# Sistem Klasifikasi Limbah Menggunakan Metode *Convolutional Neural Network* (CNN) Pada *Web Service* Berbasis *Framework Flask*

Parole Nimadinaga Dacipta[1], Ricky Eka Putra[2]

[1,2]Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

[1]parole.17051204075@mhs.unesa.ac.id
[2]rickyeka@unesa.ac.id

***Abstrak*— Di berbagai negara sampah merupakan permasalahan yang dihadapi sehari-hari seluruh aktivitas tidak bisa terhindar dari kata sampah. banyak sekali jenis–jenis sampah di lingkungan kita, dan kebanyakan masyarakat Indonesia masih belum bisa membedakan jenis sampah apa yang setiap hari mereka buang sehingga banyak sekali sampah yang tidak diklasifikasikan. Yang menyebabkan proses klasifikasi limbah pada pabrik daur ulang sangat susah untuk dilakukan. Maka dari itu bagaimana jika limbah yang ada diklasifikasikan menjadi 9 macam dan setiap masyarakat dapat membuang limbah sampah mereka sesuai dengan 9 macam class yang telah dibedakan yaitu battery, pakaian, e-limbah, kaca, bola lampu, metal, organic, kertas, plastic. sehingga mempermudah pabrik daur ulang dalam proses klasifikasi sampah. Oleh karena itu Sistem Klasifikasi Limbah Pada *Web Service* Berbasis *Framework Flask* akan sangat membantu masyarakat dalam mengklasifikasikan limbah sampah yang dihasilkan. Sistem klasifikasi limbah pada aplikasi berbasis web ini menggunakan metode *Convolutional Neural Network* (CNN). Metode CNN merupakan suatu metode *deep learning* yang dapat mengidentifikasi dan mengklasifikasi sebuah *object* pada citra digital. Penelitian ini menggunakan arsitektur *VGGNet*. *VGGNet* merupakan arsitektur dari *Convolutional Neural Network*.**

**Penelitian ini menggunakan 9 class limbah sampah. Dataset yang digunakan yaitu 8371 citra limbah sampah. Dimana dataset tersebut digunakan untuk melakukan *training* data yang telah dibagi menjadi 1.122 citra battery, 729 pakaian, 624 e-limbah, 773 kaca, 651 bola lampu, 1092 metal, 671 organic, 1468 kertas dan 1241 plastic. Pada proses training dilakukan sebanyak 28 epoch, yang mendapatkan akurasi tertinggi 69,77% dengan loss terendah 0,34. Untuk data testing didapatkan hasil 64,45% accuracy.**

*Kata Kunci*—**Sistem Klasifikasi, Limbah, *Convolutional Neural Network, Framework Flask***

## I. PENDAHULUAN

Pada saat ini di berbagai negara mengalami beberapa permasalahan, salah satunya yaitu masalah sampah. Sampah merupakan permasalahan yang sering kita dihadapi sehari - hari, apapun yang kita lakukan, tidak pernah luput dari kata sampah. Sampah ialah salah satu bahan yang sudah tidak lagi terpakai atau bahan yang akan dibuang. Sampah merupakan hasil aktivitas manusia ataupun alam yang sudah tidak digunakan lagi, walaupun begitu tetapi ada juga sampah yang dapat di daur ulang. Sampah sampah tersebut terbagi menjadi 9 kelompok yang merupakan inisiatif dari penulis, yaitu battery, pakaian, e-limbah, kaca, bola lampu, metal, organic, kertas, plastic. Namun, jika sudah tidak digunakan lagi maka akan menjadi sampah yang tidak dapat hilang begitu saja karena tidak dapat terurai oleh bakteri pengurai begitu saja sehingga dapat menyebabkan penumpukan sampah yang akan membutuhkan banyak waktu untuk dapat diuraikan. Produk plastik dunia sudah menyentuh angka 8 juta metrik ton pada tahun 2015 dan diperkirakan jumlahnya akan meningkat hingga 53-90 juta ton pada tahun 2030. Jumlah sampah yang ada didunia akan meningkat sebanyak 1,3 miliar ton pada tahun 2040 mendatang jadi tidak terpaku pada sampah plastik saja, walaupun paling banyak produksi sampah berasal dari jenis plastik yaitu plastik kemasan produk sebanyak 146 ton per tahun, disusul dengan sampah dibidang pembangunan dan juga konstruksi yang mencapai sebanyak 65 juta ton. Sampah di Indonesia sendiri jumlah sampah yang dicatatkan Indonesia mencapai sekitar 4,8 juta ton per tahun untuk sampah plastik, sampah di Indonesia tidak terkelola dengan baik seperti membakar sampah pada ruang terbuka mencapai 48%, yang tidak dikelola di tempat pembuangan sampah resmi dengan layak sebanyak 13% dan sisanya telah mencemari saluran air dan laut sebanyak 9%. Sampah di negara Indonesia diprediksi akan naik drastis mengingat jumlah produksi sampah plastik di Indonesia menunjukkan tren meningkat 5% tiap tahunnya. Melihat kenyataan tersebut, peningkatan peran pekerja sektor informal dalam ekosistem pengelolaan sampah nasional sangat penting[1]. Oleh karena itu penyelesaian masalah plastik di negara Indonesia mau tidak mau harus melibatkan sektor informal untuk memaksimalkan kontribusi pekerja informal terhadap pengurangan polusi plastik.

Bahasa Python dikembangkan oleh Guido van Rossum. Ada cerita menarik di baliknya memberi nama "Python" ke pemrograman bahasa. Pada saat melakukan pengembangan python, pengembang sedang membaca skrip "Monty's Python Terbang yang merupakan serial dari BBC. Saat membaca buku ini ia pun mendapatkan ide untuk memberi nama pemrograman bahasa sebagai "Python" memiliki nama yang singkat dan juga unik . Python yang berorientasi objek, lalu ditafsirkan, dan menjadi bahasa pemrograman *interaktif*. Ini menyediakan struktur data tingkat tinggi seperti daftar, tupel, set, *array* asosiatif (disebut juga kamus), dinamis mengetik dan mengikat, modul, kelas, pengecualian, manajemen memori otomatis, dll [2].

Beberapa tahun terakhir ini, *Artificial Intelligence* atau bisa disebut juga (kecerdasan buatan) telah mengalami perkembangan yang sangat pesat. Kecerdasan Buatan hampir diterapkan di semua aspek kehidupan sehari-hari karena dapat memecahkan masalah yang kompleks, seperti yang telah disebutkan dalam *AI*. Popularitas pembelajaran pada mesin mengalami peningkatan mengikuti popularitas Jaringan Syaraf Tiruan (JST). Saraf *Konvolusi Network* (CNN) merupakan algoritma *deep learning* yang populer dalam melakukan pengolahan citra. Umumnya digunakan untuk melakukan pengenalan objek pada citra. Model yang digunakan dalam klasifikasi penyakit padi tergantung pada Jaringan Konvolusi (CNN). CNN nantinya akan memproses data gambar, yang akan dibangun oleh jaringan menggunakan informasi dari proses yang akan dieksekusi [3]. Oleh karena itu, arsitektur LeNet mungkin juga tidak pantas dalam kasus-kasus tertentu yang menghasilkan kinerja yang buruk dari model. Model ini juga rentan terhadap overfitting, yaitu: model mampu bekerja dengan baik pada data latih tetapi tidak untuk data uji [4]. Ada beberapa metode dalam penanganan overfitting untuk menghasilkan lebih banyak kinerja model yang optimal. Oleh karena itu, penelitian ini bertujuan untuk mengetahui model klasifikasi yang terbaik antara gambar daging babi dan juga gambar sapi dengan membandingkan beberapa metode regularisasi untuk mengatasi gejala *overfitting* [5]. Parameter dasar terkait data dan juga melatih komputer agar dapat belajar sendiri dengan mengenali pola menggunakan banyak lapisan pemrosesan, parameter ini otomatis telah disiapkan oleh Deep Learning. *Deep learning* juga merupakan salah satu dasar dari *Artificial Intelligence*. Semakin meningkatnya popularitas *Artificial Intelligence* di dunia teknologi, akhirnya algoritma deep learning pun mulai banyak dipelajari. Dengan menggunakan algoritma. Nantinya seorang developer bisa meningkatkan kemampuan tools untuk mengenali, mengklasifikasikan, mendeteksi, memahami, dan juga melakukan pendeskripsian data karena menggunakan algoritma Deep Learning [6].

Manusia merupakan makhluk hidup yang memiliki keterampilan intrinsik untuk melakukan *transfer* pengetahuan dari seluruh aktivitas yang berbeda-beda. Pengetahuan yang telah kita peroleh saat melakukan suatu aktivitas tertentu, nantinya akan beroperasi di tempat yang sama untuk menyelesaikan kegiatan terkait. "*Transfer*" adalah *kognitif* praktik di mana penguasaan pengetahuan atau keterampilan pembelajaran dalam suatu konteks akan memungkinkan mereka dalam menerapkan pengetahuan atau keterampilan itu dalam konteks yang tentunya berbeda. Pembelajaran mesin dan pembelajaran mendalam algoritma telah dirancang secara *konvensional* bekerja untuk melakukan pendistribusian fitur-ruang tertentu. Setelah fitur-ruang perubahan selesai didistribusikan, perlu mendesain ulang model dari awal, dan itu juga tugas yang rumit untuk mengumpulkan pelatihan yang diperlukan data. Akibatnya, karena model pembelajaran yang mendalam selama pelatihan membutuhkan data berlabel yang cukup [7].

Banyak algoritma yang berbeda telah dikembangkan untuk melakukan klasifikasi gambar, seperti memiliki RNN, SVM, ANN dll, tetapi pada Convolutional Neural Network yang merupakan algoritma Machine Learning benar-benar bekerja lebih baik dari semuanya. CNN berhasil mendapat tempat ketika algoritme digunakan untuk memenangkan tantangan pengenalan *visual* skala besar *image-Net* 2012 yang diusulkan dalam (ILSVRC). Sejak 2012 banyak arsitektur CNN yang berbeda telah dikembangkan dengan memecahkan banyak klasifikasi pada masalah gamba. Universitas teknologi Lulea pada tahun 1999 melakukan sebuah penelitian, dan sebuah sistem dikembangkan untuk mendaur ulang logam memo menggunakan pengenal berbentuk mekanis menggunakan fitur dari SIFT dan bentuk garis besar pada komputasi Bayesian framework dan sistem mereka didasarkan pada database material Flickr, pada tahun 2016 mengembangkan Auto-Trash yang mampu untuk membedakan antara kompos dan didaur ulang dengan menggunakan Raspberry Pi, sistem mereka dikembangkan menggunakan teknologi Tensorflow Google. Untuk kekurangan dari sistem mereka adalah hanya mampu membedakan bahan dar kompos. Sebuah aplikasi smartphone telah dikembangkan secara kasar dapat mengidentifikasi bagian bawah sampah dalam gambar. Aplikasi ini memungkinkan seseorang untuk memberikan informasi sampah di daerah mereka dan memperoleh akurasi rata-rata 85% menggunakan model pra-pelatihan AlexNet [8].

*Deep Learning* memiliki sifat dinamis. Algoritma *Deep Learning* ini memiliki kemampuan untuk terus meningkatkan kinerjanya dan mudah dalam beradaptasi dengan perubahan pada pola. Deep Learning juga memiliki fungsi *pada* dunia bisnis adalah untuk melakukan analisis personalisasi pelanggan, meningkatkan keakuratan dan kinerja aplikasi dan klasifikasi gambar. Contohnya seperti aplikasi yang peneliti buat dalam mengklasifikasi sampah yang sangat berguna bagi dunia industri dan perseorangan. Salah satu contoh teknologi yang telah mengadopsi *deep learning* adalah *siri* dan *cortana.*

*Framework* merupakan kerangka kerja dalam melakukan pengembangan aplikasi berbasis website maupun desktop. Pembentukan kerangka kerja sangat membantu para developer dalam menuliskan sebuah *template* dengan lebih terstruktur dan juga tersusun dengan rapi. Penelitian ini menggunakan Framework berjenis Flask[9]. Flask merupakan framework aplikasi web yang ringan dengan ditulis dalam bahasa python dan baseband pada toolkit WSGI dan mesin template jinja2. Flask mengambil python fleksibel bahasa pemrograman dan menyediakan template sederhana untuk pengembangan web. Flask akan dapat digunakan untuk menghemat waktu dalam melakukan pembangunan aplikasi web jika selesai mengimpor ke python. Ini berfungsi untuk menjaga inti sederhana namun dapat diperluas. Tidak memiliki abstraksi data dasar lapisan, validasi formulir, atau komponen lainnya. Labu mendukung ekstensi. Ekstensi ada untuk objek-relasional pembuat peta, validasi formulir, penanganan unggahan, berbagai buka teknologi otentikasi dan banyak lagi [10].

*VGGNet*, merupakan salah satu arsitektur convolutional neural network (*CNN*) yang dapat digunakan untuk mengatasi

kebutuhan akan computing resource berlebih. Para peneliti membuat arsitektur VGGNet dan Arsitektur CNN pada umumnya merupakan lapisan ataupun konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari melakukan input image, karena sesuai dengan namanya yaitu "Mobile".. Arsitektur VGGNet atau bisa disebut juga VGG16 dikembangkan dan diperkenalkan oleh Karen Simonyan dan Andrew Zisserman dari Universitas Oxford, pada tahun 2014, melalui artikel mereka yang berjudul "Very Deep Convolutional Networks for Large-Scale Image Recognition". Model VGGNet mencapai 92,7% tingkat keakuratan pengujian top-5 di *ImageNet*, yang merupakan kumpulan data lebih dari 14 juta gambar yang termasuk dalam 1000 kelas. Itu merupakan salah satu model terkenal yang diajukan pada *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) pada tahun 2014. Nantinya, *VGGNet* akan membuat perbaikan pada arsitektur *AlexNet* dengan mengganti filter berukuran kernel besar (masing-masing 11 dan 5 di lapisan konvolusi pertama dan kedua) dengan beberapa filter berukuran tiga × tiga kernel satu demi satu [11].

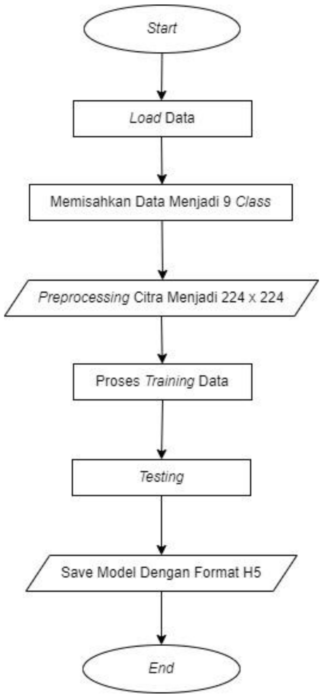
Berdasarkan penelitian terdahulu yang dilakukan oleh Adigin dan Wang [8], penulis ingin melakukan penelitian mengenai "Klasifikasi limbah menggunakan metode cnn pada *web service* berbasis Framework Flask". Penelitian ini mengimplementasikan *deep learning* dengan metode *CNN* dengan arsitektur *VGGnet* dan back end python pada *web service* berbasis *Framework Flask* sehingga diharapkan dapat melakukan pengembangan aplikasi web yang lebih efisien, fleksibel dan mempunyai ketersediaan yang tinggi (*high availability*). Kegunaan aplikasi sebagai media agar masyarakat dapat membedakan jenis sampah yang mereka miliki sehingga sampah – sampah tersebut dapat diklasifikasikan dan mempermudah proses klasifikasi pada pabrik daur ulang.

## II. Metodologi penelitian

Metodologi penelitian merupakan pedoman dalam melaksanakan penelitian. Adanya metodologi penelitian, alur dan hasil penelitian dapat tercapai, serta tidak menyimpang dari tujuan yang telah ditentukan. Penelitian ini merupakan penerapan *deep learning* metode *Convolutional Neural Network* (CNN). Perancangan arsitektur CNN dilakukan menggunakan *platform google colaboratory* dengan bahasa pemograman python. Selain itu, dataset yang digunakan merupakan citra limbah sampah yang akan diproses dengan hasil yang menunjukkan termasuk jenis limbah sampah apakah yang telah diklasifikasikan. Langkah-langkah yang dilakukan dalam proses pembuatan sistem identifikasi CNN dapat dilihat pada Gbr. 1.

### *A. Dataset*

Penelitian ini menggunakan dua dataset limbah sampah dari sumber yang berbeda yang dijadikan satu. Dua sumber berbeda tersebut didapatkan dari *website Kaggle* dan *Google Drive*. Sumber pertama [12] dan sumber kedua [13] digabung, diambil 9 class limbah sampah untuk dijadikan sebagai dataset. Dataset akan diunggah ke *drive* yang berfungsi sebagai penyimpanan pada *google collaboratory*.




Gbr 1. Sistem Klasifikasi *CNN*

### *B. Analisa Kebutuhan*

a. Kebutuhan Perangkat Lunak
   1. *Google Collaboratory*
   2. Anaconda
   3. CMD
   4. Visual Studio
   5. Spyder
b. Kebutuhan Perangkat Keras
   1. Processor Ryzen 5
   2. HDD 1 TB
   3. RAM 12 GB
   4. Sistem Operasi Windows 11 64-bit
c. Kebutuhan Library
   1. Keras dan Tensorflow
   2. Flask
   3. VGG16
   4. Matplotlib

### *C. Preprocessing*

*Preprocessing* pada penelitian ini menyiapkan dataset 9 class yang telah diunggah ke dalam *google drive*. Sebelum Proses Preprocessing pengambilan data atau load data dilakukan pada website Kaggle dan Google Drive setelah load data selesai kemudian data yang sudah kita load dimasukkan kedalam google drive untuk memisahkan menjadi 9 class yang berbeda kemudian dilanjutkan ketahap preprocessing, Karena

*dataset* pada penelitian ini menggunakan dua sumber yang berbeda, maka citra limbah sampah memiliki ukuran *pixel* yang berbeda-beda. Untuk menyamakan ukuran *pixel* perlu untuk melakukan *resize image*. Dataset 9 class tersebut, merupakan data mentah yang siap untuk proses *resize image* ukuran 224 x 224 *pixel*. *Resize image* pada *preprocessing* mempermudah ke proses *training* dan membantu proses *training* mendapatkan tingkat akurasi semaksimal mungkin.

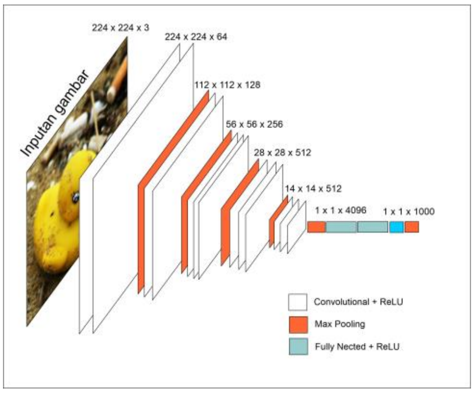
### *D. Arsitektur Convolutional Neural Network*

*Convolutional Neural Network* (CNN) adalah suatu jenis *Metode* yang difungsikan untuk mengolah data citra. Salah satu arsitketur *Convolutional Neural Network* (CNN) adalah VGG16 seperti dalam penelitian ini. Dalam VGG16 kumpulan *image* akan dipecah menjadi beberapa *layer* pada proses *training*. Tabel I dan Gbr 2 menggambarkan model CNN dengan inputan gambar menggunakan 224 x 224 x 3 *channel* RGB. Kemudian dipecah menjadi 2 bagian menggunakan Convolutional + Relu 224 x 224 pixel dan *filter* 64 untuk mengkonvolusi *layer* pertama dengan *ReLU activation*. Hal tersebut bertujuan untuk mengambil *node* tertinggi yang akan dilanjutkan pada konvolusi berikutnya dan mendapatkan parameter 1792 pada layer pertama dan parameter kedua 36.928. Setelah konvolusi 1 selesai, maka didapatkan hasil *output shape* pada max pooling yaitu 112 x 112 pixel dan filter 128. lalu dilanjutkan dengan memecah layer lagi menjadi 2 menggunakan Convolution + Relu dengan layer yang sama sesuai dengan max pooling yang didapatkan yaitu 112 x 112 pixel dan filter 128, dan didapatkan parameter 73.856 pada layer ketiga dan 147.584 parameter pada layer keempat. Setelah konvolusi 2 selesai, maka didapatkan hasil *output shape* dengan max pooling 56 x 56 pixel dan filter 256. Lalu dilanjutkan dengan memecah layer menjadi 3 menggunakan Convolutional + Relu menjadi 56 x 56 pixel dan filter 256 didapatkan parameter 295.168 pada layer kelima, 590.080 parameter pada layer keenam, dan 590.080 pada layer ketujuh. Setelah konvolusi ke 3 selesai, maka didapatkan hasil *output shape* pada max pooling yaitu 28 x 28 pixel dan filter 512. Lalu dilanjutkan dengan dengan memecah layer menjadi 3 menggunakan Convolutional + Relu menjadi 28 x 28 pixel dan filter 512 didapatkan parameter 1.180.160 pada layer kedelapan, 2.359.808 parameter pada layer kesembilan, dan 2.359.808 parameter pada layer kespuluh. Setelah konvolusi ke 4 selesai, maka didapatkan *output shape* pada max pooling 14 x 14 pixel dan filter 512. Lalu dilanjutkan dengan memecah layer lagi menjadi 3 menggunakan Convolutional + Relu menjadi 14 x 14 pixel dan filter 512 didapatkan parameter sebesar 2.359.808 pada layer kesebelas, 2.359.808 parameter pada layer kedua belas, dan 2.359.808 parameter pada layer ketiga belas, maka didapatkan *output shape* pada max pooling 7 x 7 pixel dan filter 512. Lalu dilanjutkan pada tahap terakhir fully reacted + Relu dengan hasil flatten 25.088. hasil flatten seluruhnya dimasukkan kedalam dense dengan total 9 class klasifikasi. Sehingga total keseluruhan 225.801 parameter yang ditrain dari total 14.940.489 parameter yang dimiliki oleh arsitektur VGG16 ini adalah hasil dari running data sendiri.

TABEL I
ARSITEKTUR *CONVOLUTIONAL NEURAL NETWORK VGG16*

| ***Layer*** | ***Type*** | ***Output Shape*** | ***Parameter*** |
|---|---|---|---|
| input_1 (InputLayer) | input_1 (InputLayer) | (*None*, 224, 224, 3) | 0 |
| block1_conv1 (Conv2D) | block1_conv1 (Conv2D) | (*None*, 224, 224, 64) | 1792 |
| block1_conv2 (Conv2D) | block1_conv2 (Conv2D) | (*None*, 224, 224, 64) | 36.928 |
| block1_pool (MaxPooling2D) | block1_pool (MaxPooling2D) | (*None*, 112, 112, 64) | 0 |
| block2_conv1 (Conv2D) | block2_conv1 (Conv2D) | (*None*, 112, 112, 128) | 73.856 |
| block2_conv2 (Conv2D) | block2_conv2 (Conv2D) | (*None*, 112, 112, 128) | 147.584 |
| block2_pool (MaxPooling2D) | block2_pool (MaxPooling2D) | (*None*, 56, 56, 128) | 0 |
| block3_conv1 (Conv2D) | block3_conv1 (Conv2D) | (*None*, 56, 56, 256) | 295.168 |
| block3_conv2 (Conv2D) | block3_conv2 (Conv2D) | (*None*, 56, 56, 256) | 590.080 |
| block3_conv3 (Conv2D) | block3_conv3 (Conv2D) | (*None*, 56, 56, 256) | 590.080 |
| block3_pool (MaxPooling2D) | block3_pool (MaxPooling2D) | (*None*, 28, 28, 256) | 0 |
| block4_conv1 (Conv2D) | block4_conv1 (Conv2D) | (*None*, 28, 28, 512) | 1.180.160 |
| block4_conv2 (Conv2D) | block4_conv2 (Conv2D) | (*None*, 28, 28, 512) | 2.359.808 |
| block4_conv3 (Conv2D) | block4_conv3 (Conv2D) | (*None*, 28, 28, 512) | 2.359.808 |
| block4_pool (MaxPooling2D) | block4_pool (MaxPooling2D) | (*None*, 14, 14, 512) | 0 |
| block5_conv1 (Conv2D) | block5_conv1 (Conv2D) | (*None*, 14, 14, 512) | 2.359.808 |
| block5_conv2 (Conv2D) | block5_conv2 (Conv2D) | (*None*, 14, 14, 512) | 2.359.808 |
| block5_conv3 (Conv2D) | block5_conv3 (Conv2D) | (*None*, 14, 14, 512) | 2.359.808 |
| block5_pool (MaxPooling2D) | block5_pool (MaxPooling2D) | (*None*, 7, 7, 512) | 0 |
| flatten (Flatten) | flatten (Flatten) | (*None*, 25088) | 0 |
| dense (Dense) | dense (Dense) | (*None*, 9) | 225801 |
| Total parameter | | | 14.940.489 |
| *Trainable* parameter | | | 225.801 |
| *No-trainable* parameter | | | 14.714.688 |

Pada Gbr 2. Diperlihatkan bagaimana proses dari arsitektur CNN mulai dari step pertama penginputan gambar sampai tahap akhir proses data.




Gbr 2. Gambaran Arsitektur CNN VGG16

## E. *Training*

*Training* bertujuan untuk mengenalkan model CNN yang telah dirancang sehingga dapat mengenali dan membedakan gambar limbah sampah yang telah diklasifikasikan sebelumnya. Jumlah data *training* terdiri dari 8371 citra limbah sampah. Proses *Training* pada pelatihan ini dilakukan sebanyak 28 *epoch.*

## F. *K-Fold Cross Validasi*

Validasi berisi data beserta label yang sama dengan *training*. Proses validasi dilakukan untuk menguji keakuratan hasil *training* sehingga data yang kita tes dapat masuk kedalam 9 class yang sesuai dengan limbah tersebut. validasi dapat meminimalisir *overfitting*. *Overfitting* merupakan keadaan dimana model yang digunakan terlalu fokus. Fokus yang berlebihan biasanya terjadi karena model menangkap data noise yang seharusnya diabaikan. Akibat dari *overfitting* dapat menyebabkan penurunan akurasi model.

```
[1 2 3 4 5 6 7 8] [0]
[0 2 3 4 5 6 7 8] [1]
[0 1 3 4 5 6 7 8] [2]
[0 1 2 4 5 6 7 8] [3]
[0 1 2 3 5 6 7 8] [4]
[0 1 2 3 4 6 7 8] [5]
[0 1 2 3 4 5 7 8] [6]
[0 1 2 3 4 5 6 8] [7]
[0 1 2 3 4 5 6 7] [8]
```

Gbr 3. K-Fold Cross

Gambar diatas merupakan K-Fold Cross dari proses validasi untuk meminimalisir overfitting , sehingga akan dicari pada tahap akhir bahwa gambar x masuk dalam class “0” tidak termasuk dalam 1 – 8 class yang sudah ditentukan lainnya, begitu juga jika x termasuk dalam class “1” maka x tidak akan terdeteksi class “0”, “2”, “3”, “4”, “5”, “6”, “7”, “8” begitu seterusnya hingga 9 class limbah sampah yang telah dibuat sebelumnya.

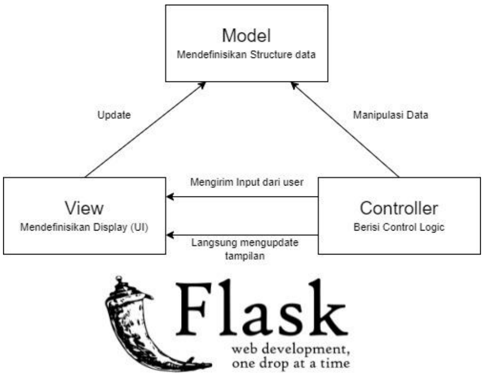
## G. *Testing*

Data *testing* atau juga disebut *testing set*, wajib dijalankan setelah proses validasi. Hal tersebut berfungsi untuk membuktikan keakuratan model CNN pada penelitian ini. Data yang telah dicoba untuk *testing* pada penelitian ini sebanyak 1 citra limbah sampah.

Gbr 4. Hasil Testing

## H. *Arsitektur Model View Controller*

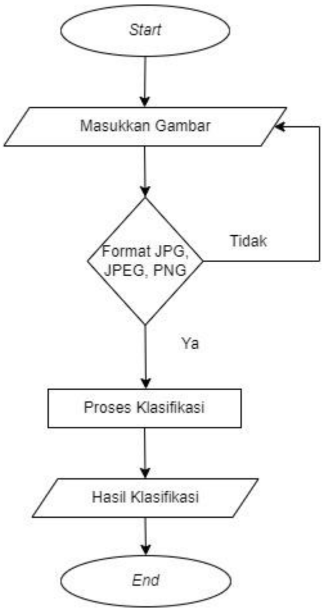
Penelitian ini menggunakan Model View Controller, dimana Model View Controller sebuah arsitektur untuk membuat aplikasi yang dipisah menjadi 3 bagian yaitu model, view, dan controller. Model berfungsi sebagai pengidentifikasian structure data, View sebagai tempat untuk menaruh desain atau User Interface (UI) dari aplikasi, dan Controller sebagai logic dari sebuah aplikasi atau biasa disebut back end dari suatu aplikasi seperti yang digunakan dalam penelitian ini.




Gbr 5. Arsitektur Model View Controller

## I. *Alur Proses Klasifikasi Data Citra Pada WebService*

Penelitian ini menggunakan Model View Controller, dimana Model View Controller sebuah arsitektur untuk membuat aplikasi yang dipisah menjadi 3 bagian yaitu model, view, dan controller




Gbr 6. Alur Klasifikasi *Citra* Pada *WebService*

## III. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 9 kelas limbah sampah untuk memenuhi kebutuhan data maka diambil data public dalam proses identifikasi. Pada tiap kelas limbah sampah diambil 1122 citra battery, 729 citra pakaian, 624 citra E-Limbah, 773 citra kaca, 651 citra bola lampu, 1092 citra metal, 671 citra organic, 1468 citra kertas, 1241 citra plastic untuk data *training*, dan 2 citra untuk data *testi*ng. Data limbah yang diambil terdapat pada Tabel II.

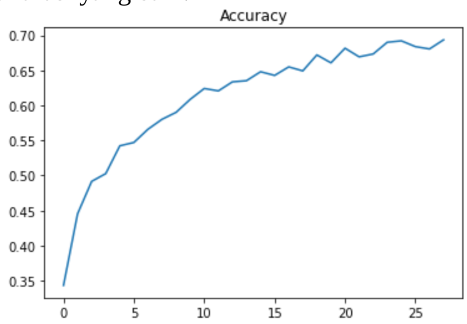
TABEL II
DATASET LIMBAH SAMPAH

| Jenis Limbah | *Training* |
|---|---|
| **Battery** | 1122 |
| **Pakaian** | 729 |
| **E-Limbah** | 624 |
| **Kaca** | 773 |
| **Bola Lampu** | 651 |
| **Metal** | 1092 |
| ***Organic*** | 671 |
| **Kertas** | 1468 |
| **Plastic** | 1241 |

### A. *Training Data Pada Convolutional Neural Network*

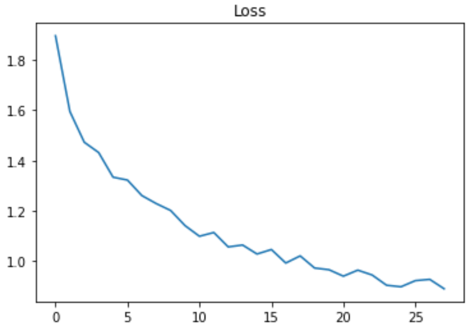
Jumlah data *training* sebanyak 8371 citra dengan menggunakan model CNN. Model CNN dijalankan untuk mencari hasil yang terbaik atau untuk mendapatkan hasil dari target yang telah ditentukan sebelumnya.

Akurasi yang didapatkan dari model CNN dalam proses *training* dapat ditunjukkan pada Gbr. 7. Berdasarkan hasil gambar, tingkat akurasi pada saat *training* dari 28 *epoch* didapatkan hasil akurasi tertinggi sebesar 69,77%. Pemilihan jumlah *epoch* berdasarkan pada beberapa percobaan yang bertujuan untuk menghindari *overfitting* dan *underfitting* agar mencapai akurasi yang baik.




Gbr 7. Hasil Akurasi

Gbr. 8 menunjukkan bahwa nilai *loss* yang didapat adalah 0,28. Hasil tersebut dipengaruhi oleh banyaknya data yang digunakan dan jumlah *epoch* pada saat *training* dan validasi. Semakin kecil nilai *loss* yang didapatkan, semakin tinggi nilai akurasinya.




Gbr 8. Hasil Loss

### B. *Hasil Perhitungan Akurasi*

Perhitungan akurasi dilakukan untuk menilai keberhasilan model CNN pada penelitian ini. Dengan adanya perhitungan akurasi dapat digunakan sebagai tolok ukur dalam perbandingan ataupun pengembangan model CNN yang akan dibuat pada masa yang akan datang. Dapat dilihat pada Tabel III menampilkan hasil *accuracy*. Pada penelitian ini didapatkan akurasi sebesar 0.6977 atau 69,77 %.

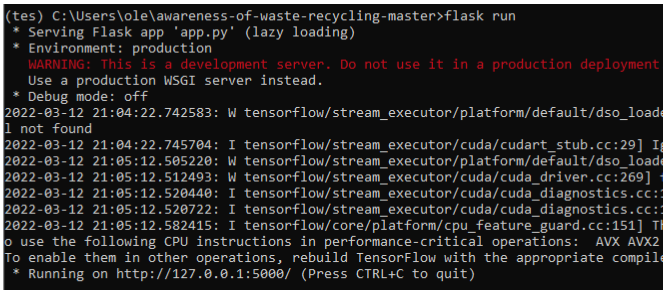
TABEL III
HASIL PERHITUNGAN AKURASI DENGAN 28 EPOCH

| Epoch | *Accuracy* |
|---|---|
| **Epoch 1** | 0.2906 |
| **Epoch 2** | 0.4493 |
| **Epoch 3** | 0.4902 |
| **Epoch 4** | 0.5115 |
| **Epoch 5** | 0.5421 |
| **Epoch 6** | 0.5446 |
| **Epoch 7** | 0.5528 |
| **Epoch 8** | 0.5857 |
| **Epoch 9** | 0.5932 |
| **Epoch 10** | 0.6118 |
| **Epoch 11** | 0.6285 |
| **Epoch 12** | 0.6304 |
| **Epoch 13** | 0.6471 |

| Epoch | *Accuracy* |
|---|---|
| Epoch 14 | 0.6420 |
| Epoch 15 | 0.6424 |
| Epoch 16 | 0.6397 |
| Epoch 17 | 0.6596 |
| Epoch 18 | 0.6607 |
| Epoch 19 | 0.6801 |
| Epoch 20 | 0.6612 |
| Epoch 21 | 0.6937 |
| Epoch 22 | 0.6714 |
| Epoch 23 | 0.6857 |
| Epoch 24 | 0.6953 |
| Epoch 25 | 0.6996 |
| Epoch 26 | 0.6858 |
| Epoch 27 | 0.6763 |
| Epoch 28 | 0.6977 |

## C. *Sistem Klasifikasi Limbah Berbasis WebService*

Sistem ini Menggunakan Backend Python Dengan Framework flask yang dapat dilihat pada Gbr 9. Yang memperlihatkan dimana proses run untuk menjalankan aplikasi flask

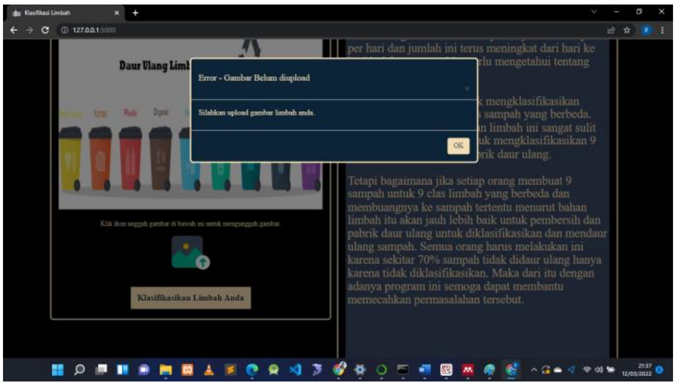



Gbr 9. Running Flask

Perintah "flask run" digunakan untuk menjalankan aplikasi berframework flask yang telah ada dikomputer penulis, sehingga didapatkan server internal 127.0.0:5000 yang akan digunakan sebagai internal server aplikasinya nanti.

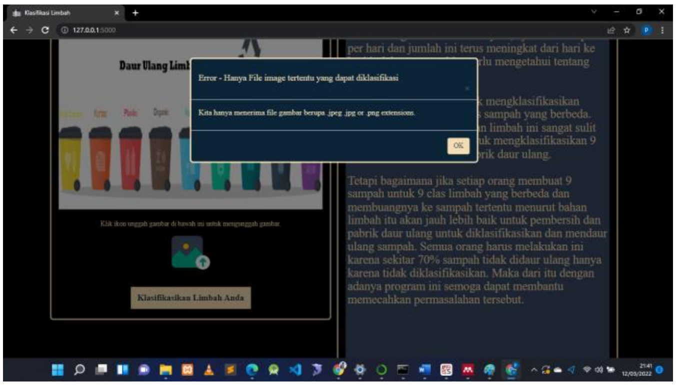
Gbr 10. Tampilan Sistem Klasifikasi Limbah Berbasis WebService

Lalu penulis membuka alamat 127.0.0:5000 pada web browser untuk membuka aplikasi flask yang telah penulis jalankan tadi sehingga muncul tampilan seperti Gbr 10.



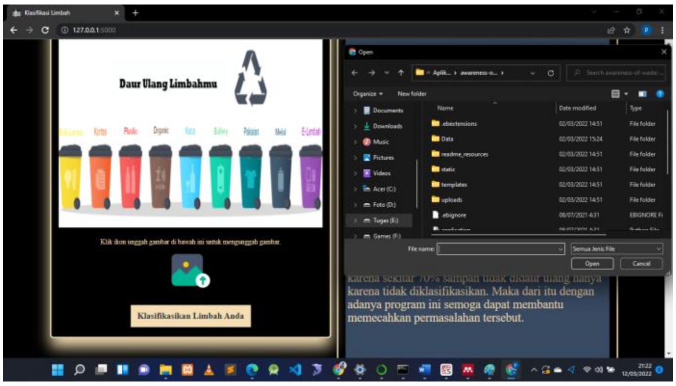
Gbr 11. Alert Ketika Belum Memilih Gambar

Ketika user belum memilih gambar maka akan muncul alert berupa "silakan upload gambar limbah anda" yang bertujuan untuk mengingatkan user bahwa user belum memilih gambar untuk proses klasifikasi seperti pada Gbr 11.



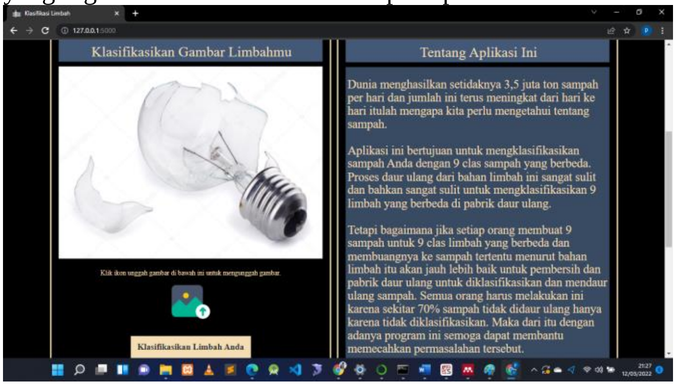
Gbr 12. Alert Ketika Format Gambar Tidak Sesuai

Sistem juga menyediakan format yang dapat diakses untuk proses klasifikasi yaitu format Jpg, Jpeg, Png. Ketika user tidak sesuai memasukkan format yang ada pada sistem maka akan muncul alert berupa "Kita hanya menerima file gambar berupa .jpeg .jpg or .png extensions" sehingga user dapat Kembali memilih gambar yang ingin diklasifikasikan, seperti yang terlihat pada Gbr 12.



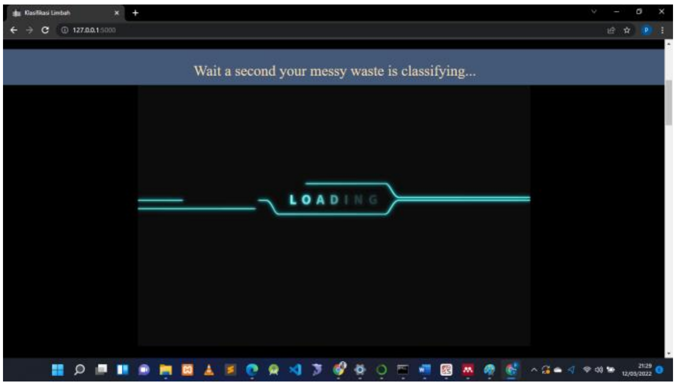
Gbr 13. Pemilihan Gambar Limbah

Lalu user dapat memilih gambar dengan menekan tombol yang telah disediakan untuk memilih gambar limbah sampah yang ingin diklasifikasi oleh user seperti pada Gbr. 13.



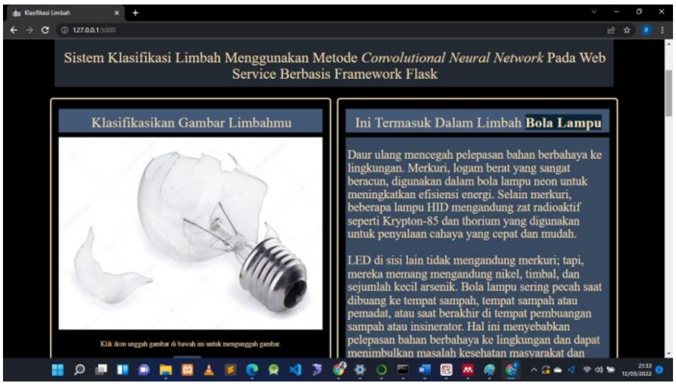

Gbr 14. Gambar Yang Telah Dipilih

Setelah user memilih gambar yang diinginkan maka gambar tersebut akan muncul pada tampilan website dan user dapat segera melanjutkan proses klasifikasi dengan menekan tombol “Klasifikasikan Limbah Anda” seperti pada Gbr 14.




Gbr 15. Tampilan Proses Loading Klasifikasi

Lalu proses klasifikasi oleh system sedang dilakukan dan tampilan Gbr 15 merupakan tampilan loading saat sistem sedang melakukan proses klasifikasi limbah.




Gbr 16. Hasil Proses Klasifikasi

Setelah Proses Loading selesai maka akan muncul hasil dari proses klasifikasi yang telah dilakukan sistem dan terdapat penjelasan yang sesuai dengan hasil dari klasifikasi yang didapatkan yaitu termasuk dalam limbah bola lampu seperti terlihat pada Gbr 16.

## IV. KESIMPULAN

Penelitian ini dilakukan dengan menggunakan metode CNN sebagai algoritma dari *deep learning* yang didesain untuk mengolah data dua dimensi, telah berhasil mengidentifikasi citra limbah sampah. Melalui proses *training* sebanyak 28 *epoch* didapatkan hasil akurasi tertinggi sebesar 69,77% dengan *loss* terendah 0,34. Sedangkan, hasil akurasi yang didapatkan melalui proses *testing* sebesar 64,45% dengan 8371 citra limbah sampah dapat klasifikasi. Setelah melalui beberapa proses, dapat disimpulkan bahwa algoritma CNN dipengaruhi oleh data *training* yang besar, semakin besar data *training* maka semakin tinggi juga akurasi yang didapatkan berdasarkan penelitian2 terdahulu dan juga jumlah epoch yang digunakan dalam arsitektur CNN mempengaruhi hasil akurasi, semakin banyak epoch maka akan semakin tinggi juga akurasi yang didapatkan.

## V. SARAN

Untuk mengoptimalkan kinerja CNN dapat menggunakan epoch diatas 50 agar arsitektur CNN dapat mentrain data klasifikasi lebih akurat lagi sehingga akan dihasilkan data loss yang lebih kecil dan akurasi yang lebih tinggi dan dapat menambahkan data training yang lebih banyak agar kinerja dari CNN bisa lebih maksimal. selain itu untuk mengoptimalkan program agar digunakan sebagai alat bantu klasifikasi limbah secara masal dapat menggunakan arsitektur berbasis backend servise berbasis RestApi. Dengan menggunakan arsitektur ini kedepannya dapat digunakan berbagai client seperti web dan mobile. Dan sebagai pengganti *k-vold cross validation* dapat menggunakan *percentage* split agar metode untuk proses validasi dapat lebih baik lagi.

## UCAPAN TERIMA KASIH

Penulis senantiasa mengucap puji syukur yang sangat besar terhadap Tuhan Yang Maha Esa atas segala berkah, rahmat dan juga hidayahNya, sehingga penulis mampu menyelesaikan artikel ilmiah ini dengan baik. Terimakasih penulis ucapkan juga kepada kedua Orang tua yang selalu memberikan dukungan serta semangat, dan Dosen Pembimbing Skripsi yang selalu memberikan saran dan juga masukan yang sangat membantu penulis, teman serta sahabat yang selalu memberikan dukungan dan dorongan kepada penulis dalam melakukan penelitian. Terimakasih juga kepada diri sendiri karena dapat berkompromi dalam menjaga komitmen untuk menggapai tujuan yang ingin dicapai.

## REFERENSI

[1] Victoria, “BBC,” 24 Juli 20202. [Online]. Available: https://www.bbc.com/indonesia/majalah-53522290

[2] Dhruv, A., Patel, R., & Doshi, N. (2021). Python The Most Advanced Language for Computer Science Applications. *Pandit Deendayal Petroleum University,* 20(20), 292-299. https://doi.org/ 10.5220/0010307902920299
[3] Priyangka, A., & Kumara, I. (2021). Classification Of Rice Plant Diseases Using the Convolutional Neural Network Method. *Department of Information Technology Udayana University,* 12(2), 123. https://doi.org/10.24843/lkjiti.2021.v12.i02.p06
[4] Hendriyana, H., Hilman, Y. (2020). Identification of Types Wood using Convolutional Neural Network with Mobilenet Architecture. Jurnal RESTI *(Rekayasa Sistem dan Teknologi Informasi),* 4(1), 70-76. https://doi.org/10.29207/resti.v4i1.1445
[5] Fitrianto, A & Sartono, B. (2021). Image Classification of Beef and Pork Using Convolutional Neural Network in Keras Framework. Department of Statistic IPB, 05(02), 5-8. https://journal.trunojoyo.ac.id/ijseit
[6] Galuh, “dqlab,” 29 Maret 2021. [Online]. Available: https://www.dqlab.id/yuk-kenali-apa-itu-algoritma-deep-learning
[7] Tammina, S (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. Jurnal International Journal of Scientific and Research Publications (IJSRP), 9(10), p9420. https://doi.org/10.29322/ijsrp.9.10.2019.p9420
[8] Adedeji, O & Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network, Procedia Manufacturing, 35, 607-612. https://doi.org/10.1016/j.promfg.2019.05.086
[9] Robith, “sekawanmedia,” 7 Agustus 2020. [Online]. Available: https://www.sekawanmedia.co.id/blog/pengertian-framework/ [Accessed 10 April 2022].
[10] Banerjee, P., Kumar, B., & Singh. (2019). Implementation of Database Using Python Flask Framework. Department of Electronics and Communication Engineering HMRITM, 8(24), 24890-24893. https://doi.org/ 10.18535/ijecs/v8i12.4399
[11] https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615
[12] https://drive.google.com/drive/folders/1CTvT_gnTvwlcKwJ8yz4jUOs0JYTKrplA
[13] https://www.kaggle.com/asdasdasasdas/garbage-classification