

Pendeteksi Sampah Metal untuk Daur Ulang Menggunakan Metode *Convolutional Neural Network*

<http://dx.doi.org/10.28932/jutisi.v8i1.4492>

Riwayat Artikel

Received: 9 Februari 2022 | Final Revision: 30 Maret 2022 | Accepted: 30 Maret 2022

Ranti Holiyanti^{#1}, Sukmawati^{#2}, Ikbal Fahmi^{#3}, Chaerur Rozikin^{#4}

Program studi Teknik Informatika, Universitas Singaperbangsa Karawang
Jl. HS. Ronggo Waluyo, Telukjambe Timur, Karawang, Indonesia 41361

¹ranti.holiyanti18241@student.unsika.ac.id

²sukma.wati18043@student.unsika.ac.id

³ikbal.fahmi18109@student.unsika.ac.id

⁴chaerur.rozikin@staff.unsika.ac.id

Abstract — Waste is part material that has no value within the scope of production. If you no longer need it, metal cans can take about 80 to 200 years to decompose. Convolutional Neural Network (CNN) is a part of the supervised learning method that exists in deep learning, where this method is capable of finding representations from images which are amenable to simpler rules solving scene recognition and object detection problems. In this study, CNN method is used as a development model, specifically the ResNet 50 network design, to operate by way of working, namely receive an input in the form of images. The input will be carried out by training that is set using the CNN architecture so that later it will produce an output that can recognize objects as expected in knowing the types of cardboard and glass waste. The implementation of this research uses Python programming language, Anvil, TensorFlow, and Keras libraries. The system has succeeded in detecting the type of metal waste from general waste and assisting third parties, namely implementing it through the website using Anvil. The input shape for CNN modeling in this study is 512x384 pixels, which has a value of 100 eras, and the data set used contains images of metal waste and general waste found 547 images, resulting in an accuracy of 96%.

Keywords— Anvil; Classification; Convolutional Neural Network; Library TensorFlow; Trash.

I. PENDAHULUAN

Di Indonesia Sampah dan limbah yang yang berasal dari penggunaan penduduk setempat sudah menjadi bagian dari suatu permasalahan secara nasional yang ada di Indonesia ini. Sampah yaitu hasil dari aktivitas secara non biologis bagi manusia serta dikatakan sebagai bahan material tersisa yang tidak diminati dan tidak diharapkan oleh penduduk setempat, jika pada masa pemakaiannya telah habis. Bentuk dari sampah tersebut seperti sampah padat, cair, dan gas [1]. Dengan adanya sampah yaitu besarnya jumlah sampah yang berasal dari penduduk pada *unit* volume ataupun berat rata-rata setiap orang per hari. Sampah tersebut merupakan barang atau bahan dalam pemakaiannya memiliki nilai yang biasa saja maupun secara khusus dalam ruang lingkup produksinya, barang yang sudah rusak dan tidak berguna lagi selama hasil atau materi yang dihasilkan sangat berlebihan, pada sebagian besar asal muasalnya berasal dari rumah tangga [2]. Di wilayah Indonesia sampah sudah menjadi permasalahan tinggi. Pada tahun 2018 Badan Pusat Statistik (BPS) menjelaskan bahwa data yang telah diperoleh yaitu pada penduduk di Indonesia masih mengalami kendala dalam melakukan suatu pemilihan sampah dan masih belum memahami dari pemanfaatan sampah yang sudah dibuang. Pada tahun 2019 data yang telah dihasilkan dari Kementerian Lingkungan Hidup dan Kehutanan bahwa sampah yang berada di wilayah Indonesia sudah mencapai 66-67 juta ton, pencapaian tersebut sangat tinggi dibandingkan dari tahun yang sebelumnya yaitu sekitar 64 juta ton [3].

Konsep daur ulang menjadi suatu keputusan yang sangat tepat dalam mengatasi masalah semakin banyaknya sampah yang terus dihasilkan. Tujuan dilakukannya daur ulang untuk menentukan konsumsi bahan baku, melakukan pengecatan pada penggunaan energi, melakukan pengurangan dengan adanya polusi lingkungan, terjadi kerusakan atau kerugian pada lahan sistem ekologi dan melakukan pengecekan pada efek dari rumah kaca. Dalam melakukan pemanfaatan daur ulang pada

material yang sudah tidak dipakai lagi atau sudah buang dari beberapa macam sampah. Seperti pada sampah kardus, plastik, kertas, logam maupun kaca merupakan bahan yang bisa dilakukan daur ulang untuk menghasilkan barang olahan yang terbaru. Selain itu, pada konsep daur ulang juga dapat memperoleh hasil yang maksimal yang dapat memberikan dampak ekonomi menjadi lebih baik lagi [4].

Perubahan yang dialami pada Revolusi Industri 4.0 menyebabkan pengaruh besar bagi dunia. Dalam perkembangan teknologi pada saat ini yaitu suatu pengaruh terjadinya Revolusi Industri 4.0, permasalahan yang sedang dialami di lingkungan dapat teratasi dengan melakukan penerapan pembaruan mengenai kecerdasan buatan, pada ilmu dalam pembelajaran mesin atau *machine learning*. Definisi *Machine Learning* yaitu sebagai pendekatan dalam *Artificial Intelligence* (AI), pada umumnya *machine learning* sudah sering dibutuhkan sebagai pengganti atau melakukan peniruan pada perilaku manusia dalam menyelesaikan suatu permasalahan yang dialami.

Perkembangan pada *machine learning* yaitu mempunyai bidang seperti *deep learning*. Dalam pemanfaatan *Deep Learning* ini sebagai jaringan tiruan untuk untuk melakukan implementasi dari permasalahan yang sedang dialami dengan menggunakan *dataset*. Maka dari itu dapat melakukan representasi *citra* atau gambar dari beberapa identifikasi dengan adanya tingkat kategori yaitu seperti melakukan *object classification*, melakukan *scene recognition*, serta *object detection* dikatakan sebagai CNN [5].

Dalam penelitian ini menerapkan penyelesaian permasalahan pada sampah metal (logam kaleng) Soda, dan minuman, dan makanan kemasan dalam penyajiannya menggunakan kaleng lainnya yang menjadi salah satu minuman dan makanan favorit di kalangan masyarakat pada saat ini. Selain memiliki rasa yang enak, dan segar, serta dalam penyajiannya sangat higienis, makanan jenis ini sering ditemukan. Maka dari itu sesuatu yang nyaman serta praktis bagi kita ternyata membawa pengaruh buruk terhadap bumi.

Jenis sampah kardus dan sampah logam yang sulit terurai, sampah tersebut yang nantinya akan di daur ulang dan dipergunakan, dengan melakukan cara membangun sebuah model dengan menggunakan metode CNN dan arsitektur jaringan *ResNet 50* untuk dilakukannya identifikasi jenis sampah kardus dan *glass*. *Dataset* terdiri dari beberapa jenis, akan tetapi lebih difokuskan hanya kepada sampah logam dan sampah kardus saja. Dalam melakukan pembangunan model ini juga melakukan penerapan proses data augmentation serta transfer *learning*, harapannya dapat mencapai akurasi yang sangat baik. dalam pengimplementasian penelitian ini menggunakan Bahasa pemrograman Python, Anvil, dan menggunakan *library TensorFlow* serta *Keras*. Sistem berhasil mendeteksi jenis sampah metal dari sampah umum dan dibantu dengan pihak ketiga yaitu dengan melakukan implementasi melalui *website* dengan menggunakan Anvil. Terdapat bentuk masukan pada model CNN ini yaitu 512x384 piksel, nilai jumlah *epoch* yaitu 100, serta *dataset* yang digunakan terdapat *citra* sampah metal dan juga sampah umum yang berjumlah 547 *citra* sehingga menghasilkan akurasi sebesar 96%.

Keberhasilan dari proses identifikasi sampah metal ini merupakan sebuah bagian awal untuk melakukan implementasi pada tempat sampah sehingga mempermudah dalam melakukan pendeteksian serta dalam melakukan proses daur ulang.

II. METODE PENELITIAN

Metode penelitian didefinisikan sebagai tahapan yang dilewati dalam melakukan suatu penelitian. Penelitian ini mengimplementasikan model CNN untuk mendeteksi jenis sampah logam. Dapat dilihat pada gambar 1 merupakan Rancangan pembuatan sistem dimulai dari awal sampai uji coba.



Gambar 1. Flowchart

A. Identifikasi Masalah

Makanan dalam kemasan kaleng saat ini sangat diminati oleh masyarakat, karena dianggap sebagai lebih aman serta mudah dibawa kemana-mana dan lebih higienis, sehingga menjadi faktor meningkatnya produksi makanan atau barang lain menggunakan kemasan kaleng atau metal ini, tingkai besarnya produksi sampah tersebut mempengaruhi lingkungan, karena prosesnya sangat lama dalam jangka waktu yang dibutuhkan untuk terurai sampah kemasan kaleng ini, rendahnya kesadaran masyarakat akan hal tersebut dan tidak pedulinya masyarakat untuk menggunakan kembali kemasan tersebut untuk fungsi yang lain.

B. Studi Literatur

Pada tahap Studi Literatur ini digunakan untuk melakukan pencarian serta mempelajari dasar-dasar secara teoritis yaitu seperti mempelajari Jurnal, ebook dan lainnya yang berhubungan dengan penelitian sebelumnya telah dilakukan. Seperti pada konsep pendeteksi sampah, pengolahan data sampah dan metode *Deep Learning* mengenai CNN dan arsitektur jaringan *ResNet 50*. Selanjutnya setelah melakukan studi *literatur*, sehingga didapatkan rumusan Langkah seperti apa yang akan dikerjakan, dan hal baru seperti apa yang akan dihasilkan sebagai halnya pada penelitian ini. Berdasarkan penjelasan yang sudah diuraikan sebelumnya maka dari itu pada penelitian ini menggunakan Teknik studi literatur ini guna untuk mencari dan memenuhi dari dasar-dasar secara teoritis yang digunakan dalam penelitian ini.

C. Proses Persiapan

Pada tahap proses persiapan data ini terbagi dalam beberapa proses seperti: melakukan suatu proses koleksi data serta rancangan aplikasi. Dalam mendapatkan data peneliti disini memperoleh dari sumber data set yaitu didapatkan dari Kaggle yaitu sebagai komunitas ilmu data terbesar di dunia serta menggunakan alat dan sumber daya yang kuat untuk membantu peneliti untuk mencapai sebuah tujuan yaitu ilmu data yang kita dapatkan. Kemudian dataset tersebut nantinya akan dilakukan *preprocessing* sehingga bisa menghilangkan *noise* pada bagian data.

Berdasarkan konsep akhir yang telah dilakukan penelitian ini dalam melakukan perancangannya diarahkan ke dalam rancangan aplikasi dengan menggambarkan kebutuhan UI dari Anvil dengan merancang komponen dari *Toolbox* sesuai dari kebutuhan penelitian ini. Selain itu, dalam melakukan proses perancangan pada *Machine Learning* menggunakan *Tools* python dengan memasukan *library Sequential, Conv2D, MaxPooling2D, Flatten, Dense, Image Data Generator* dan keras.

D. Pengumpulan Data

Dalam melakukan tahap pengumpulan data ini terdapat data yang sudah dikumpulkan seperti gambar bersifat format .jpg. Tahap ini dilakukan pencarian dengan menggunakan *Dataset* dari Kaggle.com. *Dataset* yang telah didapatkan berupa citra atau gambar sampah yang keadaannya belum melakukan klasifikasi, objek sampah yang diperoleh sebanyak 410 gambar dan metal (logam) sebanyak 410 gambar. Selanjutnya setelah mendapatkan data gambar sampah dan logam. Langkah berikutnya yaitu melakukan klasifikasi *dataset* tersebut ke dalam *training* data dan *set* data. Pada penelitian ini dataset tersebut akan diinputkan pada proses sistem, setelah itu menyiapkan data *training* data dengan menggunakan CNN, pada saat melakukan proses *training* metode CNN yang nanti akan dilakukan tahap uji seberapa besar tingkat akurat dan seberapa besar performa dari yang sudah dihasilkan.

E. Algoritma Convolutional Neural Network

CNN merupakan komponen dari algoritma *Deep Learning* atau pembelajaran yang mendalam yaitu dalam melakukan pembangunan diawali *Multi Layer Perceptron* atau MLP yang dibangun untuk melakukan pengolahan data berbentuk *grid*, seperti pada *citra* atau gambar dua dimensi, seperti suara. CNN diterapkan untuk melakukan klasifikasi sebuah data yang mempunyai identitas dengan menerapkan metode *supervised learning*, Pengerjaan proses *supervised learning* yaitu terdapat sebuah data yang akan dilakukan pelatihan dan didalamnya terdapat suatu variabel yang dijadikan target, Maka dari itu tujuan dari metode CNN ini untuk melakukan pengelompokan suatu data ke data yang sudah ada saat ini. CNN biasanya digunakan untuk melakukan pengenalan suatu benda atau pengetahuan, serta melakukan pendeteksian serta pemetakan atau segmentasi objek.

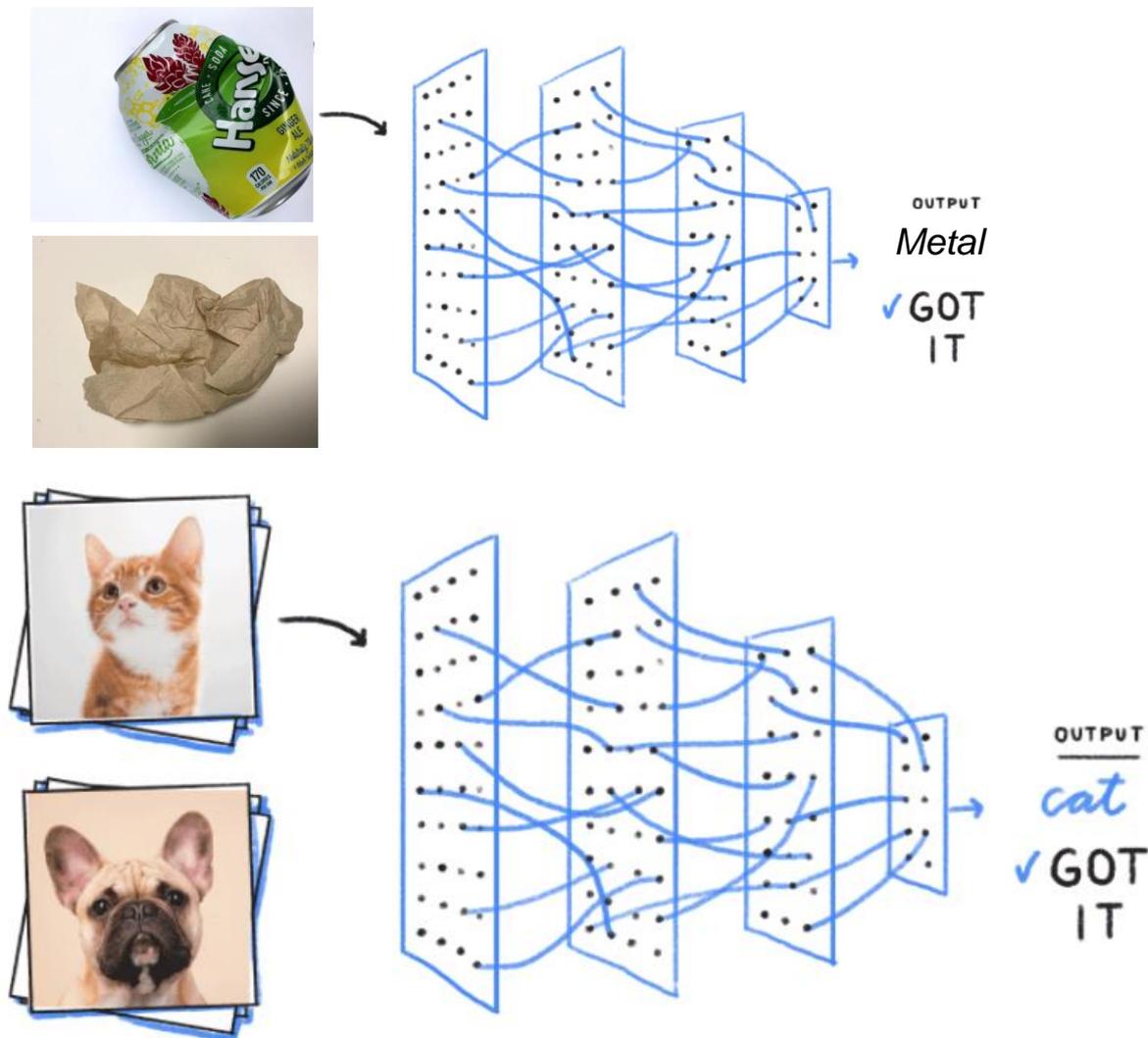
Algoritma CNN yang dilakukan pada penelitian ini bertujuan untuk melakukan klasifikasi sebuah *citra* atau gambar dari sampah, dan jaringan ResNet 50 untuk dilakukannya identifikasi jenis sampah kardus dan *glass*. Sangat berbeda dari algoritma klasifikasi yang biasanya, apabila pada algoritma klasifikasi pada umumnya hanya melakukan proses pengelompokan fitur atau tampilan serta klasifikasi dilakukan dengan terpisah maka dari itu pemodelan algoritma yang termasuk dari bidang *deep learning* ini akan melakukan pengelompokan pada tampilan selanjutnya melakukan klasifikasi

citra atau gambar pada bagian satu proses. Dan juga bisa disebut sebagai ekstraksi atau pengelompokan *fitur* pada algoritma CNN ini juga akan ikut melakukan pembelajaran.

F. Training dan Testing

Selanjutnya setelah melakukan tahap pra proses data, tahapan berikutnya akan melakukan proses *training* dan *testing* pada data dengan menggunakan model *architecture* dari *Deep Learning* yaitu CNN. Pada inputan data dengan ukuran 512×384 pixel. *Output* yang dihasilkan dalam proses ini menghasilkan *recognition* data.

Pada penelitian ini akan membangun sebuah CNN dengan menggunakan data latih dari beberapa ratus gambar sampah kardus dan sampah logam, supaya bisa mendeteksi yang mana sampah kardus dan sampah logam [6] Pada penelitian ini akan membangun sebuah CNN dengan menggunakan data latih dari beberapa ratus gambar sampah umum dan sampah logam supaya bisa mendeteksi yang mana sampah umum atau sampah logam. Gambar 2 merupakan suatu pemodelan yang digunakan pada penelitian ini.



Gambar 2. Training dan Testing [7]

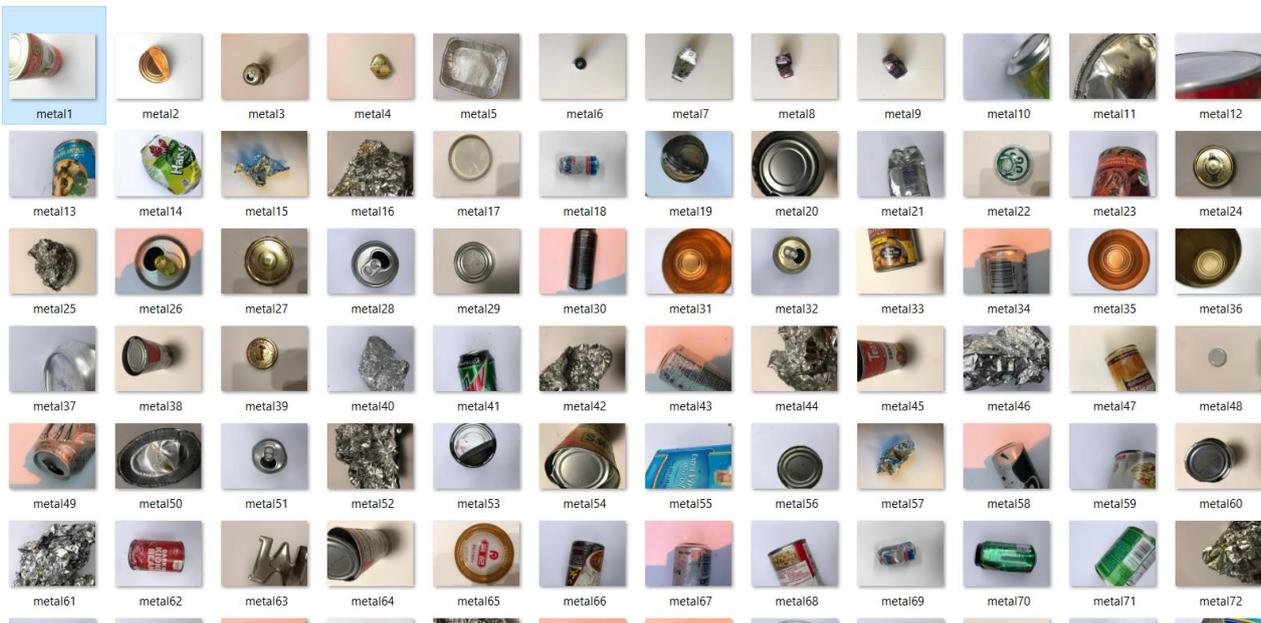
G. Uji Coba

Sebelum melakukan tahap uji coba terlebih dahulu harus melakukan proses pengkodean (*coding*). Pengkodean (*coding*) yaitu bahasa pemrograman yang menerjemahkan *design* kedalam bahasa yang dapat dipahami oleh komputer. Pada tahap uji coba ini yaitu tahapan secara nyata (*real*) dalam melakukan pengerjaan sistem yaitu dapat melakukan proses pendeteksian sampah pada data. Maksudnya dalam penggunaan komputer akan dioptimalkan dalam tahap ini. Setelah selesai melakukan proses pengkodean (*coding*), selanjutnya membuat proses uji coba dengan menggunakan *set testing*.

III. HASIL DAN PEMBAHASAN

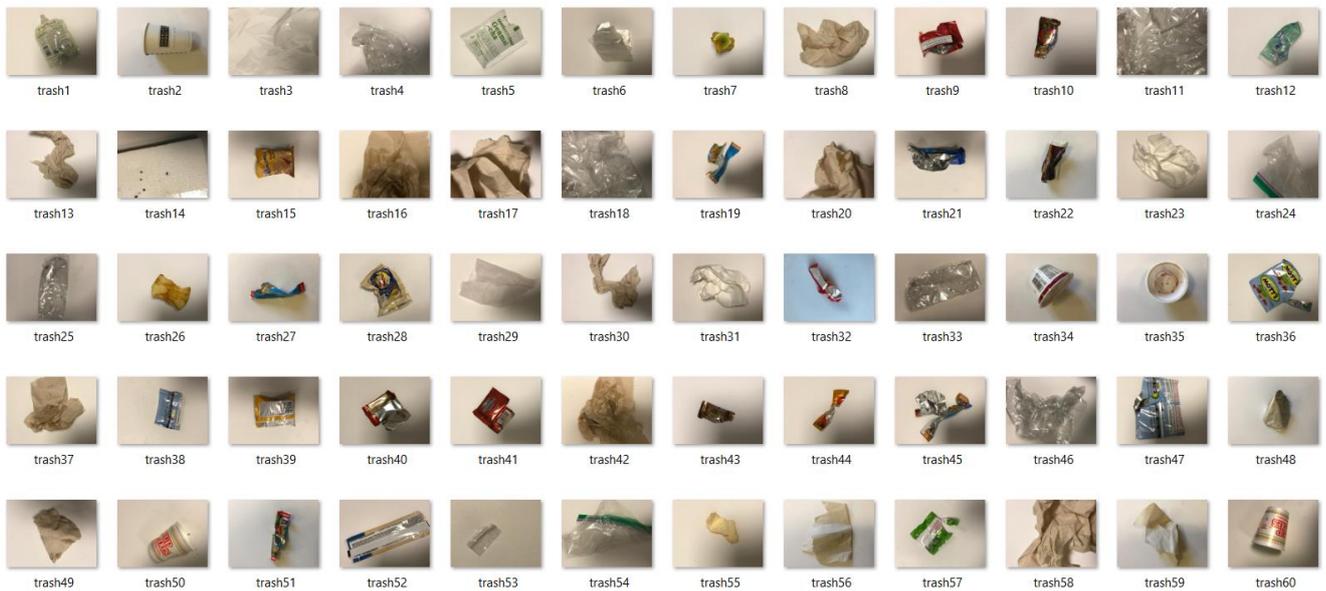
A. Pengumpulan Gambar

Berikut merupakan pengumpulan gambar *dataset* untuk *training set* dan *test set* pada sampah logam yang berjumlah 410. Dapat dilihat pada gambar 3 merupakan kumpulan gambar *dataset* untuk *training set* dan *test set* pada sampah logam.



Gambar 3. Hasil pengambilan gambar Metal (logam)

Selanjutnya pengumpulan gambar *dataset* untuk *training set* dan *test set* pada sampah umum seperti kertas, plastik, *paper*, dan kardus yang berjumlah 137. Dapat dilihat pada gambar 4 merupakan *dataset* untuk *training set* dan *test set* pada sampah umum seperti kertas, plastik, *paper*, dan kardus.



Gambar 4. Hasil pengambilan gambar Sampah

Setelah melakukan rancangan arsitektur CNN. Tahapan berikutnya yaitu melakukan proses training data untuk menghasilkan suatu model. Pada Langkah awal dalam melakukan pelatihan, memperoleh model yang belum menghasilkan tingkat akurat.

B. Perbandingan Loss dan Akurasi Pelatihan

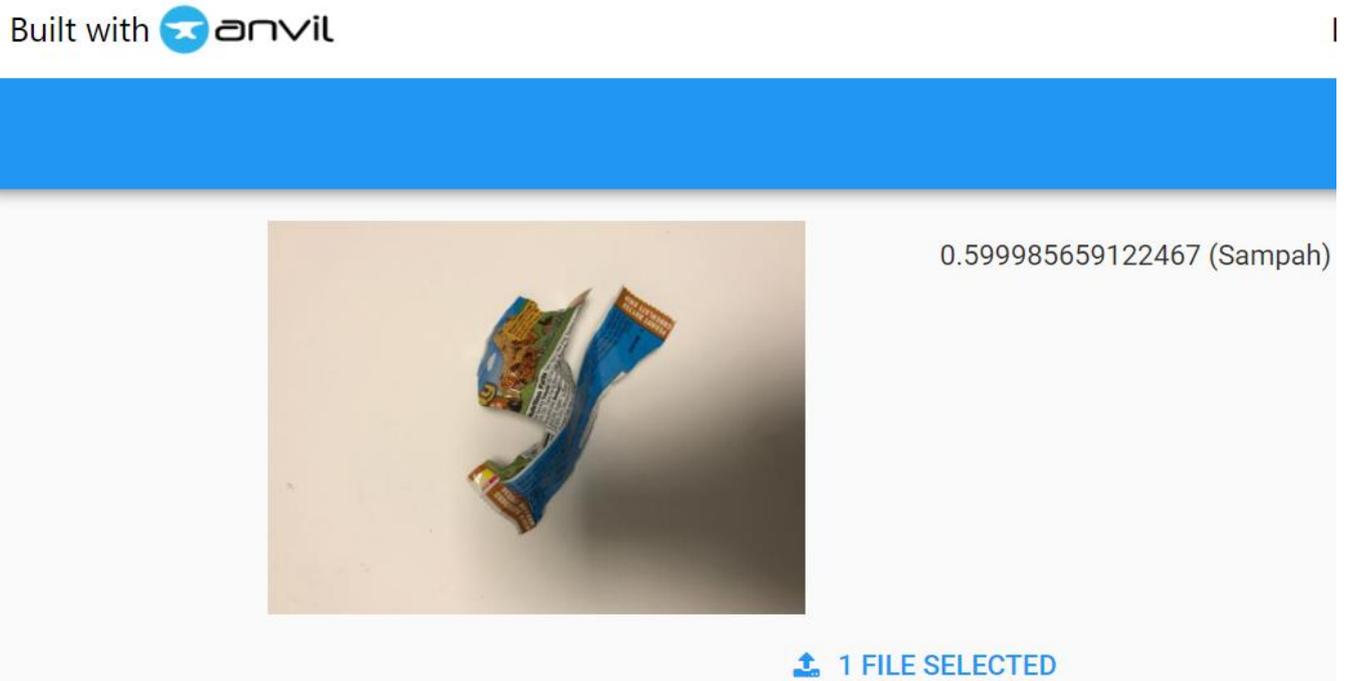
Pada tahapan proses *training* pertama ini merupakan suatu proses guna untuk memperoleh model *machine learning* atau pembelajaran mesin. Pada saat melakukan *training* atau pelatihan sebelumnya telah diatur sedemikian rupa untuk jumlah *step* per *epoch* yaitu 50. Gambar 5 merupakan pembelajaran mesin pada *loss* akurasi.

```
classifier.fit(training_set, steps_per_epoch = len(training_set)//2,  
epochs = 50, validation_data = test_set, validation_steps = len(test_set)//2)
```

Epoch 1/50
9/9 [=====] - 15s 2s/step - loss: 1.1707 - accuracy: 0.6319 - val_loss: 0.9860 - val_accuracy: 0.7604
Epoch 2/50
9/9 [=====] - 7s 745ms/step - loss: 0.5577 - accuracy: 0.7692 - val_loss: 0.4338 - val_accuracy: 0.7396
Epoch 3/50
9/9 [=====] - 4s 419ms/step - loss: 0.4219 - accuracy: 0.7577 - val_loss: 0.3242 - val_accuracy: 0.8785
Epoch 4/50
9/9 [=====] - 2s 243ms/step - loss: 0.3834 - accuracy: 0.8154 - val_loss: 0.3628 - val_accuracy: 0.8056
Epoch 5/50
9/9 [=====] - 2s 170ms/step - loss: 0.4137 - accuracy: 0.8154 - val_loss: 0.3070 - val_accuracy: 0.8750
Epoch 6/50
9/9 [=====] - 2s 270ms/step - loss: 0.3392 - accuracy: 0.8403 - val_loss: 0.3348 - val_accuracy: 0.8542
Epoch 7/50
9/9 [=====] - 3s 285ms/step - loss: 0.3104 - accuracy: 0.8808 - val_loss: 0.2769 - val_accuracy: 0.8681
Epoch 8/50
9/9 [=====] - 2s 178ms/step - loss: 0.2938 - accuracy: 0.8715 - val_loss: 0.3726 - val_accuracy: 0.8229
Epoch 9/50
9/9 [=====] - 1s 148ms/step - loss: 0.4031 - accuracy: 0.8192 - val_loss: 0.2830 - val_accuracy: 0.8819
Epoch 10/50
9/9 [=====] - 2s 167ms/step - loss: 0.3550 - accuracy: 0.8500 - val_loss: 0.3251 - val_accuracy: 0.8507
Epoch 11/50
9/9 [=====] - 1s 160ms/step - loss: 0.2763 - accuracy: 0.8576 - val_loss: 0.2749 - val_accuracy: 0.8889
Epoch 12/50
9/9 [=====] - 1s 162ms/step - loss: 0.3440 - accuracy: 0.8462 - val_loss: 0.2675 - val_accuracy: 0.8993
Epoch 13/50
9/9 [=====] - 1s 151ms/step - loss: 0.2917 - accuracy: 0.8731 - val_loss: 0.3360 - val_accuracy: 0.8333
Epoch 14/50
9/9 [=====] - 1s 163ms/step - loss: 0.3000 - accuracy: 0.8731 - val_loss: 0.3940 - val_accuracy: 0.8194
Epoch 15/50
9/9 [=====] - 1s 166ms/step - loss: 0.3310 - accuracy: 0.8500 - val_loss: 0.2759 - val_accuracy: 0.8924
Epoch 16/50
9/9 [=====] - 1s 162ms/step - loss: 0.2657 - accuracy: 0.8854 - val_loss: 0.3251 - val_accuracy: 0.8507
Epoch 17/50
9/9 [=====] - 1s 161ms/step - loss: 0.2488 - accuracy: 0.9097 - val_loss: 0.2682 - val_accuracy: 0.8889
Epoch 18/50

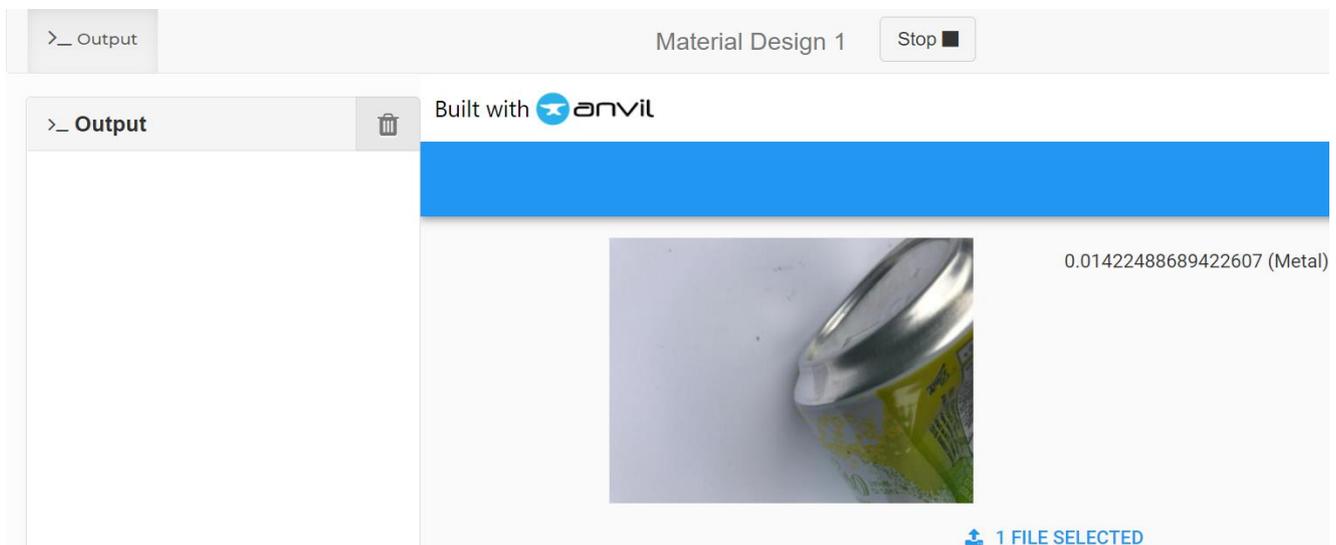
Gambar 5. Loss dan Akurasi

Selanjutnya melakukan pengujian terhadap *citra* yang diuji. Dapat dilihat pada gambar 6 merupakan proses pengujian terhadap citra yang diuji.



Gambar 6. Deteksi sampah

Selanjutnya setelah didapatkan model, kemudian melakukan pengujian mendeteksi sampah berjenis *metal* (logam) dengan mengambil sampel berupa sampah logam, dapat kita lihat pada hasil gambar 7, yaitu sebagai berikut :



Gambar 7. Deteksi Sampah

C. Source Code Program

Langkah Pertama melakukan *import* semua *package* / *library* yang diperlukan

1. *Import Library*

Digunakan untuk melakukan pemodelan *neural network* seperti *sequential network*. Agar dapat mengetahui dasar dari inisialisasi *neural network* selain dari *sequential* ada juga seperti *graph*. Selanjutnya melakukan *import* Conv2D fungsinya untuk mengoperasikan konvolusional dari gambar (*citra*) yang akan dilatih. lalu *import* Max Pooling 2D untuk pengoperasian pada *pooling*. Dan melakukan *import* Flatten fungsinya untuk *flattening*. Dan juga melakukan *import* Dense fungsinya untuk mengoperasikan *full connection neural network* yang ada di penelitian ini. Pada gambar 8 merupakan gambar melakukan proses *import library*.

```
In [4]: #Import the Keras Libraries and packages
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
```

Gambar 8. *Import library*

2. *Import anvil server*

Setelah mendapatkan kuncinya, selanjutnya tambahkan kode seperti pada gambar 8 ini ke *Notebook* Jupyter, karena sudah mempunyai model untuk artikel sebelumnya, jadi hanya menambahkan ke sel terakhir dari proyek tersebut. Dapat dilihat pada gambar 9 merupakan proses *import anvil server*.

```
In [6]: import anvil.server

anvil.server.connect("DAIS2VRD6PJJRAE2IG4L7WQW-N2VTGLSPK6TPDQPN")
```

Gambar 9. *Import anvil server*

3. *Object Sequential Class*

Selanjutnya, akan melakukan pembuatan *object sequential class*, dalam menentukan langkah kompleks konvolusinya yaitu dapat dilihat pada gambar 9, nilai *filter* parameter pertama yaitu (32). parameter kedua sebagai dimensi *filter* nya yaitu (3×3), parameter keempat yaitu sebagai fungsi kegiatan yang sudah digunakan (relu) sehingga menggunakan *rectifier function*. Dan akan mulai melakukan penambahan objek *classifier* ke dalam *layer pooling* yang sudah ada. *Pooling* yang ada di dalam penelitian ini berbentuk matrik 2×2 sebagai *pixel loss* minimum serta *precise region* dimana feature nya akan didistribusikan.

Kemudian yaitu melakukan proses *flattening*. Data *pooling* yang telah dimiliki berupa *array* 2 dimensi selanjutnya dilakukan konversi menjadi satu data dimensi *single vector*. Pada gambar 10 merupakan proses melakukan *object sequential class*.

```
In [7]: classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

Gambar 10. *Object sequential class*

4. *Proses flattening*

Kemudian melakukan proses *flattening*. Terdapat dua dimensi selanjutnya melakukan konversi menghasilkan data satu dimensi *single vector*. Lalu selanjutnya yaitu melakukan proses *dense*. *Dense* digunakan untuk penambahan *layer* yang terhubung dengan semuanya (*fully connected*). Satuan (*Units*) membuktikan jumlah *node* yang harus ada di dalam di *hidden layer*, seperti nilainya antara jumlah *input node* dan *output node*. lalu fungsi aktivasinya akan memerlukan relu. Selanjutnya melakukan inisialisasi *layer output* yang telah diperoleh, yang hanya terdapat pada satu *node* atau *unit* dikarenakan pada saat melakukan klasifikasi ini termasuk dalam *binary classification*.

Kode berikut membuktikan bahwa pada penelitian ini hanya memerlukan satu *node* atau *unit*, serta fungsi dari aktivasi sigmoid pada proses *layer* yang paling akhir.

Setelah selesai dalam melakukan pembangunan model CNN, Langkah selanjutnya yaitu melakukan kompilasi.

- parameter optimasi untuk penentuan algoritma *stochastic gradient descent*
- parameter *loss* digunakan untuk menyelesaikan *loss function*
- parameter *loss* digunakan untuk menyelesaikan *loss function*

Pada gambar 11 merupakan proses *flattening*.

```
In [8]: classifier.add(Flatten())
classifier.add(Flatten())
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
classifier.save('classifier.h5')
```

Gambar 11. Proses *flattening*

5. Source Code Preproses

Sebelum melakukan tahap *fitting* model, cara yang harus dilakukan terlebih dahulu yaitu pada data *training*, pada penelitian ini supaya tidak terjadi *overfitting*. Yang terjadi pada *source code* berikut merupakan proses paduan data yang dihasilkan dari sebuah gambar (*citra*) yang mirip dengan melakukan pengoperasian seperti proses *flipping*, *rotasi*, *blurring* dan lain sebagainya. Pada gambar 12 merupakan suatu pemrosesan pada *flattening*.

```
In [9]: train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('training_set',
target_size = (64, 64), batch_size = 32, class_mode = 'binary')
test_set = test_datagen.flow_from_directory('test_set', target_size = (64, 64), batch_size = 32, class_mode = 'binary')

Found 547 images belonging to 2 classes.
Found 547 images belonging to 2 classes.
```

Gambar 12. Source Code Preproses

6. Fitting Model

Selanjutnya melakukan *fitting* model. *Source code* berikut, '*stepper epoch*' membuktikan bahwa terdapat banyaknya *citra* yang dilatih yang tersedia di dalam *folder training set*. Dapat dilihat pada gambar 13 merupakan proses fitting model.

```
In [10]: classifier.fit(training_set, steps_per_epoch = len(training_set)//2,
epochs = 100, validation_data = test_set, validation_steps = len(test_set)//2)

Epoch 95/100
9/9 [=====] - 5s 641ms/step - loss: 0.3060 - accuracy: 0.8610 - val_loss: 0.1329 - val_accuracy: 0.9
514
Epoch 96/100
9/9 [=====] - 5s 587ms/step - loss: 0.1971 - accuracy: 0.9167 - val_loss: 0.1261 - val_accuracy: 0.9
653
Epoch 97/100
9/9 [=====] - 5s 622ms/step - loss: 0.1994 - accuracy: 0.9271 - val_loss: 0.1293 - val_accuracy: 0.9
479
Epoch 98/100
9/9 [=====] - 5s 632ms/step - loss: 0.1210 - accuracy: 0.9653 - val_loss: 0.1130 - val_accuracy: 0.9
688
Epoch 99/100
9/9 [=====] - 5s 614ms/step - loss: 0.1484 - accuracy: 0.9421 - val_loss: 0.0942 - val_accuracy: 0.9
757
Epoch 100/100
9/9 [=====] - 5s 576ms/step - loss: 0.1330 - accuracy: 0.9382 - val_loss: 0.1024 - val_accuracy: 0.9
688

Out[10]: <keras.callbacks.History at 0x1532568e580>
```

Gambar 13. Fitting model

7. Prediksi Berdasarkan Model Latih

Membuat prediksi berdasarkan model latih, Langkah pertama yaitu melakukan konversi terlebih dahulu dari gambar *test* nya menjadi resolusinya yaitu 64×64, dikarenakan model yang ada di penelitian ini hanya bisa menerima dari pernyataan tersebut. Hasil yang dihasilkan dari fungsi *predict()* yaitu berbentuk *binary* nol. Dapat dilihat pada gambar 14 merupakan penentuan prediksi berdasarkan model latih.

```
In [12]: import numpy as np
         from keras.preprocessing import image

         test_image = image.load_img('trash5.jpg', target_size = (64, 64))
         test_image = image.img_to_array(test_image)
         test_image = np.expand_dims(test_image, axis = 0)
         result = classifier.predict(test_image)
         training_set.class_indices
         if result[0][0]==0:
             prediction = 'Metal'
             print(prediction)
         else:
             prediction = 'Sampah'
             print(prediction)
```

Sampah

Gambar 14. Prediksi berdasarkan model latih

8. *Import Anvil Server*

Setelah memperoleh kuncinya, tambahkan kode seperti pada gambar 15 ke *Notebook* Jupyter, untuk ini karena telah memiliki model untuk artikel sebelumnya, jadi hanya menambahkan ke sel terakhir dari proyek. Dapat dilihat pada gambar 15 merupakan proses import anvil server.

```
In [16]: import anvil.server

         anvil.server.connect("DAIS2VRD6PJJRAE2IG4L7WQW-N2VTGLSPK6TPDQPN")
```

Gambar 15. *Import anvil server*

9. *Hasil Score*

Penentuan *score* pada sampah metal dan sampah jenis lainnya. Dapat dilihat pada gambar 16 merupakan hasil dari *score* yaitu:

```
In [17]: import anvil.media

         import PIL
         from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array

         @anvil.server.callable
         def classify_image(file):
             with anvil.media.TempFile(file) as filename:
                 img = image.load_img(filename)

                 img = img.resize((64, 64), resample=PIL.Image.BICUBIC)
                 arr = img_to_array(img)
                 arr = np.expand_dims(arr, axis=0)
                 arr /= 255

                 score = classifier.predict(arr)
                 return ('Metal' if score < 0.5 else 'Sampah', float(score))
```

```
In [ ]: import anvil.server

         anvil.server.connect("DAIS2VRD6PJJRAE2IG4L7WQW-N2VTGLSPK6TPDQPN")

         @anvil.server.callable
         def say_hello(name):
             print("Hello from the uplink, %s!" % name)

         anvil.server.wait_forever()
```

Gambar 16. Hasil *score*

VI. UJI COBA

Hasil Uji Coba menghasilkan nilai akurasi yaitu jumlah *step* per *epoch* yaitu 50, sehingga belum menghasilkan akurasi yang tinggi atau belum menghasilkan akurasi yang terbaik. Selanjutnya terdapat bentuk masukan pada model CNN ini yaitu 512x384 piksel, nilai jumlah *epochs* yaitu 100, dengan waktu menyelesaikan *epochs* 5 detik, sehingga sudah menghasilkan nilai akurasi terbaik. serta *dataset* yang digunakan terdapat *citra* sampah metal dan juga sampah umum yang berjumlah 547 *citra* sehingga menghasilkan akurasi sebesar 96%. Dalam menggunakan *padding* dan *stride* menyebabkan waktu penyelesaian menjadi lebih cepat, apabila pengujian atau *testing* tanpa menggunakan *padding* dan *stride* menghasilkan waktu penyelesaian per *epochs* nya lebih dari 5 detik. Selanjutnya hasil Uji Coba menggunakan arsitektur jaringan *ResNet 50*, uji coba penelitian ini juga menggunakan optimasi adam menghasilkan nilai akurasi terbaik.

V. SIMPULAN

Pada penelitian ini menggunakan klasifikasi metode CNN dalam membangun klasifikasi pada sampah jenis daur ulang berupa *glass*, metal dan juga menggunakan *web app builder* pihak ketiga yaitu *anvil*. Dari penelitian yang telah dilakukan sebelum

Pada proses *training* atau pelatihan ini merupakan suatu proses guna untuk memperoleh model *machine learning* atau pembelajaran mesin. Pada saat melakukan *training* pada penelitian ini sebelumnya sudah diatur sedemikian rupa, pada jumlah *step* per *epoch* yaitu 100, waktu penyelesaian *epochs* adalah 5 detik. Dalam menggunakan *padding* dan *stride* menyebabkan waktu penyelesaian menjadi lebih cepat, apabila pengujian atau *testing* tanpa menggunakan *padding* dan *stride* menghasilkan waktu penyelesaian per *epochs* nya lebih dari 5 detik.

Dalam implementasinya menggunakan Bahasa pemrograman Python, *Anvil*, dan juga menggunakan *library TensorFlow* dan *Keras*. Sistem berhasil mendeteksi jenis sampah metal dari sampah umum yang diimplementasikan berbasis web dengan menggunakan *Anvil*. *Input shape* pada model CNN ini yaitu 512x384 piksel, jumlah *epochs* 100, serta *dataset* untuk *training* data pada sampah logam berjumlah 410, dan pada sampah umum berjumlah 137. Selanjutnya *dataset* untuk *testing* data pada sampah logam berjumlah 410, dan pada sampah umum berjumlah 137, sehingga perbandingan *dataset* untuk *training* dan *testing* adalah 50:50, karena proses paduan data yang dihasilkan dari sebuah gambar (*citra*) yang mirip dengan melakukan pengoperasian seperti proses *flipping*, rotasi, dan *blurring*, maka dari itu hasil latih dari penelitian ini menghasilkan akurasi yang tinggi. Pada penelitian ini menghasilkan akurasi sebesar 96%, karena diambil dari nilai akurasi yang paling tinggi yaitu pada *epochs* ke 98 dari 100.

UCAPAN TERIMA KASIH

Ucapan terima kasih dipersembahkan kepada seluruh pihak yang sudah membantu dan mendukung pada saat pengerjaan penelitian ini sehingga penelitian ini bisa berjalan lancar sebagai mestinya.

DAFTAR PUSTAKA

- [1] Arrofiqoh, E. N and Harintaka, H, "Implementasi Metode *Convolutional Neural Network* Untuk Klasifikasi Tanaman Pada *Citra* Resolusi Tinggi," *Geomatika*, 24 (2), p. 61.24-2.810, 2018, doi: 10.24895/jig.
- [2] T. A. M. Irfan Nugraha Pratama and Tatang Rohana, "Pengenalan Sampah Plastik Dengan Model *Convolutional Neural Network*," *Conf. Innov. Appl. Sci. Technol. (CIASTECH)*, no. *Ciastech*, pp. 691–698, 2020.
- [3] N. Fadlia and R. Kosasih, "Klasifikasi Jenis Kendaraan Menggunakan Metode *Convolutional Neural Network* (Cnn)," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 3, pp. 207–215, 2019, doi: 10.35760/tr.v24i3.2397.
- [4] Rima Dias Ramadhani, A. Nur Aziz Thohari, C. Kartiko, A. Junaidi, T. Ginanjar Laksana and N. Alim Setya Nugraha, "Optimasi Akurasi Metode *Convolutional Neural Network* untuk Identifikasi Jenis Sampah," *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 312–318, 2021, doi: 10.29207/resti.v5i2.2754.
- [5] Bambang Pulu Hartato, "Penerapan *Convolutional Neural Network* pada *Citra* Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [6] S. Juliansyah and A. D. Laksito, "Klasifikasi *Citra* Buah Pir Menggunakan *Convolutional Neural Networks*," *J. Telekomun. dan Komput.*, vol. 11, no. 1, p. 65, 2021, doi: 10.22441/incomtech.v11i1.10185.
- [7] Y. Y. N. Y. Alkautsar, C. N. Arbaatun, and F. N. Prawita, "Matrash : the Use of Machine Learning in the Waste Bank Based Iot Integrated With Smart," *e-proceeding Appl. Sci.*, vol. 6, no. 2, pp. 4068–4076, 2020.
- [8] Bambang Pulu Hartato, "Penerapan *Convolutional Neural Network* pada *Citra* Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [9] E. N. Arrofiqoh and H. Harintaka, "Implementasi Metode *Convolutional Neural Network* Untuk Klasifikasi Tanaman Pada *Citra* Resolusi Tinggi," *Geomatika*, vol. 24, no. 2, p. 61, 24-2.810, 2018, doi: 10.24895/jig.
- [10] M. R. D. Septian, A. A. A. Paliwang, M. Cahyanti and E. R. Swedia, "Penyakit Tanaman Apel Dari *Citra* Daun Dengan *Convolutional Neural Network*," *Sebatik*, vol. 24, no. 2, pp. 207–212, 2020, doi: 10.46984/sebatik.v24i2.1060.

- [11] S. R. Suartika E. P, I Wayan and Wijaya Arya Yudhi, "Klasifikasi Citra Menggunakan *Convolutional Neural Network* (Cnn) Pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, p. 76, [Online]. Available: repository.its.ac.id/48842, 2016, doi: 12962/j23373539.v5i1.15696.
- [12] H. Fonda, "Klasifikasi Batik Riau Dengan Menggunakan *Convolutional Neural Networks* (Cnn)," *J. Ilmu Komputer.*, vol. 9, no. 1, pp. 7–10, 2020, doi: 10.33060/jik/2020/vol9.iss1.144.
- [13] L. Penerbangan, A. Nasional, D. L. Penerbangan, A. Nasional, S. Pendeteksi and H. Spasial. "Klasifikasi Data Radar Menggunakan Algoritma *Convolutional Neural Network*," *DoubleClick : Journal of Computer and Information Technology* Klasifikasi Data Radar(Ihsan) | 115 *DoubleClick : Journal of Computer and Information Technology* E-ISSN : 2579-5317 116 | Klasifikasi Data Radar(Ihsan), vol. 4, no. 2, pp. 115–121, 2021, doi: 10.25273/doubleclick.v4i2.8188.