

PERANCANGAN APLIKASI *MOBILE* UNTUK KLASIFIKASI SAYURAN MENGGUNAKAN *DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK*

Muhamad Jaelani Akbar¹⁾, Mochamad Wisuda Sardjono²⁾, Margi Cahyanti³⁾ dan Ericks Rachmat Swedia⁴⁾

^{1,4}Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma

^{2,3}Jurusan Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma

^{1,2,3,4}Jalan Margonda Raya No. 100, Depok, 16424

E-mail: m.jaelaniakbar@gmail.com¹⁾, moch_wisuda@staff.gunadarma.ac.id²⁾, margi@staff.gunadarma.ac.id³⁾ dan ericks_rs@staff.gunadarma.ac.id

ABSTRAK

Sayuran merupakan sebutan bagi bahan pangan asal tumbuhan yang biasanya mengandung kadar air tinggi dan dikonsumsi dalam keadaan segar atau setelah diolah secara minimal. Keanekaragaman sayur yang terdapat di dunia menyebabkan keragaman pula dalam pengklasifikasian sayur. Oleh karena itu diperlukan adanya pendekatan digital agar dapat mengenali jenis sayuran dengan cepat dan mudah. Dalam penelitian ini jumlah jenis sayuran yang digunakan sebanyak 7 jenis di antara: brokoli, jagung, kacang panjang, pare, terung ungu, tomat dan kubis. Dataset yang digunakan berjumlah 941 gambar sayur dari 7 jenis sayur, ditambah 131 gambar sayur dari jenis yang tidak terdapat pada dataset, selain itu digunakan 291 gambar selain sayuran. Untuk melakukan klasifikasi jenis sayuran digunakan algoritme *Convolutional Neural Network* (CNN), yang merupakan salah satu bidang ilmu baru dalam *Machine Learning* dan berkembang dengan pesat. CNN merupakan salah satu algoritme yang terdapat pada metode *Deep Learning* dengan memiliki kemampuan yang baik dalam *Computer Vision*, salah satunya yaitu *image classification* atau klasifikasi objek citra. Uji coba dilakukan pada lima perangkat seluler berbasis sistem operasi Android. Python digunakan sebagai bahasa pemrograman dalam merancang aplikasi *mobile* ini dengan menggunakan modul Tensor flow untuk melakukan *training* dan *testing* data. Metode yang dapat digunakan dalam melakukan klasifikasi citra ini yaitu *Convolutional Neural Network* (CNN). Hasil *final test accuracy* yang diperoleh yaitu didapat keakuratan mengenali jenis sayuran sebesar 98.1% dengan salah satu hasil pengujian yaitu klasifikasi sayur jagung dengan akurasi sebesar 99.98049%.

Kata kunci: Sayuran, *Computer Vision*, *CNN*, *Image Classification*, *Accuracy*

1. PENDAHULUAN

Perkembangan teknologi dalam kehidupan manusia sangat mempengaruhi segala kinerja tiap individu dalam melakukan aktivitasnya, teknologi membantu manusia dalam melakukan segala pekerjaan menjadi lebih efektif dan efisien terhadap waktu. Pada bidang pertanian merupakan salah satu contoh teknologi yang dapat berguna untuk memudahkan masyarakat dalam membedakan jenis sayuran.

Sayuran merupakan bahan pangan asal tumbuhan yang mengandung kadar air tinggi dan dikonsumsi dalam keadaan segar atau setelah diolah secara minimal. Keanekaragaman sayur yang terdapat di dunia menyebabkan keragaman pula dalam pengklasifikasian sayur yaitu berdasarkan organ yang dimakan, klasifikasi berdasarkan cara budidaya, klasifikasi berdasarkan syarat tumbuh terutama suhu, dan klasifikasi botani (Nurainy, 2018).

Mengetahui banyaknya klasifikasi yang ditemukan pada sayuran, diperlukan adanya pendekatan digital agar dapat mengenali beragam jenis sayuran dengan cepat dan mudah. Salah satu caranya adalah dengan menggunakan teknologi *Deep Learning*, yang merupakan salah satu bidang ilmu baru dalam *Machine Learning* dan memiliki perkembangan yang cukup pesat

(Widodo, 2016). *Convolutional Neural Network* merupakan salah satu metode *Deep Learning* yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital. CNN dapat memecahkan permasalahan *object detection* dan *object recognition*. Pada tahun 2012, penelitian tentang CNN dapat melakukan pengenalan citra digital dengan akurasi yang menyaingi manusia pada dataset tertentu. (Kurniadi, Kusriani, & Sadikin, 2020)

James F. Peter (2017) mengungkapkan bahwa *Computer Vision* adalah cabang ilmu yang memberikan komputer atau mesin dengan penglihatan atau kemampuan untuk melihat seperti manusia (Peters, 2017). Klasifikasi citra saat ini menjadi salah satu masalah yang telah lama dicari solusinya dalam *computer vision*. Tujuan klasifikasi citra itu sendiri adalah mengklasifikasikan beberapa citra pada kategori tertentu. Bagaimana menduplikasikan kemampuan manusia dalam memahami informasi citra digital, supaya komputer dapat mengenali objek pada citra selayaknya manusia.

Selain itu, penggunaan teknologi salah satunya teknologi *mobile* (bergerak) sudah menjadi kebutuhan di masyarakat yang memberikan kemudahan untuk menjalankan aktivitas, mendukung produktivitas dan

mendapatkan informasi. Salah satu macam teknologi *mobile* adalah *smartphone*. Masyarakat dapat memperoleh informasi di dalam aplikasi berbasis android yang diterapkan pada *mobile smartphone*. Android adalah salah satu platform sistem operasi yang digemari masyarakat karena sifatnya yang *open source* sehingga memungkinkan pengguna untuk melakukan pengembangan. Android merupakan generasi baru platform *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi (Hansun, Kristanda, & Saputra, 2018).

Pada penelitian yang dilakukan oleh Ahmad Kurniadi dkk di tahun 2020 menuliskan bahwa beberapa penelitian telah dilakukan pada pengklasifikasian sayuran dan buah dengan menggunakan metode jaringan saraf tiruan. Dalam penelitiannya disebutkan metode yang digunakan untuk klasifikasi diantaranya *Support Machine Vector* (SVM) dan *Convolution Neural Network* (CNN). Diantara dua metode yang digunakan, CNN merupakan metode yang sering digunakan dan menghasilkan tingkat akurasi yang lebih baik (Kurniadi, Kusriani, & Sadikin, 2020).

Berdasarkan latar belakang masalah tersebut, maka dapat dirumuskan sebuah masalah yaitu bagaimana mengimplementasikan konsep *Deep Learning* dengan menggunakan CNN untuk melakukan klasifikasi jenis sayuran.

Penelitian ini bertujuan untuk merancang sebuah aplikasi yang berisikan informasi dari beberapa jenis sayuran menggunakan *deep learning* sehingga sistem dapat mengenali sayuran secara *real-time* sebagai implementasi dari pembelajaran mesin pada *smartphone* android (LeCun, 2015).

Pada penelitian ini Tensor flow digunakan sebagai *library* pada bahasa pemrograman Python (Johansson, 2019). Tensor flow merupakan generasi kedua dari sistem pembelajaran kecerdasan buatan dikembangkan oleh Google yang mendukung jaringan saraf konvolusi (CNN) (Om, 2018).

Pemanfaatan teknologi Deep Learning pada klasifikasi sayuran menghasilkan tingkat performa klasifikasi yang tinggi (Zhu, Zhenbo, Chen, Jing, & Jun, 2018).

2. RUANG LINGKUP

Dalam penelitian ini permasalahan mencakup:

1. Sayuran merupakan bahan pangan asal tumbuhan yang mengandung kadar air tinggi dan dikonsumsi dalam keadaan segar atau setelah diolah secara minimal. Keanekaragaman sayur yang terdapat di dunia menyebabkan keragaman pula dalam pengklasifikasian sayur (Nurainy, 2018). Mengetahui banyaknya keragaman yang ditemukan pada sayuran, diperlukan adanya pendekatan digital agar dapat mengenali nama sayuran dengan cepat dan mudah.
2. Pada penelitian ini citra gambar yang digunakan berjumlah 941 gambar sayur dari 7 jenis sayur,

ditambah 131 gambar sayur dari jenis yang tidak terdapat pada dataset, selain itu digunakan 291 gambar selain sayuran.

3. Dari penelitian ini yang diharapkan menghasilkan sebuah aplikasi pada perangkat telepon selular yang dapat mengenali jenis sayuran dengan mudah dan cepat.

3. BAHAN DAN METODE

Convolutional Neural Network (CNN) merupakan salah satu model *Deep Learning* yang banyak digunakan untuk keperluan analisis citra atau visual. CNN memiliki neuron-neuron yang disusun secara tiga dimensi, memiliki panjang, lebar dan tinggi. Sehingga CNN sangat efektif dan efisien untuk menganalisis *image* atau gambar (Primartha, 2018).

Algoritme CNN cukup banyak diimplementasikan pada berbagai perangkat yang memiliki kamera, seperti: *hand phone*, CCTV, robot dan sebagainya. Algoritme ini didesain secara khusus untuk mengidentifikasi *image* atau gambar dua dimensi (Primartha, 2018). Adapun algoritme CNN dapat dijelaskan sebagai berikut:

1. Membaca dataset berupa citra gambar
2. Memecah gambar menjadi beberapa gambar berukuran kecil (*extract region*)
3. Melakukan perhitungan dengan algoritme CNN
4. Melakukan klasifikasi

Berdasarkan buku yang dituliskan oleh Rifkie Primartha pada tahun 2018, cara kerja CNN dapat dijelaskan terdiri dari 4 langkah, yaitu: *convolution*, *subsampling*, *activation* dan *fully connected*.

Pada langkah pertama, *convolution*, berfungsi menerima input (input citra gambar). *Convolution* secara umum adalah operasi antar dua fungsi sehingga menghasilkan fungsi ketiga yang merupakan modifikasi dari kedua fungsi aslinya. Formulasi operasi pada *convolution* dapat dilihat pada Tabel 1.

Tabel 1. Operasi pada *Convolution*

Fungsi	Persamaan
Umum	$f(x, y) * g(x, y) = f(x, y) \otimes g(x, y) \quad (1)$
Kontinu	$h(x) = f(x) * g(x)$ $= \int_{-\infty}^{\infty} f(a) g(x - a) da \quad (2)$
Diskret	$h(x) = f(x) * g(x)$ $= \sum_{a=-\infty}^{\infty} f(a) g(x - a) \quad (3)$

Langkah selanjutnya adalah *subsampling*, input yang berasal dari langkah *convolution* dapat diperbagus untuk mengurangi tingkat sensitivitas terhadap *noise* proses ini disebut dengan *subsampling* yang dapat dicapai dengan cara mengambil nilai rata-rata atau nilai maksimum dari beberapa buah sinyal. Contoh *subsampling* yaitu

mengurangi ukuran gambar atau mengurangi kontras warna (RGB).

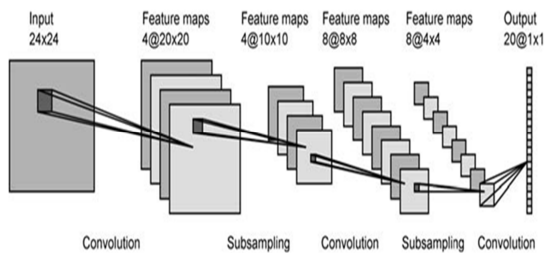
Setelah itu langkah ketiga adalah *activation*, mengendalikannya sinyal yang mengalir dari satu lapisan ke lapisan berikutnya mengikuti cara kerja neuron. Sinyal output yang terkait dengan referensi sebelumnya akan mengaktifkan banyak neuron yang lain, sehingga sinyal dapat di propagasi secara efisien. Pada *multiclass classification* terdapat kendala Ketika output layer memiliki lebih dari 1 neuron. Untuk kasus ini, fungsi aktivasi yang biasa digunakan adalah fungsi *softmax*. Fungsi ini dirumuskan pada persamaan (4).

$$\sigma : \mathbb{R}^K \rightarrow 0,1^K$$

$$\sigma z_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_j}} \text{ for } j = 1, \dots, K \quad (4)$$

Dimana: K: Vektor, z: bilangan riil, σz : bilangan riil antara 0 dan 1 (0,1).

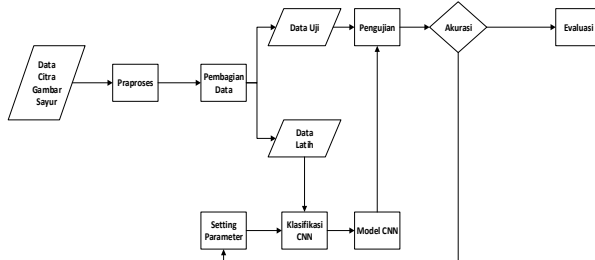
Bentuk *subsampling* dapat dilihat pada Gambar 1.



Gambar 1. *Subsampling*

Langkah terakhir adalah *fully connected layer*. Tujuan langkah ini adalah untuk mengklasifikasi *image* atau gambar, pada penelitian ini jumlah klasifikasi nya tujuh diantaranya: brokoli, jagung, kacang panjang, pare, terung ungu, tomat dan kubis.

Dalam penelitian ini ada beberapa tahapan yang harus dilakukan sebelum melakukan klasifikasi yaitu, akuisisi data, pra-proses data, pembagian data, klasifikasi CNN, pengujian dan evaluasi yang dapat dilihat pada Gambar 2.



Gambar 2. Alur Proses Klasifikasi Sayuran

3.1. Akuisisi Data

Proses akuisisi data citra gambar sayuran dilakukan dengan cara mengunduh data gambar sayuran secara

online menggunakan fasilitas Fatkun Batch Unduh Gambar. Citra gambar yang digunakan berjumlah 941 gambar sayur dari 7 jenis sayur, yaitu Brokoli, Jagung, Kacang Panjang, Kubis, Pare, Terung Ungu, dan Tomat, ditambah 131 gambar sayur dari jenis yang berbeda-beda, ditambah lagi 291 gambar selain sayuran.

3.2. Setting Parameter

Pada tahapan ini nilai parameter telah ditentukan pada saat pembuatan model CNN. Parameter yang digunakan meliputi jumlah *epoch*, jumlah *batch*, ukuran dimensi citra, ukuran *convolution layer*.

3.3. Pra proses

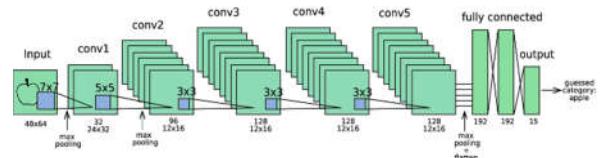
Citra gambar sayur terlebih dahulu dilakukan pra-proses. Tahapan pra proses yang pertama adalah mengubah ukuran piksel citra asli menjadi 32 x 32 piksel. Python 3.6.7 dan Tensor flow 1.13.1 digunakan dalam proses analisis dalam penelitian ini.

3.4. Pembagian Data

Tahap selanjutnya setelah dilakukan pra proses, semua data citra sayur dibagi menjadi dua yaitu data latih dan data uji. Data latih adalah sekumpulan file citra gambar sayur yang diproses oleh jaringan saraf untuk dipelajari dan menghasilkan model yang akan digunakan sebagai pembanding dari data uji. Dalam penelitian ini model yang dihasilkan sebanyak 7 model sesuai dengan jumlah jenis sayuran yang diklasifikasikan.

3.5. Klasifikasi CNN

Dalam proses data yang berbentuk struktur kotak (*grid*), hanya jaringan saraf CNN yang mampu melakukannya. Pada citra gambar yang berbentuk dua dimensi, lapisan konvolusi merupakan operasi aljabar linear yang mengalikan matriks dari filter pada citra yang akan diproses. Lapisan konvolusi ini merupakan lapisan utama yang paling penting digunakan. Jenis lapisan yang penting lainnya adalah lapisan *pooling*, yaitu lapisan yang digunakan untuk mengambil rata-rata atau mengambil nilai maksimal dari bagian-bagian lapisan piksel pada sebuah citra gambar. Arsitektur CNN yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur CNN

3.6. Pengujian

Tahapan selanjutnya adalah pengujian, memasukkan sebuah file citra gambar pada data uji ke dalam model klasifikasi CNN yang telah dibuat. Citra ini akan di masukkan ke dalam *fit model* dan akan menghasilkan sebuah output yang dihitung menggunakan bobot yang telah ditingkatkan. Lalu dibandingkan dengan bobot

model dari data latih, sehingga model klasifikasi data latih dapat mengklasifikasikan data uji tersebut ke kelas yang benar. Tahapan ini dilakukan untuk menghitung nilai akurasi yang dihasilkan pada model klasifikasi data latih.

3.7. Evaluasi

Pada tahapan ini akan dihasilkan nilai akurasi, dimana nilai tersebut akan diujikan. Jumlah data uji yang dinyatakan ter klasifikasi benar dibagi dengan jumlah keseluruhan data uji. Untuk mengetahui tingkat keberhasilan klasifikasi terhadap model hasil data latih maka dilakukan pengujian pada data uji. Banyaknya data uji yang diklasifikasikan secara benar dan tidak benar oleh model bisa didapatkan dari proses evaluasi dari kinerja model. Nilai akurasi dapat dihitung menggunakan formulasi pada persamaan (5).

$$Akurasi = \frac{Jumlah\ data\ uji\ yang\ terklasifikasi\ benar}{Jumlah\ data\ uji\ keseluruhan} \times 100\% \quad (5)$$

Nilai akurasi juga dapat diperoleh melalui *Confusion matrix*. *Confusion matrix* adalah suatu metode perhitungan dalam menghasilkan nilai akurasi yang berbentuk tabel matriks. Tabel ini sering digunakan untuk menggambarkan kinerja dari sebuah model klasifikasi, terdiri dari baris dan kolom sebanyak jumlah kelas yang menunjukkan nilai *false positives*, *false negatives*, *true positives* dan *true negatives*.

Tabel 2. Confusion Matrix Dengan 7 Kelas

X	P1	P2	P3	P4	P5	P6	P7	Y
P1								
P2								
P3								
P4								
P5								
P6								
P7								

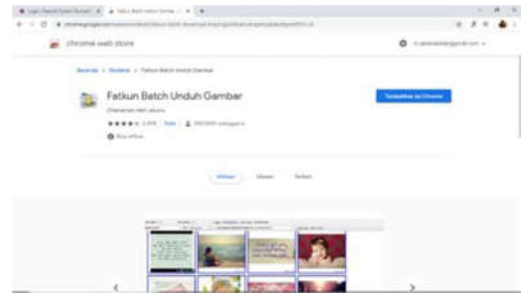
Pada Tabel 2, parameter X menyatakan Aktual, Y menyatakan akurasi dan P1 sampai dengan P7 menyatakan jumlah kelas.

4. PEMBAHASAN

Pada bagian ini dijelaskan tentang tahapan yang dilakukan pada penelitian ini, diantaranya:

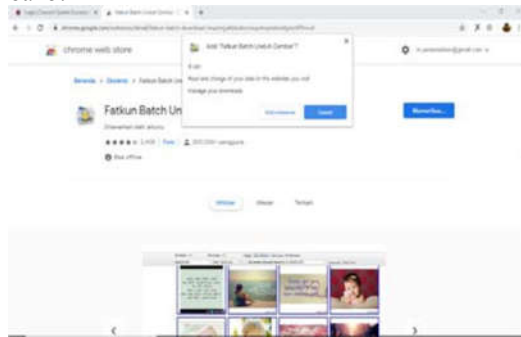
1. Mengunduh Dataset

Dataset yang digunakan berjumlah 941 gambar sayur dari 7 jenis sayur, yaitu Brokoli, Jagung, Kacang Panjang, Kubis, Pare, Terung Ungu, dan Tomat, ditambah 131 gambar sayur dari jenis yang berbeda-beda, ditambah lagi 291 gambar selain sayuran. Untuk mengambil ratusan gambar sekaligus, gunakan Fatkun Batch Unduh Gambar. Tampilan awal dari website Fatkun Batch Unduh Gambar dapat dilihat pada Gambar 4.



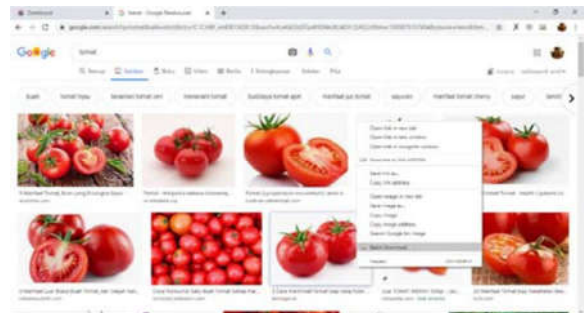
Gambar 4. Tampilan Awal Fatkun Batch Unduh Gambar

Setelah masuk ke website Fatkun Batch Unduh Gambar, ditekan *button* Tambahkan ke Chrome agar fungsi dari Fatkun Batch Unduh Gambar dapat muncul di Google Chrome. Proses tersebut dapat dilihat pada Gambar 5.



Gambar 5. Menambahkan Fatkun Batch Unduh Gambar ke Google Chrome

Setelah proses menambahkan Fatkun Batch Unduh Gambar selesai, penulis mencari gambar yang akan dijadikan dataset pada Google. Untuk menggunakan fungsi Fatkun Batch Unduh Gambar, tab Gambar pada Google dibuka lalu diklik kanan pada gambar yang dipilih, seperti pada Gambar 6. Kemudian penulis memilih tulisan Batch Download. Halaman akan berpindah ke halaman Fatkun Batch Unduh Gambar sehingga penulis dapat memilih banyak gambar yang akan diunduh.



Gambar 6. Mencari Gambar Sayur Tomat

2. Melatih Dataset

Tensor flow sudah menyediakan *source code* untuk melakukan pelatihan ulang terhadap model yang sudah ada di www.tfhub.dev. *Source code* tersebut dapat diunduh melalui situs github milik tensor flow dengan perintah seperti Gambar 8.

```
(em) C:\tensorflow\src\1.10.0> https://github.com/tensorflow/hub/raw/master/examples/image_retraining/retrain.py
% Total % Received % Xferd Average Speed Time Time Current
100 158 100 158 0 0 39 0 0:00:04 0:00:04 --:--:-- 30
100 55947 100 55947 0 0 11389 0 0:00:05 0:00:05 --:--:-- 109K
```

Gambar 8. Mengambil Source Code Python

Setelah mengunduh *source code*, dataset yang sudah dikumpulkan dilatih dengan model mobilenet_v2_140_224 yang memiliki kecepatan deteksi sebesar 117 ms. Gambar 9 memperlihatkan perintah dalam melatih model dengan menggunakan tensor flow-hub.

```
(em) C:\tensorflow\src\1.10.0> python retrain.py --output_graph c:\tensorflow\src\1.10.0\tf_files\retrained_graph_coba.pb --output_labels c:\tensorflow\src\1.10.0\tf_files\retrained_label_coba.txt --img_dir c:\tensorflow\src\1.10.0\tf_files\dataset --tfhub_module https://tfhub.dev/google/mobilenet_v2_140_224/classification/1
INFO:tensorflow:Looking for images in 'Brokoli'
INFO:tensorflow:Looking for images in 'Brokoli'
INFO:tensorflow:Looking for images in 'Bukan Sayur'
INFO:tensorflow:Looking for images in 'Bukan Sayur'
INFO:tensorflow:Looking for images in 'Daging'
INFO:tensorflow:Looking for images in 'Kacang-Panjang'
INFO:tensorflow:Looking for images in 'Kacang-Panjang'
INFO:tensorflow:Looking for images in 'Kubis'
INFO:tensorflow:Looking for images in 'Kubis'
INFO:tensorflow:Looking for images in 'Others'
INFO:tensorflow:Looking for images in 'Others'
INFO:tensorflow:Looking for images in 'Papa'
INFO:tensorflow:Looking for images in 'Papa'
INFO:tensorflow:Looking for images in 'Papa'
INFO:tensorflow:Looking for images in 'Terong-tingpi'
INFO:tensorflow:Looking for images in 'Terong-tingpi'
INFO:tensorflow:Looking for images in 'Tomat'
INFO:tensorflow:Looking for images in 'Tomat'
```

Gambar 9. Melatih Dataset Sayuran

Berdasarkan perintah, untuk melakukan proses melatih model digunakan perintah Python dengan menjalankan *source code* retrain.py sesuai dengan lokasi direktori dataset dan menggunakan model mobilenet_v2_140_224 yang dijalankan dengan URL tensor flow-hub. Selain model, akan didapat juga label dari dataset yang ikut diolah bersama model.

Model tersebut dilatih menggunakan dataset dengan total dataset 941 objek sayur selama 10.000 kali. Sehingga menghasilkan persentase nilai akurasi dari data yang dilatih (*Train Accuracy*), data yang diuji (*Validation Accuracy*) dan juga nilai dari data yang hilang (*Cross Entropy*). Gambar 10 menunjukkan potongan proses pelatihan dataset yang menampilkan akurasi data yang dilatih dan data yang diuji, juga nilai data yang hilang.

```
INFO:tensorflow:2020-12-19 13:10:57.948929: Step 2940: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:10:57.948929: Step 2940: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:10:58.873652: Step 2950: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:10:58.873652: Step 2950: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:10:58.874651: Step 2950: Cross entropy = 0.006450
INFO:tensorflow:2020-12-19 13:10:58.967621: Step 2950: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:10:58.967621: Step 2950: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:10:59.883312: Step 2960: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:10:59.883312: Step 2960: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:10:59.884314: Step 2960: Cross entropy = 0.005999
INFO:tensorflow:2020-12-19 13:10:59.884314: Step 2960: Cross entropy = 0.005999
INFO:tensorflow:2020-12-19 13:10:59.977377: Step 2960: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:10:59.977377: Step 2960: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:00.918062: Step 2970: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:00.918062: Step 2970: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:00.918065: Step 2970: Cross entropy = 0.008809
INFO:tensorflow:2020-12-19 13:11:00.918065: Step 2970: Cross entropy = 0.008809
INFO:tensorflow:2020-12-19 13:11:01.816029: Step 2970: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:01.816029: Step 2970: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:01.928743: Step 2980: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:01.928743: Step 2980: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:01.921741: Step 2980: Cross entropy = 0.006399
INFO:tensorflow:2020-12-19 13:11:01.921741: Step 2980: Cross entropy = 0.006399
INFO:tensorflow:2020-12-19 13:11:02.014728: Step 2980: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:02.014728: Step 2980: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:02.937416: Step 2990: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:02.937416: Step 2990: Train accuracy = 100.0%
INFO:tensorflow:2020-12-19 13:11:02.939419: Step 2990: Cross entropy = 0.007078
INFO:tensorflow:2020-12-19 13:11:02.939419: Step 2990: Cross entropy = 0.007078
INFO:tensorflow:2020-12-19 13:11:03.035462: Step 2990: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2020-12-19 13:11:03.035462: Step 2990: Validation accuracy = 100.0% (N=100)
```

Gambar 10. Potongan Proses Pelatihan Dataset

Setelah proses pelatihan model selesai, akan ditampilkan persentase nilai akurasi sebesar 98.1% dan disimpan dengan nama retrained_graph_coba.pb, sedangkan label hasil pelatihan disimpan dengan nama retrained_label.txt seperti terlihat pada Gambar 11.

```
Use standard file APIs to check for files with this prefix.
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Final test accuracy = 98.1% (N=156)
INFO:tensorflow:Final test accuracy = 98.1% (N=156)
INFO:tensorflow:Save final result to : c:\tensorflow\src\1.10.0\tf_files\retrained_graph_coba.pb
INFO:tensorflow:Save final result to : c:\tensorflow\src\1.10.0\tf_files\retrained_graph_coba.pb
```

Gambar 11. Hasil Melatih Dataset Sayuran

Hasil dari proses pelatihan model dapat dilihat secara grafik menggunakan tensor board. Perintah untuk melihat hasil pelatihan model dapat dilihat pada Gambar 12.

```
(em) C:\tensorflow\src\1.10.0> tensorboard --logdir c:\tmp\retrain_logs --host=127.0.0.1
TensorBoard 1.13.1 at http://127.0.0.1:6006 (Press CTRL+C to quit)
```

Gambar 12. Membuat Akses ke Tensor board

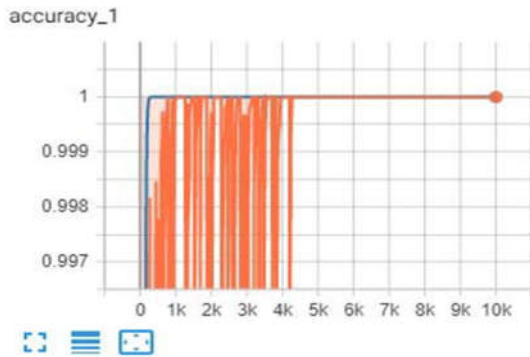
Setelah itu, buka link tersebut pada browser untuk menampilkan tensor board. Gambar 13 merupakan tampilan grafik hasil dari melatih model.



Gambar 13. Hasil Pelatihan Berupa Grafik pada Tensor board

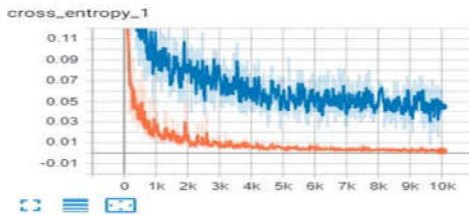
Pada Gambar 14, terlihat bahwa garis berwarna oranye bergerak dari tengah ke atas, lalu stabil. Hal ini berarti bahwa semakin banyak langkah dalam pelatihan model, nilai akurasi pada data yang dilatih semakin mendekati 1. Garis berwarna biru bergerak dari tengah ke arah bawah, yang berarti bahwa

semakin banyak langkah, nilai akurasi data yang diuji semakin kecil.



Gambar 14. Grafik Akurasi

Sedangkan pada Gambar 15, menunjukkan hasil yang berbanding terbalik dengan Gambar 14. Hal ini berarti bahwa nilai yang hilang dari data yang dilatih semakin lama semakin kecil, sedangkan nilai yang hilang dari data yang diuji semakin lama semakin besar.



Gambar 15. Grafik Nilai yang Hilang






Berdasarkan kedua grafik yang ditampilkan pada Gambar 14 dan Gambar 15, maka dapat disimpulkan bahwa model telah cukup baik dalam mempelajari data yang dilatih (data train).





Pada tahap ini, dilakukan proses *scanning* (memindai) terhadap objek untuk mengetahui nilai akurasi yang ditampilkan oleh aplikasi dan menghasilkan model.

Apabila yang dihasilkan oleh aplikasi sudah sesuai dengan yang diharapkan, maka model yang digunakan pada aplikasi sudah benar. Akan tetapi, apabila yang dihasilkan tidak sesuai dengan harapan, maka dapat dipastikan bahwa masih terdapat kesalahan pada model yang dilatih.

Hasil dari uji coba akurasi disajikan di dalam bentuk tabel yang dapat dilihat pada Tabel 3. Dengan jumlah data uji sebanyak 7 sayuran, diantara kubis, jagung, brokoli, kacang Panjang, terung ungu, tomat dan pare. Selain itu terdapat gambar yang bukan termasuk dalam klasifikasi sayuran diantaranya *others* dan bukan sayur.

Tabel 3. Hasil Uji coba

Objek Asli	Prediksi dan Akurasi Model	Kesimpulan
Brokoli	Prediksi: Brokoli 97.237114% Kacang Panjang 1.9900851% Kubis 0.3928532% Hasil Tangkap Layar: 	Akurat
Jagung	Prediksi: Jagung 99.98049% Pare 0.00709833% Others 0.0059523694% Hasil Tangkap Layar: 	Akurat
Kacang Panjang	Prediksi : Kacang Panjang 99.10925% Pare 0.8837433% Terung Ungu 0.005559233% Hasil Tangkap Layar : 	Akurat
Pare	Prediksi: Pare 94.945% Kacang Panjang 4.936% Terung Ungu 0.1168% Hasil Tangkap Layar: 	Akurat
Terung Ungu	Prediksi: Terung Ungu 97.38% Kacang Panjang 1.833% Pare 0.407% Hasil Tangkap Layar: 	Akurat

Tomat	Prediksi: Tomat 99.826% Kubis 0.0914% Pare 0.0279% Hasil Tangkap Layar: 	Akurat
Kubis	Prediksi: Kubis 99.71% Others 0.20% Pare 0.045% Hasil Tangkap Layar: 	Akurat
Others	Prediksi: Others 96.06% Terung Ungu 3.61% Jagung 0.234% Hasil Tangkap Layar: 	Akurat
Bukan Sayur	Prediksi: Bukan Sayur 99.94% Kubis 0.028% Brokoli 0.0157% Hasil Tangkap Layar: 	Akurat

5. KESIMPULAN

Beberapa kesimpulan yang dapat diambil pada penelitian ini berdasarkan pengujian aplikasi untuk mengenali jenis sayuran berbasis android menggunakan metode *Convolutional Neural Network* Jumlah data set yang digunakan sebagai data training sebanyak 941 objek sayuran dengan data uji sebanyak 7 objek sayuran. Dari 7 data uji tersebut semua berhasil diklasifikasikan. Metode *Convolutional Neural Network* mampu melakukan klasifikasi sayur. Hasil dari *final test accuracy* yang diperoleh dari keseluruhan proses pelatihan yaitu sebesar 98,1%. Terdapat 3 hasil klasifikasi pengujian dengan akurasi tertinggi yaitu klasifikasi jenis sayur jagung dengan akurasi sebesar 99.98049%, tomat dengan akurasi sebesar 99,82574% dan klasifikasi jenis sayur kubis dengan akurasi sebesar 99.712364%.

6. SARAN

Dalam penelitian ini yang dihasilkan adalah sebuah aplikasi *mobile* yang mengklasifikasikan 7 jenis sayuran menggunakan algoritme *Convolution Neural Network* (CNN). Dataset yang digunakan masih menggunakan citra gambar. Kesempatan untuk pengembangan klasifikasi sayuran ini dapat dilakukan dengan menerapkan klasifikasi sayuran ini pada media video selain gambar. Tantangan yang akan dihadapi adalah melakukan proses pemotongan gambar pada video tersebut agar objek gambar tersebut dapat diklasifikasikan atau dikenali. Selain itu jumlah data latih dan data uji yang dibutuhkan pun harus ditingkatkan. Dengan mengembangkan penelitian ini maka membutuhkan perangkat keras komputer dengan spesifikasi yang lebih baik. Penggunaan perangkat *Graphical Processing Unit* (GPU) dan peningkatan *Random Access Memory* (RAM) diharapkan dapat membantu dalam pengembangan penelitian ini.

7. DAFTAR PUSTAKA

- Hansun, S., Kristanda, M. B. & Saputra, M. W., 2018. Pemrograman Android Dengan Android Studio IDE. Yogyakarta: Andi.
- Johansson, R., 2019. Numerical Python Second Edition. APress Media.
- Kurniadi, A., Kusri, Sadikin, 2020 M F., Implementasi Convolution Neural Network untuk klasifikasi Varietas pada citra daun sawi menggunakan Keras. *Journal of Computer Information Technology* (4), 25-33. Agustus 2020.
- LeCun, Y., Bengio, Y., & Hinton, G. 2015. Deep learning. *nature*, 521(7553), 436-444..
- Nuraini, F., 2018. Buku Ajar Pengetahuan Bahan Nabati I: Sayur-sayuran, Buah-buahan, Kacang- kacang, Serelia dan Umbi-umbian. Lampung: Universitas Lampung.
- Om, P., & Vijay, G. (2018). Classification of vegetables using Tensor Flow. *International Journal for Research in Applied Science and Engineering Technology*, 6(4), 2926-2934.
- Peters, J. F., 2017. Foundation of Computer Visions. Springer Publisher. New York.
- Primartha, R. 2018. Belajar Machine Learning – Teori dan Praktik. Palembang. Informatika Bandung.
- Widodo, B. 2016. Machine Learning Dan Computational Intelligence.
- Zhu, L., Li, Z., Li, C., Wu, J., & Yue, J. 2018. High performance vegetable classification from images based on alexnet deep learning model. *International Journal of Agricultural and Biological Engineering*, 11(4), 217-223.