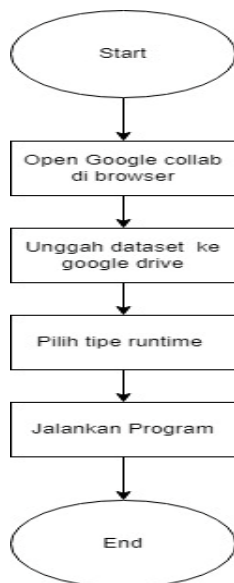
Gambar 7. Callback model

Model akan disimpan dalam *directory* Google drive dengan format *.hdf5*.

### 3.10. Desain Sistem

Tahap desain sistem dilakukan untuk membuat desain dari sistem dalam penelitian ini. Dalam proses pembangunan sistem diperlukan UML (*Unified Modelling Language*) yaitu *flowchart* sistem, *use case diagram*, *sequence diagram*, *class diagram*, *activity diagram* dan rancangan antarmuka sistem.

- a. Flowchart sistem menunjukan tahapan yang harus dilakukan secara berurutan dimana langkah-langkah tersebut di jabarkan dalam bentuk simbol.

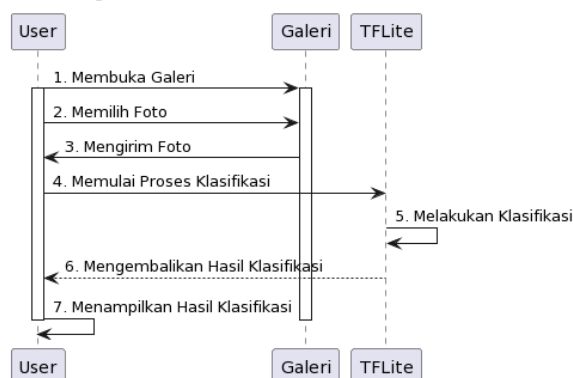


Gambar 8. Flowchart sistem

Dalam *flowchart* diatas menunjukan tahap jalannya sistem dalam penelitian ini yang dimana tahap pertama yaitu membuka website google collab pada browser, kemudian masukkan dataset gambar sampah yang akan dideteksi jenis type sampahnya, selanjutnya pilih tipe runtime pada google collab untuk menjalankan program. Dari tahapan tersebut program akan menjalankan proses klasifikasi sampah berdasarkan gambar yang telah kita input dan akan mendapatkan hasil klasifikasi jenis sampah yang sesuai.

- b. Sequence diagram menjabarkan secara luas terkait alur suatu sistem. *Sequence* diagram pada sistem klasifikasi sampah organik dengan metode *Confolutional Neural Network*.

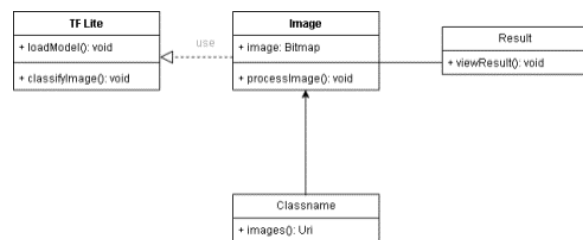
#### Mengambil Gambar dari Galeri dan Klasifikasikan



Gambar 9. Sequence diagram

Pada penelitian ini hanya terdapat 1 aktor, yaitu *user* atau pengguna aplikasi. *Sequence* ini menggambarkan jalannya proses seorang aktor ketika mengunggah gambar sampah, lalu sistem akan mengirim gambar yang telah dipilih untuk diproses oleh model berbasis *Tensorflow Lite* (TF Lite). Model akan melakukan klasifikasi pada gambar tersebut lalu akan menampilkan hasilnya kepada *user*.

- c. Class diagram.



Gambar 10. Class diagram

Merepresentasikan struktur objek dari suatu sistem, dengan menunjukan kelas objek yang membentuk sistem, dan interaksi didalamnya.

## 4. HASIL DAN PEMBAHASAN

Pada penelitian ini, langkah awal yang dilakukan adalah klasifikasi objek menjadi dua kelas jenis sampah yaitu organik dan anorganik dengan menggunakan algoritma Convolutional Neural Network (CNN). Proses utama yang dilakukan yaitu training data, hal ini bertujuan untuk melatih objek sampah dari dataset. Parameter yang digunakan yaitu 10 Epoch, 32 Batch Size, Optimizer Adam dan 0,001 Learning Rate. Untuk ukuran gambar ditetapkan menjadi 224x224 Pixel Sistem rancangan yang digunakan pada saat penelitian dilakukan adalah masukan (*input*), pengolahan data, dan keluaran (*output*) Memuat hasil, Pengujian dan pembahasan tentang skripsi yang telah dilakukan.

### 4.1. Jenis Dan Sumber Data

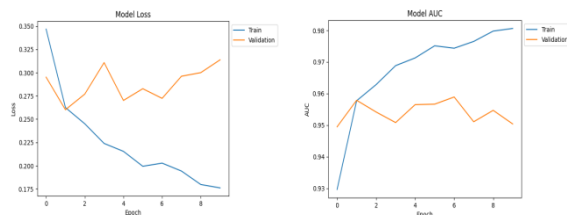
Proses training bertujuan untuk melatih sistem dalam mengenali pola objek sampah organik dan anorganik. Berikut adalah gambar dari training model pada Tabel 3.

Tabel 3. Hasil Training Model

| Epoch | Waktu        | Accuracy loss | Accuracy train | Validation loss | Validaton train |
|-------|--------------|---------------|----------------|-----------------|-----------------|
| 1/10  | 363s 2s/step | 0.3467        | 0.9296         | 0.2951          | 0.9495          |
| 2/10  | 329s 2s/step | 0.2624        | 0.9577         | 0.2601          | 0.9579          |
| 3/10  | 344s 2s/step | 0.2450        | 0.9628         | 0.2769          | 0.9541          |

| Epoch | Waktu        | Accuracy loss | Accuracy train | Validation loss | Validaton train |
|-------|--------------|---------------|----------------|-----------------|-----------------|
| 4/10  | 326s 2s/step | 0.2239        | 0.9688         | 0.3108          | 0.9508          |
| 5/10  | 326s 2s/step | 0.2154        | 0.9713         | 0.2701          | 0.9565          |
| 6/10  | 326s 2s/step | 0.1993        | 0.9751         | 0.2827          | 0.9566          |
| 7/10  | 327s 2s/step | 0.2027        | 0.9743         | 0.2723          | 0.9589          |
| 8/10  | 325s 2s/step | 0.1942        | 0.9765         | 0.2961          | 0.9511          |
| 9/10  | 325s 2s/step | 0.1797        | 0.9798         | 0.3000          | 0.9547          |
| 10/10 | 342s 2s/step | 0.1761        | 0.9806         | 0.3138          | 0.9504          |

Tabel 3 merupakan proses training yang terdapat pada Google Colaboratory. Lamanya proses traing model dipengaruhi oleh nilai batch size dan epoch. Agar proses training tidak terlalu lama maka peneliti menggunakan epoch 10 dan batch size 32. grafik *lost* dan grafik *akurasi* model CNN dapat dilihat pada Gambar 11.



Gambar 11. Grafik training model CNN

Berdasarkan gambar 11 model memperoleh akurasi validasi sebesar 95%. Hal ini menunjukkan bahwasannya model mampu menggeneralisasi dengan baik. Hal ini berarti model belum mencapai konvergen, namun hal tersebut tidak selalu mengindikasikan adanya overfitting atau underfitting pada model. Pengujian dan pembahasan tentang skripsi yang telah dilakukan.

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan diantaranya yaitu *Dataset* yang digunakan dalam pembuatan model CNN adalah data *train*, dan data uji. Data *train* berjumlah 18052 gambar dan data uji berjumlah 4512 gambar. Model CNN dalam penelitian ini menggunakan input *shape* berukuran 224x224 piksel dan dilakukan augmentasi data didalamnya. Selain itu, menggunakan *hyperparameter* seperti nilai *batch size* 32, jumlah *epochs* 10, *learning rate* 0,001, dan *optimizer* Adam, sehingga memperoleh akurasi pelatihan sebesar 98%.

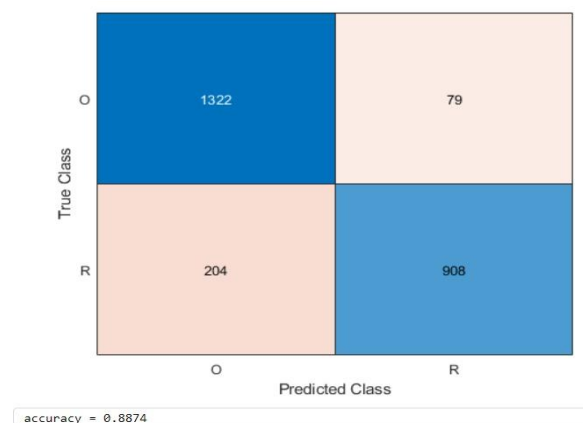
## 4.2. Hasil Pengujian



Gambar 12. Hasil pengujian

Proses testing pada penelitian ini menggunakan 2513 citra gambar dari 2 kelas citra sampah. Proses testing dilakukan untuk menguji performa dari model CNN yang telah diperoleh dari proses learning. Variabel yang akan dinilai pada proses testing adalah organik dan anorganik dari klasifikasi citra yang dihasilkan dari CNN. Hasil testing diperoleh pada Gambar 12.

Berdasarkan gambar diatas dapat dilihat bahwa model CNN berhasil melakukan klasifikasi terhadap citra wortel sebagai sampah organic. Hal ini menunjukkan bahwasannya system ini berhasil melakukan klasifikasi dengan baik.



Gambar 13. Confusion matrix

Berdasarkan gambar 13 dari 1401 data sampah organik, 1322 terdeteksi sebagai sampah organik dan 79 sebagai sampah anorganik. Serta dari 1112 data anorganik bahwasannya 904 terdeteksi sebagai sampah anorganik dan 204 terdeteksi sebagai sampah organik.

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan diantaranya yaitu *Dataset* yang digunakan dalam pembuatan model CNN adalah data *train*, dan data validasi. Data *train* berjumlah 18052 gambar dan data uji berjumlah 4512 gambar. Model CNN dalam penelitian ini menggunakan input *shape* berukuran 224x224 piksel dan dilakukan augmentasi data didalamnya. Selain itu, menggunakan *hyperparameter* seperti nilai *batch size* 32, jumlah *epochs* 10, *learning rate* 0,001, dan *optimizer* Adam. Sehingga memperoleh akurasi sebesar 95% dan *loss* sebesar 35%.

Berdasarkan kesimpulan yang telah diuraikan, saran yang dapat disampaikan yaitu Sistem klasifikasi sampah organik dan anorganik pada penelitian ini bersifat sederhana dan masih banyak kekurangannya sehingga dibutuhkan perancangan sistem yang lebih baik.

#### DAFTAR PUSTAKA

- [1] Didit Iswantoro and Dewi Handayani UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," JURNAL ILMIAH UNIVERSITAS BATANGHARI JAMBI (JIUBJ), 2022.
- [2] A. Mulyanto, E. Susanti, F. Rossi, W. Wajiran, and R. I. Borman, "Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR)," JEPIN (Jurnal Edukasi Dan Penelitian Informatika), 2021.
- [3] Richard Steven Immanuel Sihombing, Rafif Nauval Tuah Siregar, Vijay Sitorus, and Timotius Selar Sitompul, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural," Journal of Creative Student Research (JCSR), 2023.
- [4] J. S. Liunanda, S. Rostianingsih, and A. N. Purbowo, "Implementasi Algoritma YOLO Pada Aplikasi Pendeteksi Senjata Tajam Di Android," Jurnal Teknik, 2020.
- [5] Indra Bakti and Mohamad Firdaus, "Arsitektur Convolutional Neural Network InceptionResNet-V2 Untuk Pengelompokan Pneumonia Chest X-Ray," Jurnal Komputer dan Teknologi (JUKOMTEK), 2023.
- [6] Rangga Gelar Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," Jurnal Teknologi Dan Sistem Informasi Bisnis (JTEKSIS), 2023.
- [7] Felix, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun," Jurnal SIFO Mikroskil, 2020.
- [8] Vina Ayumi and Ida Nurhaida, "Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network," JSAI (Journal Scientific and Applied Informatics), 2021.
- [9] C. W. Wiguna, "Penerapan Metode Convolutional Neural Network Pada Aplikasi Deteksi Wajah Buronan Berbasis WEB," Jurnal Mahasiswa Teknik Informatika, 2022.
- [10] F. M. Qotrunnada and P. H. Utomo, "Metode Convolutional Neural Network Untuk Klasifikasi Wajah Bermasker," Prisma, 2022.
- [11] N. Ibrahim, "Klasifikasi Tingkat Kematangan Pucuk Daun Teh Menggunakan Metode Convolutional Neural Network," ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, 2022.