

Implementasi Model Convolutional Neural Network dalam Aplikasi Android untuk Identifikasi Limbah Infeksius

Affix Mareta^{1*}, Beta Estri Adiana², Olivia Wardhani³, Ikhwan Alfath Nurul Fathony⁴
^{1,2,3,4}Teknologi Informasi / Fakultas Teknik, Universitas Tidar
*email: affixmareta@untidar.ac.id

DOI: <https://doi.org/10.31603/komtika.v8i2.12693>

Received: 21-11-2024, Revised: 23-11- 2024, Accepted: 25-11-2024

ABSTRACT

After the Covid 19 pandemic passed, Indonesian citizen were still strict about using masks because active cases were still found. However, not all Indonesian people are aware that masks are an infectious waste, so after use they are still disposed of carelessly. Apart from masks, other infectious waste in the form of battery waste which contains hazardous chemicals and food waste are potentially to spread infectious diseases, is also dangerous for humans. These kind of waste are major contributor to global pollution. Research on waste classification has been carried out a lot, but especially for infectious waste has not received much attention from researchers. For this reason, this research is useful to help the public distinguish infectious waste such as used food scraps, masks, and batteries so that they are more careful in disposing of waste. The research started with collecting datasets, which came from combining several infectious waste datasets available on the internet. This is done because there is no publicly available dataset that specifically contains infectious waste. Then, a classification model is created with Convolutional Neural Network (CNN) algorithm which has a accuracy of more than 90%. This algorithm has been widely used in previous studies, but has never been used as a model applied to android applications to classify infectious waste. In this study, the CNN model is applied to Android applications. From this research an Android application with the CNN algorithm will be produced which can help Indonesians identify infectious waste with an accuracy of 94%.

Keywords: GlobalPollution, Waste Classification, Android Application, Covid19.

ABSTRAK

Selepas pandemic Covid 19 berlalu, penduduk Indonesia masih ketat dalam penggunaan masker karena masih ditemukan kasus aktif. Meski demikian, tidak semua masyarakat Indonesia sadar bahwa masker merupakan salah satu sampah infeksius dan, sehingga setelah digunakan masih dibuang sembarangan. Selain masker, sampah infeksius lain berupa sampah baterai yang mengandung bahan kimia berbahaya dan sisa makanan yang berpotensi menyebarkan penyakit menular, juga berbahaya bagi manusia. Sampah-sampah ini merupakan penyumbang utama polusi global. Penelitian klasifikasi sampah sudah banyak dilakukan, akan tetapi khusus untuk sampah infeksius juga belum mendapat banyak perhatian dari peneliti. Untuk itu penelitian ini bermanfaat untuk membantu masyarakat membedakan sampah infeksius seperti bekas sisa makanan, masker, dan baterai sehingga lebih berhati-hati dalam membuang sampah. Penelitian dimulai dari pengumpulan dataset, yang berasal dari penggabungan beberapa dataset sampah infeksius yang tersedia di internet. Hal ini dilakukan karena belum ada dataset yang tersedia secara publik dan khusus berisi sampah infeksius. Kemudian dilakukan pembuatan model klasifikasi dibantu dengan algoritma Convolutional Neural Network (CNN) yang telah memiliki akurasi pengenalan lebih dari 90%. Meskipun algoritma ini telah banyak digunakan, akan tetapi belum pernah diimplementasikan sebagai model yang diterapkan pada aplikasi android untuk memilah sampah infeksius. Untuk itu, penelitian ini bertujuan untuk menerapkan model CNN pada aplikasi Android. Dari penelitian ini akan dihasilkan aplikasi Android dengan algoritma CNN yang dapat membantu penduduk Indonesia dalam mengenali sampah infeksius dengan akurasi sebesar 94%.

Kata kunci: Polusi Global, CNN, Waste Classification, Android Application, Covid19

PENDAHULUAN

Indonesia telah mengalami pandemic COVID 19 dari tahun 2020 hingga paruh akhir 2021, dan bahkan hingga penelitian ini dilakukan status pandemi masih diberlakukan oleh Pemerintah Indonesia. Hal ini membuat budaya menggunakan masker sangat dipatuhi oleh masyarakat hingga kini. Akan tetapi masih sedikit masyarakat yang sadar bahwa masker merupakan sampah infeksius berbahaya. Di lingkungan sekitar, masih ditemukan masker yang dibuang sembarangan. Jika masker saja belum dikenali sebagai sampah infeksius, apalagi jenis sampah infeksius lain seperti perban, tisu, bekas bungkus makanan dan perban. Sampah-sampah ini merupakan penyumbang utama polusi global.

Dari uraian singkat di atas, maka peneliti mencoba membantu masyarakat Indonesia untuk lebih mengenali sampah infeksius berbahaya dengan membuat aplikasi Android. Android dipilih karena masyarakat Indonesia sudah akrab dengan aplikasi android serta penggunaannya. Menurut data pemerintah Indonesia[1], pada tahun 2017, 67 persen masyarakat Indonesia menggunakan smartphone dengan pengguna Android mencapai 66%.

Tentu agar aplikasi Android yang dibuat bisa melakukan klasifikasi pada sampah infeksius, perlu dibuat model machine learning dengan algoritma tertentu[2]–[4]. Model *machine learning* adalah representasi matematis dari suatu masalah atau fenomena yang ingin dipecahkan atau diprediksi menggunakan teknik-teknik *machine learning*. Model tersebut menggambarkan hubungan antara input (fitur, foto/citra digital) dengan output (prediksi atau klasifikasi) yang dihasilkan oleh algoritma *machine learning*. Model ini amat bergantung pada *dataset* berupa citra atau foto sampah yang sudah tersedia di internet. Kaggle dan Google menjadi rujukan utama dalam mencari *dataset* karena tersedia secara gratis. Akan tetapi *dataset* yang tersedia tidak khusus membahas sampah infeksius, hal ini merupakan hambatan yang harus diatasi.

Penelitian *machine learning* pada klasifikasi sampah biasanya ditujukan untuk mengklasifikasi besi (metal), plastik dan karton (cardboard) karena *dataset* banyak tersedia di kaggle dan google. Misalnya pada penelitian Wulansari, dkk.[5] mengumpulkan 48 hasil penelitian dari berbagai negara selama 5 tahun (2017-2021) mengenai klasifikasi sampah dan 56,8% membahas klasifikasi sampah besi, plastik serta karton. Misalnya penelitian Leonardo dkk[6]., digunakan algoritma Support Vector Machine dan Local Binary Pattern dengan akurasi sebesar 89% untuk sampah metal, dan 87% untuk sampah plastik. Selain itu juga digunakan dalam melakukan klasifikasi sampah daur ulang, seperti plastik, kertas karton dan kaca, seperti dilakukan oleh [6]–[15].

Ricky dkk., membuat aplikasi pengenalan sampah dengan algoritma *Convolutional Neural Network*[15] dengan akurasi sebesar 66,5%. *Dataset* berupa limbah sampah organik dan anorganik yang dikumpulkan dari kaggle dan google. Seperti Ricky dkk., CNN banyak digunakan oleh banyak peneliti dalam klasifikasi sampah organik dan anorganik. Misalnya pada penelitian yang dilakukan oleh Altikat dkk.[16], CNN digunakan untuk melakukan pengenalan pada sampah organik dengan akurasi sebesar 83%. Ibnul Rasidi dkk.[17], melakukan penelitian yang hampir sama dengan akurasi sampah anorganik sebesar 96% dan sampah organik sebesar 62%. Rima dkk.[18] melakukan optimasi pada algoritma CNN untuk mengklasifikasi sampah organik dan anorganik sehingga menghasilkan tingkat akurasi sebenar 91.2%. Adedeji dkk.[19], menggunakan CNN untuk klasifikasi sampah dan memperoleh akurasi 87%.

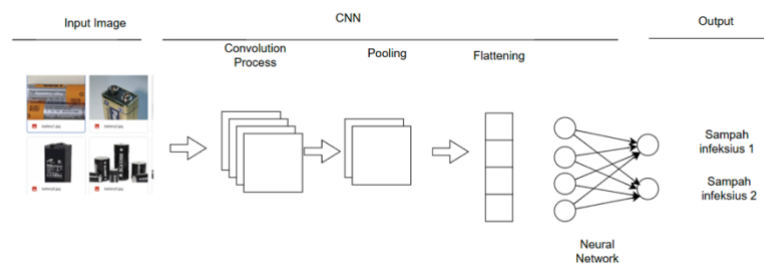
Dari beberapa penelitian yang dibahas sebelumnya, Algoritma *Convolutional Neural Network* (CNN) digunakan karena dapat menghasilkan akurasi klasifikasi yang tinggi. Selain itu CNN juga sudah teruji dalam penelitian klasifikasi sampah[5]. Meski demikian belum banyak yang menggunakannya untuk mengklasifikasi sampah infeksius, seperti yang dilakukan oleh [20], [21]. Akan tetapi terdapat kekurangan pada dua penelitian tersebut. Pada penelitian mengenai *iWaste*[20], datanya berupa video limbah medis (bukan foto), dan akurasi yang dihasilkan sebesar 79%. Pada penelitian Zhou dkk.[21], klasifikasi dengan CNN menghasilkan akurasi sebesar 97.2%, akan tetapi data yang digunakan merupakan data pribadi (dari rumah sakit tertentu di China) dan permodelannya tidak diterapkan pada aplikasi android atau website.

Untuk pembuatan produk berupa aplikasi pengenalan sampah seperti yang dilakukan pada penelitian ini, telah dilakukan sebelumnya oleh Ricky dkk. [15], dalam bentuk website. Website ini dibuat dengan bantuan *framework* Flask. Selain penelitian dari Ricky dkk., dari berbagai referensi yang telah dikumpulkan, dan hasil analisis review paper mengenai penggunaan machine learning dalam klasifikasi sampah dari Wulansari, dkk.[5] belum ada produk pengenalan sampah berbentuk aplikasi Android. Oleh karena penggunaan Android belum banyak digunakan sehingga diharapkan dapat memberikan kebaruan penelitian ini.

Penelitian ini berfokus dalam penerapan algoritma CNN pada aplikasi Android dengan akurasi pengenalan yang lebih baik dari penelitian sebelumnya. Agar hasil akurasi tinggi, dataset dikumpulkan dari hasil kolase dari beberapa dataset sampah infeksius di internet. Bentuk dan jenis sampah medis pun akan lebih banyak. Dataset ini juga diharapkan lebih banyak dari dataset Zhou, dkk[21]., yang hanya 400-500 citra sampah per kelas.

METODE PENELITIAN

Pembuatan model diperlukan agar aplikasi Android yang dibuat nantinya dapat membedakan jenis sampah infeksius. Dalam proses pembuatan model dengan metode *Convolutional Neural Network* atau CNN, dibutuhkan 3 komponen seperti yang disajikan pada Gambar 1, yaitu: *input image* atau gambar yang akan diolah, proses penghitungan dengan CNN, dan *output*.



Gambar 1. Metode penelitian untuk pembuatan model CNN

Pembuatan Model CNN

Pembuatan model CNN dimulai dengan pengumpulan foto sampah infeksius. Pada *input image*, jika gambar berwarna kita bisa menggunakan input dengan 3d array, karena memiliki kanal warna Red, Green dan Blue. Setelah data berhasil didapatkan tahap berikutnya yaitu ekstraksi fitur dengan melakukan konvolusi. Proses konvolusi merupakan proses utama yang mendasari CNN. Proses ini mirip dengan *filtering image* menggunakan konvolusi di

bidang digital *image processing*. Caranya yaitu dengan menggunakan kernel yang dilakukan proses konvolusi dengan gambar 2 dimensi. Proses konvolusi ini dilakukan secara keseluruhan pada seluruh bagian citra, dengan urutan mulai di kiri atas menuju kanan bawah. Proses konvolusi antara kernel dan citra ini bertujuan untuk mendapatkan *feature map*. Kegunaan dari *feature map* ini untuk memperkecil ukuran image. Fitur map ini nantinya yang akan diolah pada perhitungan selanjutnya.

Tahap selanjutnya yaitu memanfaatkan fungsi aktivasi untuk mengubah data-data yang tidak linier menjadi linier dengan ReLU (*Rectified Linear Unit*), dengan $f(x)=\max(0,x)$ yang dipilih karena sederhana dalam penggunaannya. Output ReLU akan nol jika inputnya negatif, dan akan menghasilkan nilai yang sama dengan inputnya jika inputnya positif. Pada penelitian ini ReLU berfungsi untuk mengubah data non-linear seperti piksel, batas, warna menjadi data yang linier.

Pooling layer adalah bagian berikutnya dari CNN. Fungsinya adalah untuk melakukan reduksi spasial pada input dengan melakukan operasi *down-sampling*, tujuannya untuk mengurangi jumlah parameter. Kemudian digunakan metode *max pooling*, di mana nilai terbesar dari setiap wilayah diambil sebagai representasi. Selanjutnya yaitu dilakukan metode *flattening*, sebelum data akan masuk dalam perhitungan *neural network*. *Flattening* merupakan cara mengubah data yang bersifat multidimensi menjadi satu dimensi (1D), hal ini karena lapisan *fully connected layer* dalam *neural network* hanya dapat menggunakan data satu dimensi. *Fully connected layer* sendiri, merupakan komponen terakhir pada jaringan CNN, yang bertujuan untuk melakukan klasifikasi dari data fitur yang telah diekstraksi sebelumnya.

Pengembangan Aplikasi Android

Metode pengembangan SDLC (*Software Development Life Cycle*) dengan *Agile* adalah kerangka kerja yang digunakan untuk mengembangkan dan mengelola proyek perangkat lunak. Metodologi ini terdiri dari tahap analisis kebutuhan atau *requirements*, tahap desain, tahap pengembangan (*coding*), kemudian tahap uji coba (*test*), kemudian diakhiri dengan tahap *deploy*.



Gambar 2. Metode SDLC dengan Agile

1. Analisis Kebutuhan

Tahapan pembuatan aplikasi Android dimulai dengan analisis kebutuhan. Untuk kebutuhan fungsional yaitu pengguna (*user*) dapat mengambil foto limbah infeksius lalu mendeteksi jenis limbah infeksius. Untuk kebutuhan model yaitu kebutuhan model CNN yang dibuat.

2. Desain

Tahap kedua yaitu desain yaitu berupa desain wireframe/mockup aplikasi, dan desain model CNN yang sudah dijelaskan sebelumnya.



Gambar 3. Wireframe halaman aplikasi android

3. Pengembangan (*Development*)

Pada tahap ketiga ini dipersiapkan instalasi Android Studio dan Software Development Kit serta library Machine Learning. Setelah mempersiapkan Android Studio, kemudian dilakukan menulis bahasa pemrograman (coding) dengan bahasa Java dan menambahkan asset. Kemudian dilanjutkan dengan tahap Build and Run atau mencoba hasil coding yang telah ditulis pada gawai android atau emulator Android yang tersedia di Android Studio.

4. Pengujian (*Test*):

Tahap selanjutnya yaitu ketika pengembang Android mulai melakukan tes pada aplikasi yang dibuat dengan metode *black box*. Pengujian ini menekankan pada aspek fungsionalitas, yang diperlihatkan oleh Tabel 1.

Tabel 1. Pengujian Black Box

No	Kriteria	Penjelasan
1	Masuk halaman awal	a. Aplikasi dapat diinstal di gawai android b. Aplikasi dapat masuk halaman awal
2	Klik tombol <i>Take Picture</i>	a. Tombol <i>Take Picture</i> dapat berjalan b. Setelah take picture dan mengambil gambar, jenis sampah infeksius dapat dikenali.
3	Klik tombol <i>Launch Gallery</i>	a. Tombol <i>Launch Gallery</i> dapat berjalan b. Setelah pilih gambar di gallery, jenis sampah infeksius dapat dikenali.

5. *Deploy*:

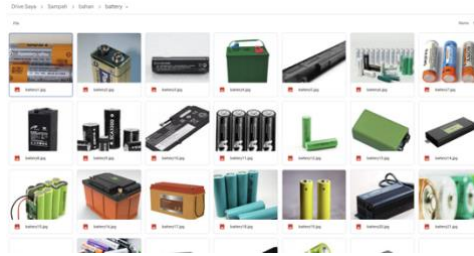
Tahap terakhir dengan menginstal aplikasi pada gawai dengan sistem operasi Android. Pada pengembangannya, aplikasi ini dapat digunakan untuk memilah sampah infeksius dan dapat dikembangkan lebih lanjut (membuat versi lebih baru) dengan menambah jenis sampah yang bisa dikenali.

HASIL DAN PEMBAHASAN

Persiapan *Dataset*

Hal pertama yang dilakukan yaitu pembuatan *dataset* limbah infeksius di Indonesia. *Dataset* ini didapatkan dengan mencarinya dengan mesin pencari Google secara otomatis, dan juga mencari database yang sudah ada di Kaggle. *Dataset* ini bisa digunakan karena telah

memenuhi kriteria *fair use*, atau pengecualian yang mengizinkan karya yang dilindungi oleh hak cipta untuk digunakan tanpa izin dalam kasus tertentu yang “bermanfaat bagi masyarakat.”

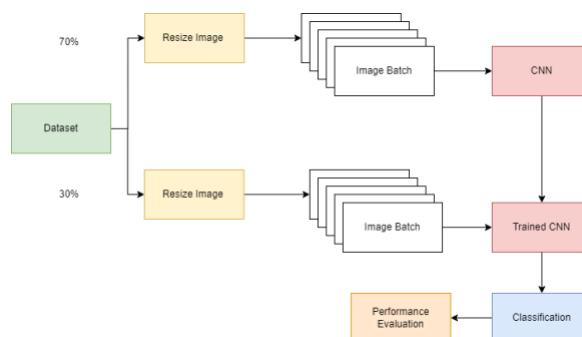


Gambar 3. Contoh dataset yang digunakan

Pada gambar di atas, merupakan beberapa contoh citra yang dikumpulkan dari Google dan Kaggle. Citra yang dikumpulkan yaitu limbah infeksius berupa sampah baterai dan sampah bekas makanan. Sampah baterai, seperti terlihat pada gambar di atas, tidak hanya terbatas pada baterai berbentuk silinder tapi juga baterai aki. Sementara sampah sisa makanan yaitu buah-buahan dan roti yang sudah membusuk. Setelah itu citra dibagi menjadi tiga folder, pada folder pertama terdapat 940 citra untuk baterai, folder kedua berisi 980 citra sampah bekas makanan dan folder ketiga berisi 900 citra masker.

Pembuatan Model CNN

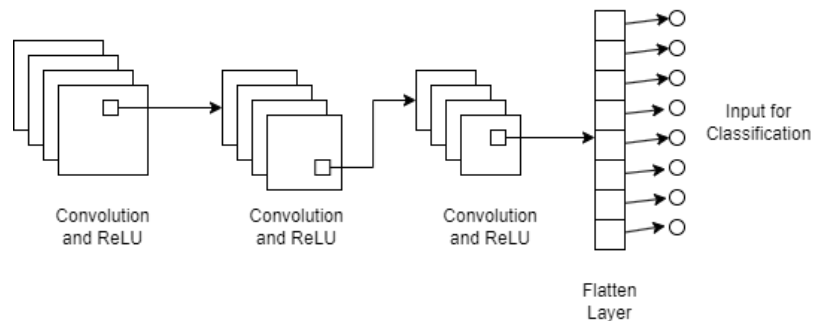
Tahap pertama yaitu membagi citra yang sudah dikumpulkan ke dalam tiga folder, folder pertama untuk *training*, folder kedua untuk validasi dan folder ketiga untuk *testing*. Masing-masing *dataset* di atas (*training*, validasi dan *testing*) diinisialisasi dengan memanggil fungsi "tf.keras.utils.image_dataset_from_directory" yang membaca data gambar dari direktori tertentu (*train_dir*, *validation_dir*, atau *test_dir*) dengan ukuran gambar (*img_height*, *img_width*) = (32, 32) dan *batch_size* = 20. Hal ini bertujuan untuk preprocessing data dengan mengubah ukuran data menjadi lebih kecil.



Gambar 4. Pembagian dataset *testing* dan *training*

Setelah data berhasil didapatkan tahap berikutnya yaitu kita tentukan layer CNN dan fungsi aktivasi yang akan digunakan. model = tf.keras. *Sequential* bertujuan untuk membuat sebuah objek model *Sequential Keras* yang baru. Untuk fungsi *tf.keras.layers.Rescaling(1./255)*, bertujuan untuk mengubah rentang nilai *pixel* gambar menjadi 0 sampai 1. Kemudian ditambahkan layer Conv2D dengan dengan ukuran kernel 3x3 dan digunakan *ReLU* untuk fungsi aktivasinya. Sementara itu, fungsi *tf.keras.layers.MaxPooling2D()*, bertujuan untuk

menambahkan layer *MaxPooling2D* kedua untuk mengurangi resolusi spasial *output* dari layer sebelumnya. Selanjutnya fungsi *tf.keras.layers.Flatten()*, berguna untuk mengubah output dari layer sebelumnya menjadi satu dimensi. Fungsi *tf.keras.layers.Dense(2)* berguna untuk menambahkan *layer Dense* terakhir sebagai output model.



Gambar 5. Penentuan jumlah layer dan fungsi aktivasi pada CNN

Langkah selanjutnya yaitu proses training dan validasi untuk melatih model yang telah dibuat. Proses ini dilakukan selama 10 *epochs* dan berhasil mendapat tingkat akurasi sebesar 94,82%. Pada Gambar 5 fungsi *model.evaluate* adalah sebuah fungsi yang digunakan untuk mengevaluasi kinerja model machine learning. Fungsi ini mengembalikan nilai keakuratan (accuracy) dan metrik lain yang ditentukan sebelumnya selama proses pelatihan (training).

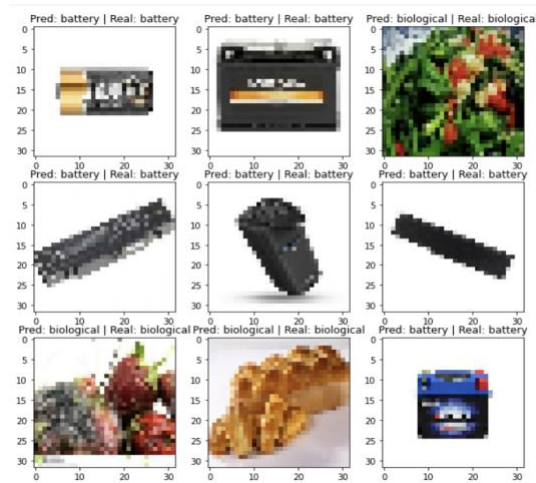
```
Epoch 1/10
68/68 [=====] - 8s 76ms/step - loss: 0.3565 - accuracy: 0.8474 - val_loss: 0.2358 - val_accuracy: 0.9021
Epoch 2/10
68/68 [=====] - 5s 71ms/step - loss: 0.2237 - accuracy: 0.9222 - val_loss: 0.2235 - val_accuracy: 0.9124
Epoch 3/10
68/68 [=====] - 5s 65ms/step - loss: 0.2015 - accuracy: 0.9296 - val_loss: 0.2317 - val_accuracy: 0.9330
Epoch 4/10
68/68 [=====] - 5s 67ms/step - loss: 0.1850 - accuracy: 0.9311 - val_loss: 0.2216 - val_accuracy: 0.9227
Epoch 5/10
68/68 [=====] - 5s 69ms/step - loss: 0.1666 - accuracy: 0.9422 - val_loss: 0.2147 - val_accuracy: 0.9278
Epoch 6/10
68/68 [=====] - 6s 80ms/step - loss: 0.1722 - accuracy: 0.9400 - val_loss: 0.2036 - val_accuracy: 0.9278
Epoch 7/10
68/68 [=====] - 5s 70ms/step - loss: 0.1332 - accuracy: 0.9541 - val_loss: 0.2063 - val_accuracy: 0.9278
Epoch 8/10
68/68 [=====] - 5s 69ms/step - loss: 0.1252 - accuracy: 0.9585 - val_loss: 0.2093 - val_accuracy: 0.9278
Epoch 9/10
68/68 [=====] - 5s 72ms/step - loss: 0.1348 - accuracy: 0.9519 - val_loss: 0.2271 - val_accuracy: 0.9330
Epoch 10/10
68/68 [=====] - 5s 72ms/step - loss: 0.1256 - accuracy: 0.9541 - val_loss: 0.2205 - val_accuracy: 0.9278
<keras.callbacks.History at 0x7f5cfaa96700>

[ ] model.evaluate(test_ds)
20/20 [=====] - 1s 38ms/step - loss: 0.1346 - accuracy: 0.9482
[0.134624183177948, 0.9481865167617798]
```

Gambar 5. Hasil *training*

Pengujian Model

Setelah model dilatih, kemudian diujicobakan menggunakan *dataset testing*. Gambar 6 merupakan gambar hasil uji coba. Terdapat tulisan *pred.* yang berarti prediksi dan *real* atau data sesungguhnya. Dari kesembilan citra yang dijadikan data uji, semuanya menunjukkan kesesuaian antara prediksi dan data sesungguhnya.

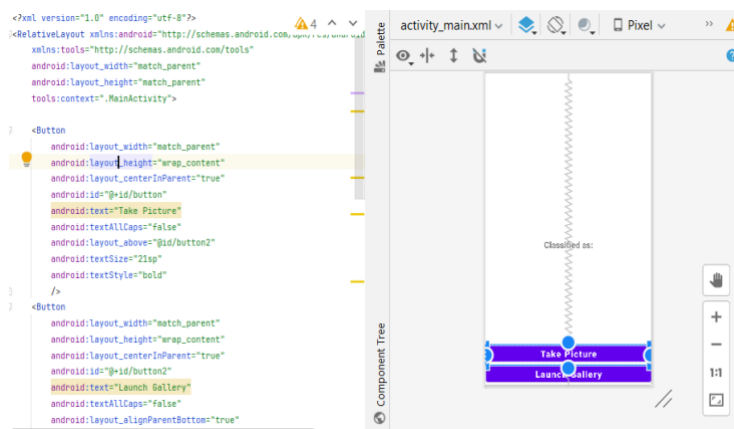


Gambar 6. Hasil uji coba model

Model kemudian diunduh dalam bentuk tflite yang kemudian dipasang pada aplikasi android. *TFLite (TensorFlow Lite)* adalah sebuah framework *open-source* dari TensorFlow yang dirancang khusus untuk menjalankan model *machine learning* pada perangkat *mobile*, *embedded*, dan *IoT* (Internet of Things) dengan sumber daya yang terbatas. TFLite memungkinkan pengembang untuk memperkecil ukuran model dan meningkatkan kecepatan dan efisiensi pada perangkat yang memiliki keterbatasan sumber daya, seperti RAM dan kapasitas penyimpanan.

Pembuatan Aplikasi Android

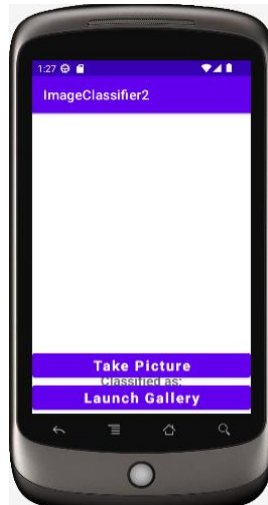
Sebelum membuat activity pada aplikasi android, terlebih dahulu membuat tampilan atau *UI/UX*. Tampilan ini ditulis dalam bahasa XML android, menggunakan IDE android studio. Contoh proses pembuatannya ditampilkan pada hasil cuplikan layar berikut ini. :



Gambar 7. Pembuatan UI/UX

Setelah dilakukan pembuatan *UI/UX* selanjutnya model CNN yang dibuat menggunakan tensorflow dipasang pada aplikasi android dengan menambahkan *tensorflow lite model*. Model ini berupa fungsi yang bisa ditambahkan pada *main activity*. Model tensor flow lite di langkah sebelumnya, kemudian dipanggil pada aktivitas utama, atau *MainActivity.java*. Setelah itu tombol-tombol pada UI/UX dipanggil oleh fungsi pada *MainActivity* sehingga dapat berfungsi

dengan baik. Aplikasi pemilahan sampah ini memiliki file size sebesar 6,2 *mega byte*. Sehingga dapat dijalankan dengan gawai android dengan RAM yang kecil atau kurang dari 2GB. Untuk versi minimal android yaitu sdk 23, sehingga versi lebih rendah tidak bisa menjalankan aplikasi ini. Sementara target sdk yaitu 33, yang berarti aplikasi ini berjalan baik pada gawai dengan sdk android 33. Berikut ini merupakan tampilan ketika aplikasi berhasil dijalankan:



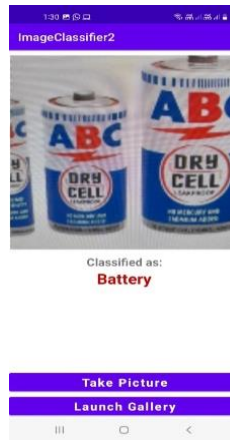
Gambar 8. Menu utama

Untuk menggunakannya, pertama kita harus menginstal dulu ekstensi .apk, kemudian setelah berhasil dijalankan, tekan tombol “*Take Picture*”, sehingga muncul fungsi untuk menggunakan kamera smartphone.



Gambar 9. Pembuatan UI/UX

Setelah itu dilanjutkan dengan klik ‘OK’ pada sudut kanan bawah sehingga didapatkan hasil klasifikasi berupa baterai. Hal ini juga bisa kita lakukan dengan sisa makanan maupun masker medis.



Gambar 10. Hasil pengenalan

Pengujian Aplikasi Android

Pengujian *Black box* yang digunakan sesuai dengan kebutuhan penelitian karena hanya bertujuan untuk memastikan semua fitur atau tombol pada aplikasi berjalan dengan baik, dan agar pembahasan tidak terlalu luas. Dalam pengujian *black box*, aplikasi diuji dengan memberikan input berupa data atau perintah, kemudian keluaran atau respon aplikasi yang dihasilkan diamati untuk mengevaluasi kinerja dan fungsionalitasnya. Dalam penelitian ini, input yang diberikan berupa gambar limbah infeksius yang diambil melalui kamera pada aplikasi Android, dan keluaran yang diharapkan adalah hasil klasifikasi jenis limbah infeksius yang benar.

Tabel 2. Pengujian *Black Box*

No	Kriteria	Penjelasan	Kesimpulan
1	Masuk halaman awal	c. Aplikasi dapat diinstal di gawai android d. Aplikasi dapat masuk halaman awal	[v] Diterima [] Ditolak
2	Klik tombol <i>Take Picture</i>	c. Tombol <i>Take Picture</i> dapat berjalan d. Setelah take picture dan mengambil gambar, jenis sampah infeksius dapat dikenali.	[v] Diterima [] Ditolak
3	Klik tombol <i>Launch Gallery</i>	c. Tombol <i>Launch Gallery</i> dapat berjalan d. Setelah pilih gambar di gallery, jenis sampah infeksius dapat dikenali.	[v] Diterima [] Ditolak

Selama pengujian, semua kriteria pada Tabel 2 dapat berfungsi dengan baik. Hasil klasifikasi yang dikeluarkan oleh aplikasi Android setelah melakukan proses pengenalan menggunakan model Convolutional Neural Network juga ditemukan cukup akurat dan sesuai dengan jenis limbah infeksius yang diinputkan.

KESIMPULAN

Dengan dataset sampah infeksius yang dikumpulkan dari internet hanya sebanyak 900 citra per kelas model CNN, model telah berhasil mengklasifikasi sampah infeksius. Meski demikian, jumlah *dataset* per kelas lebih banyak dibandingkan Chou, dkk., yang hanya

mengumpulkan sebanyak 400-500 foto/citra. Jika dibandingkan penelitian yang dilakukan oleh Ricky dkk., dengan akurasi 66%, model CNN dalam penelitian ini telah menghasilkan akurasi lebih baik, yaitu sebesar 94%. Hasil pengenalan pun dapat digunakan dalam mengenali jenis limbah infeksius yang berbeda. Penerapan model CNN pada aplikasi Android dilakukan agar masyarakat dapat dengan mudah mengklasifikasikan sampah infeksius seperti masker, sisa makanan dan baterai. Hal ini dapat membantu masyarakat menjadi lebih sadar akan pentingnya pengelolaan limbah infeksius yang baik dan benar. Meskipun model memiliki akurasi yang tinggi, terdapat beberapa kendala yang masih perlu diperbaiki seperti dalam pengenalan limbah infeksius yang kurang jelas, sehingga dapat memengaruhi akurasi pengenalan. Oleh karena itu, berikutnya dapat dilakukan penelitian yang berfokus untuk memperbaiki model CNN dan menambahkan data sampah infeksius yang lebih lengkap dan bervariasi, sehingga dapat meningkatkan akurasi pengenalan dan klasifikasi limbah infeksius secara lebih optimal.

DAFTAR PUSTAKA

- [1] KOMINFO, "Survey Penggunaan TIK 2017," 2017.
- [2] A. Géron, *Hands-On Machine Learning with*. O'Reilly Media, 2023.
- [3] A. S. Glassner, *Deep learning : a visual approach / Andrew Glassner*. 2021.
- [4] A. W. Trask, *Grokking Deep Learning*, 1st ed., no. 1. New York, New York, USA: Manning Publications, 2020.
- [5] A. Wulansari, A. Setyanto, and E. T. Luthfi, "Systematic Literature Review of Waste Classification Using Machine Learning," *J. Informatics Telecommun. Eng.*, vol. 5, no. 2, pp. 405–413, 2022, doi: 10.31289/jite.v5i2.6211.
- [6] L. Leonardo, Y. Yohannes, and E. Hartati, "Klasifikasi Sampah Daur Ulang Menggunakan Support Vector Machine Dengan Fitur Local Binary Pattern," *J. Algoritm.*, vol. 1, no. 1, pp. 78–90, 2020, doi: 10.35957/algoritme.v1i1.440.
- [7] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," *CS229Project Rep.*, pp. 1–6, 2016.
- [8] F. A. Azis, H. Suhaimi, and E. Abas, "Waste Classification using Convolutional Neural Network," *ACM Int. Conf. Proceeding Ser.*, pp. 9–13, 2020, doi: 10.1145/3417473.3417474.
- [9] S. Meng and W. T. Chu, "A Study of Garbage Classification with Convolutional Neural Networks," *Indo - Taiwan 2nd Int. Conf. Comput. Anal. Networks, Indo-Taiwan ICAN 2020 - Proc.*, pp. 152–157, 2020, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181311.
- [10] D. Gyawali, A. Regmi, A. Shakya, A. Gautam, and S. Shrestha, "Comparative Analysis of Multiple Deep CNN Models for Waste Classification," 2020, [Online]. Available: <http://arxiv.org/abs/2004.02168>
- [11] F. Song, Y. Zhang, and J. Zhang, "Optimization of CNN-based Garbage Classification Model," *ACM Int. Conf. Proceeding Ser.*, 2020, doi: 10.1145/3424978.3425089.
- [12] W. Mao, W. Chen, C. Wang, and Y. Lin, "Resources , Conservation & Recycling Recycling waste classification using optimized convolutional neural network," *Resour. Conserv. Recycl.*, vol. 164, no. July 2020, p. 105132, 2021, doi: 10.1016/j.resconrec.2020.105132.
- [13] S. Li, M. Yan, and J. Xu, "Garbage object recognition and classification based on Mask

- Scoring RCNN,” *Proc. - 2020 Int. Conf. Cult. Sci. Technol. ICCST 2020*, pp. 54–58, 2020, doi: 10.1109/ICCST50977.2020.00016.
- [14] M. H. Zayd, M. W. Oktavian, D. G. T. Meranggi, J. A. Figo, and N. Yudistira, “Improvement of garbage classification using pretrained Convolutional Neural Network,” *Teknologi*, vol. 12, no. 1, pp. 1–8, 2022, doi: 10.26594/teknologi.v12i1.2403.
- [15] P. N. Dacipta and R. E. Putra, “Sistem Klasifikasi Limbah Menggunakan Metode Convolutional Neural Network (CNN) Pada Webservice Berbasis Framework Flask,” *J. Informatics Comput. Sci.*, vol. 3, no. 04, pp. 394–402, 2022, doi: 10.26740/jinacs.v3n04.p394-402.
- [16] A. Altikat, A. Gulbe, and S. Altikat, “Intelligent solid waste classification using deep convolutional neural networks,” *Int. J. Environ. Sci. Technol.*, vol. 19, no. 3, pp. 1285–1292, 2022, doi: 10.1007/s13762-021-03179-4.
- [17] A. Ibnul Rasidi, Y. A. H. Pasaribu, A. Ziqri, and F. D. Adhinata, “Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network,” *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, pp. 142–149, 2022, doi: 10.28932/jutisi.v8i1.4314.
- [18] 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 Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 312–318, 2021, doi: 10.29207/resti.v5i2.2754.
- [19] O. Adedeji and Z. Wang, “Intelligent waste classification system using deep learning convolutional neural network,” *Procedia Manuf.*, vol. 35, pp. 607–612, 2019, doi: 10.1016/j.promfg.2019.05.086.
- [20] J. Chen, J. Mao, C. Thiel, and Y. Wang, “IWaste: Video-Based Medical Waste Detection and Classification,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 5794–5797, 2020, doi: 10.1109/EMBC44109.2020.9175645.
- [21] H. Zhou *et al.*, “A deep learning approach for medical waste classification,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–9, 2022, doi: 10.1038/s41598-022-06146-2.

