

IDENTIFIKASI PENYAKIT PADA DAUN TOMAT DAN DAUN SINGKONG BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID

DISEASES IDENTIFICATION OF TOMATO LEAVES AND CASSAVA LEAVES IMAGE-BASED USING CONVOLUTIONAL ARTIFICIAL NEURAL NETWORK (CNN) METHOD ON ANDROID

Jepri¹, Imron Rosyadi, S.T., M.Sc.², Farida Asriani, S.Si., M.T.³

jepri.tugas@gmail.com¹, pak.imron@gmail.com², faridapamuji@gmail.com²

¹Mahasiswa Pemakalah

²Dosen Pembimbing I

³Dosen Pembimbing II

^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

Abstrak— Pertanian memiliki arti penting dalam pembangunan perekonomian bangsa. Pemerintah telah menetapkan pertanian sebagai prioritas utama pembangunan di masa mendatang. Sektor pertanian tidak saja sebagai penyedia kebutuhan pangan bagi penduduknya, tetapi juga merupakan sumber pendapatan ekspor (devisa) serta pendorong dan penarik bagi tumbuhnya sektor-sektor ekonomi lainnya. Misalnya pada produksi tanaman tomat dan singkong yang memiliki penjualan yang tinggi. Salah satu penyebab utama penurunan produksi hasil tersebut yaitu munculnya berbagai macam penyakit. Identifikasi penyakit tanaman biasanya dilakukan di laboratorium. Dalam hal ini penulis membuat penelitian tentang identifikasi penyakit tanaman menggunakan *deep learning* berbasis android agar mudah digunakan menggunakan *Smartphone*. Cara penggunaannya hanya dengan memasukan obyek yang akan diidentifikasi kedalam aplikasi *Plant Diseases* melalui media pengambilan gambar secara langsung menggunakan kamera atau *import gallery*. Berdasarkan hasil pelatihan dan pengujian yang dilakukan menggunakan tiga arsitektur (VGG16, *InceptionV3*, dan *mobileNet*) proses identifikasi menggunakan arsitektur *mobileNet* memiliki akurasi yang lebih tinggi, *mobileNet* memiliki akurasi sebesar 95,33%, arsitektur *Inception* 92,67%, dan VGG16 76,67%. Selain itu *mobileNet* juga memiliki respon deteksi yang lebih cepat yaitu memiliki respon deteksi rata-rata sebesar 134,66, sedangkan pada arsitektur *Inception* dan VGG16 masing masing sebesar 643,6 dan 789,03.

Kata kunci — Penyakit Tanaman, Deep Learning, CNN.

Abstract— Agriculture has an important role in the economic development of the nation. The government has set agriculture as priority development in the future. Agriculture not only as a provider of food for the population, but also a source of export revenue (foreign exchange), and push and pull of the growth sectors of the economy. For example in the production of tomatoes and potatoes that have high sales. One of the main causes of the decrease in production is the result of the emergence of various diseases. Identification of plant diseases is usually done in a laboratory. In this case the authors make research on the identification of plant diseases using Android-based deep learning so that it is easy to use using smartphones. How to use it only by entering the object that will be identified into the application *Plant Diseases* through media shooting directly using a camera or *import gallery*. Based on the results of training and testing conducted using three architectures (VGG16, *InceptionV3*, and *mobileNet*) the identification process using *mobileNet* architecture has higher accuracy, *mobileNet* has an accuracy of 95.33%, *Inception* architecture 92.67%, and VGG16 76.67 %. In addition, *mobileNet* also has a faster detection response, which has an average detection response of 134.66, while the *Inception* and VGG16 architectures are 643.6 and 789.03, respectively.

Keywords — Plant disease, Deep Learning, CNN.

I. PENDAHULUAN

A. Latar Belakang

Pertanian memiliki arti penting dalam pembangunan perekonomian bangsa. Pemerintah telah menetapkan pertanian sebagai prioritas utama pembangunan di masa mendatang. Sektor pertanian tidak saja sebagai penyedia kebutuhan pangan bagi penduduknya, tetapi juga merupakan sumber pendapatan ekspor (devisa) serta pendorong dan penarik bagi tumbuhnya sektor-sektor ekonomi lainnya. Pembangunan pertanian yang dikelola dengan baik dan bijak akan dapat meningkatkan pertumbuhan dan sekaligus pemerataan ekonomi secara berkelanjutan, mengatasi kemiskinan dan pengangguran yang pada akhirnya menyejahterakan masyarakat Indonesia secara keseluruhan [1].

Secara umum, sektor pertanian terdiri dari beberapa sub sektor, yaitu sub sektor pangan, hortikultura, dan perkebunan. Salah satu sub sektor pertanian yang cukup penting dan merupakan salah satu penyumbang peningkatan PDB Indonesia adalah sub sektor hortikultura (buah dan sayuran). Tanaman tomat (*Lycopersicum esculentum* Mill.) merupakan salah satu komoditas sayuran yang berpotensi multiguna. Selain itu komoditas tomat juga salah satu komoditas sayuran yang menjadi penyumbang ekspor selain kol, wortel dan kentang [2].

Selain tomat, komoditas pertanian pangan yang mempunyai prospek untuk dikembangkan dalam rangka memenuhi kebutuhan pasar domestik maupun internasional adalah singkong. Singkong (*Manihot utilisima*) merupakan makanan pokok ketiga setelah padi dan jagung bagi masyarakat Indonesia. Pada tahun 2011 produksi singkong di Indonesia mencapai 24.044.025 ton, sedangkan pada tahun 2012 meningkat menjadi 24.177.327 ton [3].

Salah satu penyebab utama penurunan produksi hasil tersebut yaitu munculnya berbagai macam penyakit. Tanaman dikatakan sakit bila ada perubahan seluruh atau sebagian organ-organ tanaman yang menyebabkan terganggunya kegiatan fisiologis sehari-hari.

Secara singkat, penyakit tanaman adalah penyimpangan dari keadaan normal. Penyebab sakitnya tanaman bermacam-macam. Ada yang disebabkan oleh cendawan, bakteri, virus, dan lain-lain [4].

Identifikasi penyakit tanaman biasanya dilakukan di laboratorium. Dalam hal ini penulis membuat penelitian tentang identifikasi penyakit tanaman menggunakan *deep learning*.

Deep learning adalah sub bidang khusus dari pembelajaran mesin (*machine learning*) yang merupakan pandangan baru tentang representasi pembelajaran dari data yang menekankan pada lapisan-lapisan (layer) pembelajaran berturut turut dari representasi yang semakin berarti [5]. *Deep learning* banyak digunakan untuk klasifikasi obyek berdasarkan citra atau gambar dengan cara mempelajari representasi atau fitur data secara otomatis.

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image [6].

Penanganan terhadap penyakit tanaman harus segera dilakukan. Oleh karena itu, identifikasi penyakit tanaman menggunakan metode *Deep Learning* sangat diperlukan sebagai langkah awal untuk mengetahui penyakit. Untuk mempermudah melakukan identifikasi, maka *deep Learning* dibuat dalam aplikasi android yang dapat di *install* pada *Smartphone*.

Pada Tugas Akhir ini penulis mengambil topik identifikasi penyakit tanaman. Namun, tanaman yang di ambil untuk di jadikan bahan uji baru hanya penyakit tanaman pada daun tomat dan daun Singkong. Sehingga diperoleh judul “IDENTIFIKASI PENYAKIT PADA DAUN TOMAT DAN DAUN SINGKONG BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS ANDROID”.

B. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang diajukan adalah:

1. Bagaimana perancangan arsitektur CNN untuk identifikasi penyakit tanaman?
2. Bagaimana pelatihan dan pengujian untuk identifikasi dengan metode CNN?
3. Bagaimana pengujian identifikasi pada aplikasi Android?

C. Batasan Masalah

Pada penelitian tugas akhir yang dilakukan memiliki batasan masalah sebagai berikut :

1. Identifikasi Penyakit tanaman hanya meliputi sepuluh penyakit tanaman pada daun tomat dan lima penyakit pada daun singkong.
2. Metode *deep learning* yang digunakan adalah CNN (*Convolutional Neural Network*).
3. Menggunakan Bahasa Pemrograman *Python*
4. Infrastruktur yang digunakan adalah Google *Colaboratory*.
5. Menggunakan *Framework* Keras.
6. Aplikasi dibuat menggunakan Android Studio untuk *smartphone*.

D. Tujuan dan Manfaat

a. Tujuan

Maksud penulis melaksanakan tugas akhir ini adalah sebagai berikut :

1. Merancang arsitektur CNN untuk identifikasi penyakit tanaman pada daun tomat dan daun singkong dan penerapannya pada aplikasi Android.
2. Melatih arsitektur CNN untuk identifikasi penyakit tanaman.
3. Menguji arsitektur CNN identifikasi penyakit tanaman yang sudah dilatih sebelumnya..

b. Manfaat

Manfaat yang diharapkan dalam pelaksanaan kerja praktik ini adalah sebagai berikut :

1. Bagi Mahasiswa Mampu menerapkan ilmu yang didapat pada mata kuliah yang bersangkutan untuk menyelesaikan tugas akhir.
2. Memudahkan petani dalam mengidentifikasi penyakit tanaman sehingga bisa dilakukan penanganan.

II. TINJAUAN PUSTAKA

A. Pengolahan Citra

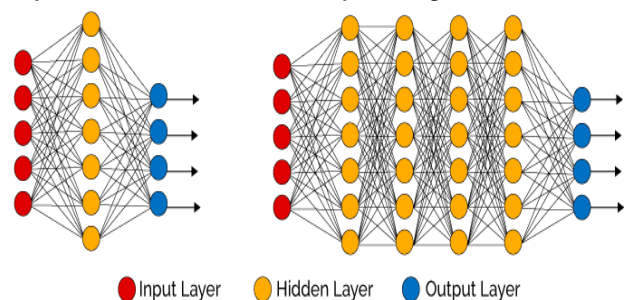
Pengolahan Citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra yang berkualitas lebih baik daripada citra masukan [7].

B. Deep Learning

Deep Learning (Pembelajaran Dalam) atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam [8].

Simple Neural Network

Deep Learning Neural Network



Gambar- 1 Perbedaan antara lapisan layer pada Jaringan Saraf Tiruan dengan Jaringan *Deep Learning*

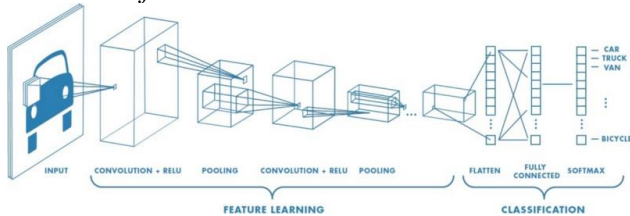
Pada gambar 2, jaringan mengubah gambar digit menjadi representasi yang semakin berbeda dari gambar asli dan semakin informatif tentang hasil akhir.

C. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*. CNN bisa

digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image* [6].

Secara garis besar CNN tidak jauh beda dengan *neural network* biasanya. CNN terdiri dari neuron yang memiliki *weight*, bias dan *activation function*.



Gambar- 2 Alur proses CNN dalam mengolah citra

D. Google Colaboratory

Google Colaboratory adalah salah satu produk Google berbasis *cloud* yang bisa kita gunakan secara gratis. Perbedaannya adalah Google Colab dibuat khusus untuk para *programmer* atau *researcher* yang mungkin kesulitan untuk mendapatkan akses komputer dengan spek tinggi. Google Colab adalah *coding environment* bahasa pemrograman *Python* dengan format “*notebook*” (mirip dengan *Jupyter notebook*), atau dengan kata lain Google seakan meminjamkan kita komputer secara gratis! untuk membuat program oleh Google [11].

E. Framework Keras

Saat ini kita mengenal cukup banyak *framework* untuk penggunaan *deep learning*. Beberapa yang cukup populer di antaranya adalah: *Tensorflow*, *Keras*, *MXNet*, dan *PyTorch*.

Pada analisis ini, saya hanya akan menggunakan *framework* *Keras* dengan bantuan *R interface* dari *RStudio*. *Keras* adalah *high-level neural network API* yang dikembangkan dengan *Python* dengan fokus tujuan untuk mempercepat proses riset atau percobaan [12]. Beberapa fitur utama dari *Keras*:

1. Mampu menjalankan *source code* yang sama menggunakan CPU atau GPU dengan lancar.
2. API yang *user-friendly* sehingga mempermudah penggunaanya dalam proses prototipe model *deep learning*.
3. Dukungan *built-in* untuk CNN atau *Convolutional Neural Networks* (*Computer*

Vision), RNN atau *Recurrent Neural Networks* (untuk *sequence processing*), dan kombinasi keduanya.

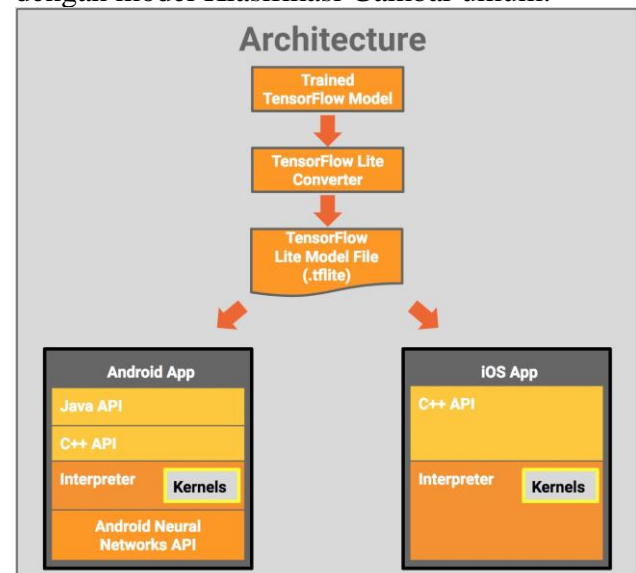
4. Dapat digunakan untuk hampir semua jenis dari model *deep learning*.

F. TensorFlow Lite

TensorFlow Lite adalah solusi mudah untuk perangkat seluler dan *embedded devices*. Ini memungkinkan untuk menjalankan model yang dipelajari mesin pada perangkat seluler dengan latensi rendah, sehingga dapat memanfaatkannya untuk melakukan klasifikasi, regresi, atau apa pun yang mungkin Anda inginkan tanpa harus melakukan perjalanan bolak-balik ke server [13].

TensorFlow Lite terdiri dari *runtime* tempat Anda dapat menjalankan model yang sudah ada sebelumnya, dan seperangkat alat yang dapat Anda gunakan untuk mempersiapkan model Anda untuk digunakan pada perangkat seluler dan *embedded devices*.

TensorFlow Lite saat ini dalam pratinjau pengembang, sehingga mungkin tidak mendukung semua operasi di semua model *TensorFlow*. Meskipun demikian, ini berfungsi dengan model Klasifikasi Gambar umum.



Gambar- 11 Arsitektur *TensorFlow Lite*

G. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk

pengembangan aplikasi Android, yang didasarkan pada Intelijen IDEA [14].

H. Penyakit Pada Tumbuhan

Penyakit tanaman dapat diartikan gangguan terhadap tanaman yang disebabkan oleh *pathogen* dan non *pathogen* yang menyebabkan terganggunya proses pertumbuhan pada bagian-bagian tertentu dari tanaman yang tidak dapat berjalan sesuai fungsinya dengan normal dan dengan baik sehingga menghambat pertumbuhan pada tanaman [15].

a. Penyakit Daun Tomat

Tanaman tomat sangat rentan terhadap serangan penyakit. Jika lingkungan sekitar tanaman tomat tidak bersih maka patogen seperti jamur, bakteri atau virus akan sangat mudah untuk menyerang tanaman tomat [16].

1. Bacterial Spot

Penyakit ini disebabkan oleh bakteri *Xanthomonas vesicatoria*, yang menyerang tomat hijau tetapi tidak pada tomat merah [17].



Gambar- 12 Bacterial Spot

2. Early Blight (Bercak Daun)

Pada daun terdapat bercak kecil berwarna coklat yang dapat meluas dan menyebabkan daun berlubang. Bercak dicirikan dengan adanya lingkaran yang konsentris. Penyebab bercak daun ini adalah karena serangan jamur *Alternaria solani* [16]. Jamur ini biasanya menyerang di saat musim hujan.



Gambar-13 Early Blight

3. Late Blight (Busuk daun)

Pada daun muncul bercak coklat dengan tepi berwarna kuning. Di bagian bawah bercak akan terlihat adanya lapisan putih seperti beludru. Bercak ini kemudian meluas ke seluruh daun.

Penyakit ini disebabkan oleh jamur *Phytophthora infestans*. Penyebarannya sangat aktif di musim hujan, terutama pada suhu lingkungan 18-20 °C dan kelembapan 91-100%.



Gambar- 141 Late Blight

4. Leaf Mold (Jamur Daun)

Jamur *Passalora fulva* menyebabkan jamur daun. Ini pertama kali diamati pada daun yang lebih tua di dekat tanah di mana gerakan udara buruk dan kelembaban tinggi [17].



Gambar-15 *Leaf Mold*

5. *Mosaic Virus*

Gejala infeksi virus dapat muncul sebagai bercak daun hijau terang dan gelap. *Tobacco mosaic virus* (TMV) menyebabkan bercak daun yang lebih tua dan dapat menyebabkan malformasi selebaran, yang dapat berbentuk seperti tali sepatu.



Gambar- 16 *Mosaic Virus*

6. *Septoria Leaf Spot* (Bintik Daun Septoria)

Penyakit merusak daun tomat, daun dan batang (buah tidak terinfeksi) ini disebabkan oleh jamur *Septoria lycopersici* [17].



Gambar-17 *Septoria Leaf Spot*

7. *Two Spotted Spider Mite*

Two Spotted Spider Mite adalah spesies tungau paling umum yang menyerang tanaman sayuran dan buah-buahan di New England [19].



Gambar-18 *Two Spotted Spider Mite*

8. *Target Spot* (Bintik target mentimun)

Banyak bintik bulat berwarna krem, hingga diameter 4 mm; sering kali, berbentuk tidak teratur atau bersudut, dibatasi oleh pembuluh darah. Daun mengering dan rontok sebelum waktunya [18].



Gambar-19 *Target Spot*

9. *Yellow Leaf Curl Virus*

Daun muda yang terinfeksi virus ini akan terlihat mengerut atau keriting dan berwarna

kuning. Vektor utama virus ini adalah kutu kebul [16].



Gambar-20 *Yellow Leaf Curl Virus*

b. Penyakit Tanaman Pada Daun Singkong

Gangguan hama, penyakit, dan gulma merupakan masalah yang dihadapi petani dalam budidaya ubi kayu karena selain dapat menurunkan hasil, juga dapat mengakibatkan penurunan kualitas ubi [19]. Berikut beberapa penyakit pada daun tanaman ubi :

1. *Cassava Bacterial Blight (CBB)*

CBB disebabkan oleh *Xanthomonas campestris* pv. *Manihotis* merupakan penyakit bakteri terpenting pada ubi kayu [19]. Gejala awal berupa lesi berwarna abu-abu mirip bekas tersiram air panas. Lesi dibatasi oleh tulang-tulang daun sehingga terbentuk lesi menyudut, terlihat lebih jelas pada sisi bawah daun.



Gambar-21 *Cassava Bacterial Blight*

2. *Cassava Brown Streak Disease (CBSD)*

CBSD yang diketahui disebabkan oleh *Cassava Brown Streak Virus* (CBSV). Gejala infeksi virus bergaris coklat ubi kayu sangat

bervariasi antar varietas dan antar musim yang berbeda. Gejala infeksi CBSV dapat diamati pada daun, batang, buah, dan umbi [20]. Pada varietas yang peka, gejala dapat diamati pada semua bagian tanaman tersebut, sementara pada varietas yang toleran umumnya gejala hanya dapat diamati pada salah satu organ tanaman umumnya daun.



Gambar-22 *Cassava Brown Streak Disease*

3. *Cassava Green Mottle (CGM)*

Gejala tanaman ubi kayu yang terinfeksi Virus belang hijau ubi kayu adalah pada daun yang muncul menunjukkan gejala belang sistemis dengan nekrosis, namun pada daun-daun berikutnya tidak menunjukkan gejala meskipun mengandung virus [19].



Gambar-23 *Cassava Green Mottle*

4. *Cassava Mosaic Disease (CMD)*

Penyakit ini disebabkan oleh spesies virus dalam genus *Begomovirus*. Tingkat paling parah pada *cassava Mosaic Disease* dipengaruhi oleh faktor lingkungan seperti intensitas cahaya, angin, curah hujan, kepadatan tanaman dan suhu. Mengingat bahwa virus ditularkan oleh kutu

kebul, penyebaran virus akan sangat tergantung pada vektor [19].



Gambar 24 *Cassava Mosaic Disease*

III. METODE PENELITIAN

A. Waktu dan Tempat Penelitian

Penelitian dilaksanakan dalam waktu 4 bulan dimulai dari bulan Oktober 2019 hingga bulan Januari 2020 bertempat di Kampus Fakultas Teknik Universitas Jenderal Soedirman, Kabupaten Purbalingga.

B. Alat dan Bahan

Dalam penelitian ini, daftar alat dan bahan yang digunakan selama penelitian sebagai berikut.

1. Perangkat keras:

Perangkat keras yang digunakan adalah sebagai berikut :

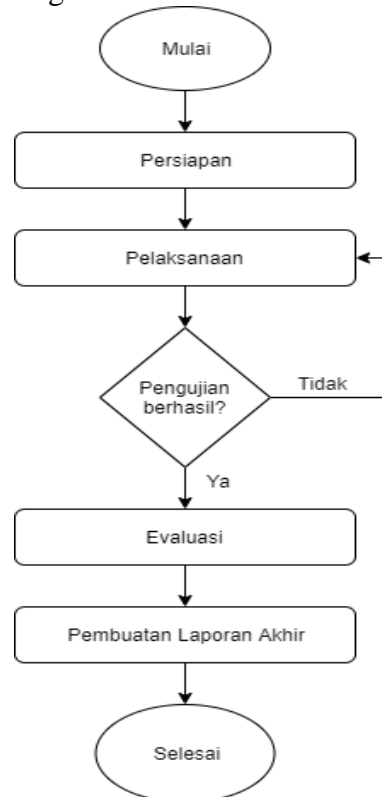
- a. Laptop Lenovo B41-35 dengan spesifikasi processor AMD A8 7410 RAM 4 GB.
- b. Smartphone OPPO A37fw dengan spesifikasi prosesor Snapdragon 410 dan RAM 2 GB.

2. Perangkat lunak:

- a. Sistem Operasi *Windows* 10 64 bit
- b. Peramban Internet *Google Chrome* versi 77.0.3865.120 64 bit
- c. *Google Colaboratory* (*Jupyter Notebook* versi *cloud*).
- d. *Android Studio* versi 3.5.0.0.
- e. Sistem Operasi *Android Lollipop* 5.1.
- f. Layanan Repositori Web *Development* pada Platform *Github*.

C. Metode Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan sebagai berikut.



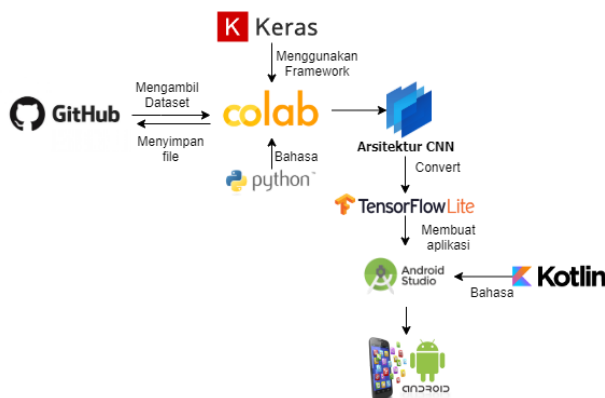
Gambar- 2 Desain arsitektur sistem

a. Persiapan

Pada tahap ini penulis melakukan studi pustaka melalui *website*, jurnal, buku, dan melakukan bimbingan langsung dengan dosen pembimbing. Penulis mencari materi-materi yang berkaitan dengan *deep learning*, *Convolutional Neural Network* (CNN), dan pengaplikasian dalam *smartphone*.

b. Pelaksanaan

Pada tahap ini penulis melakukan persiapan dan *preprocess*, mendesain arsitektur yang dipakai, dan melakukan pengujian.



Gambar -25 Alur Pelaksanaan

1. Persiapan dan *Preprocess*

Pada tahap ini penulis melakukan pengumpulan data set dan menyiapkan beberapa penelitian sebelumnya sebagai panduan untuk melakukan penelitian ini. Dalam hal ini penulis memperoleh data set yang telah disediakan oleh Github.

2. Desain Arsitektur

Pada tahap ini penulis membuat model *deep Learning* menggunakan *framework* keras dengan Google *Colaboratory* sebagai infrastrukturnya. Diawali dengan pengambilan data set pada Github, kemudian membuat model CNN (*Convolutional Neural Network*) dengan arsitektur yang berbeda-beda : VGG16, MobileNet V1, dan ResNet.

Setelah model terbentuk data set dilatih agar dapat mengidentifikasi penyakit tanaman pada daun tomat dan daun singkong mendapat data hasil pelatihan berupa *file* h5. Kemudian data pelatihan di *convert* menjadi *TensorFlow Lite* agar memudahkan dalam pemodelan aplikasi di android. Setelah di *convert*, lalu dilakukan pemodelan aplikasi menggunakan Android studio.

3. Tahap Pengujian

Tahap ini penulis melakukan pengujian keseluruhan terhadap sistem yang telah dirancang. Sistem akan diuji apakah sudah sesuai yang diharapkan atau belum. Sistem dianggap sudah baik jika nilai dari kesalahan atau *error (loss)* semakin mendekati 0. Jika tidak sesuai maka peneliti melakukan

pembenahan dan pengkajian ulang terhadap berbagai hal yang timbul selama pengujian.

c. Evaluasi

Pada tahap ini penulis melakukan evaluasi mengenai hasil dari penelitian. Dari tiga arsitektur yang dipakai akan memperoleh hasil dan tingkat keakuratan yang berbeda. Karena setiap arsitektur memiliki kelebihan dan kekurangan masing-masing.

d. Tahap Akhir

Tahap ini adalah tahapan terakhir yang di mana semua hasil penelitian sudah diperoleh kemudian penulisan laporan penelitian tersebut.

D. Alur Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang dimulai dari tahap persiapan, dilanjutkan dengan tahap pengumpulan data set pelatihan dan data set pengujian. Setelah itu dilakukan tahap perancangan sistem yang dibuat. Kemudian dilakukan tahap pengujian dan evaluasi sistem. Pada tahap terakhir yaitu tahapan pembuatan laporan penelitian.

E. Waktu dan Jadwal Penelitian

Penelitian dilaksanakan dalam waktu 4 bulan dimulai dari bulan Oktober 2019 sampai dengan bulan Januari 2020 dengan rencana jadwal kegiatan sebagai berikut.

Tabel 1 Jadwal Penelitian

No.	Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1.	Persiapan dan Studi Pustaka																
2.	Pelaksanaan																
3.	Evaluasi																
3.	Pembuatan laporan																

IV. Hasil Dan Pembahasan

A. DATASET

1. DATASET DAUN TOMAT

Dataset yang digunakan untuk melakukan proses *training* pendeteksi penyakit pada daun tomat diperoleh dari <https://github.com/spMohanty/PlantVillage-Dataset> milik spMohanty. Adapun jumlah *dataset* yang digunakan dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4.1 Dataset daun Tomat

No	Nama Penyakit	Jumlah
1	Tomat <i>Bacterial spot</i>	1.702
2	Tomat <i>Early blight</i>	1.920
3	Tomat <i>healthy</i>	1.926
4	Tomat <i>Late blight</i>	1.851
5	Tomat <i>Leaf Mold</i>	1.882
6	Tomat <i>Septoria leaf spot</i>	1.745
7	Tomat <i>Spider mites Two spotted spider mite</i>	1.741
8	Tomat <i>Target_Spot</i>	1.827
9	Tomat <i>Mosaic virus</i>	1.790
10	Tomat <i>Yellow Leaf Curl Virus</i>	1.961
Jumlah		18.345

2. DATASET DAUN SINGKONG

Dataset yang digunakan untuk melakukan proses *training* pendeteksi penyakit pada daun singkong diperoleh dari <https://github.com/icassava/fgvcx-icassava>. Adapun jumlah dataset pada masing-masing penyakit dapat dilihat pada tabel 4.2 berikut ini.

Tabel 4.1 Dataset daun Tomat

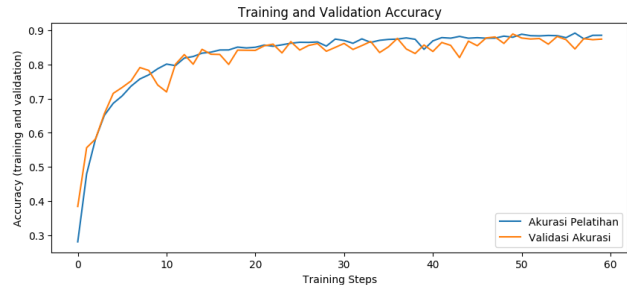
No	Nama Penyakit	Jumlah
1	<i>Cassava Bacteial Blight</i>	466
2	<i>Cassava Brown Streak Disease</i>	1.443
3	<i>Cassava Green Mottle</i>	773
4	<i>Cassava Mosaic Disease</i>	2.658
5	<i>Healty</i>	316
Jumlah		5.656

B. PELATIHAN DAN PENGUJIAN PADA GOOGLE COLABORATORY

1. ARSITEKTUR VGG16

Pelatihan dan pengujian *dataset* menggunakan arsitektur VGG16 menggunakan

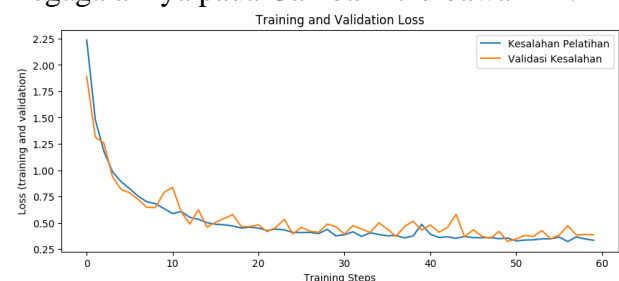
gambar berukuran $64 \times 64 \times 3$ *pixel*. Menggunakan *dataset train* Sebanyak 19.052, *dataset validasi* sebanyak 2.858, dan *dataset test* sebanyak 1.906.



Gambar-26 Kurva Akurasi Training dan Validasi menggunakan Arsitektur VGG16

Pada gambar 26 diatas memperlihatkan kondisi kurva akurasi pelatihan yang mendekati nilai 1 seiring bertambahnya epoch, baik pada *dataset* pelatihan maupun validasinya. Sehingga diperoleh nilai akurasi pada *epoch* terakhir (*epoch* ke-60) yaitu nilai akurasi pelatihan dan akurasi validasi masing-masing adalah 0.8860 dan 0.8747.

Adapun ditampilkan hasil pelatihan pada dataset pelatihan dan validasi dalam keagalannya pada Gambar 27 dibawah ini.

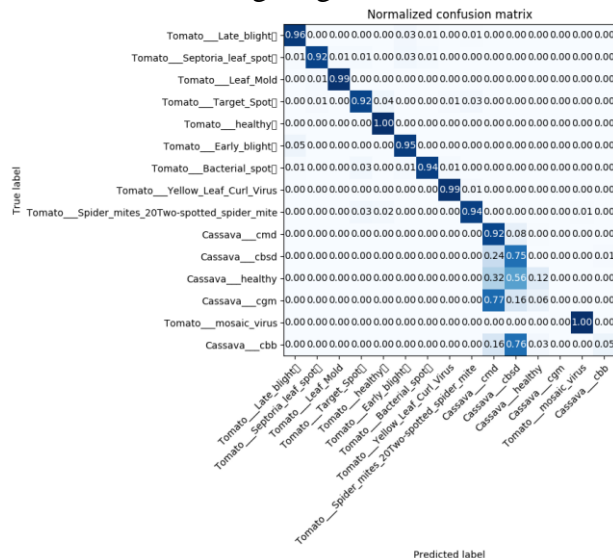


Gambar-27. Kurva Loss training dan validasi menggunakan Arsitektur Vgg16

Pada gambar 27 diatas memperlihatkan kondisi kurva kegagalan/kesalahan pelatihan yang mendekati nilai 0 seiring bertambahnya epoch, baik pada dataset pelatihan maupun validasinya. Nilai kegagalan pada epoch terakhir pada kegagalan pelatihan sebesar 0.3328, sedangkan pada kegagalan validasi sebesar 0.3848.

Kemudian terdapat *confusion matrix* yang digunakan untuk mengetahui tingkat keberhasilan dan kegagalan pada pengujian yang dilakukan. Dalam *confusion matrix* ini

terdapat label benar dan label prediksi yang ditampilkan dengan masing-masing label terdapat nama penyakit-penyakit pada daun tomat dan daun singkong.



Gambar-28 Confusion Matrix menggunakan arsitektur VGG16.

2. ARSITEKTUR INCEPTION-V3

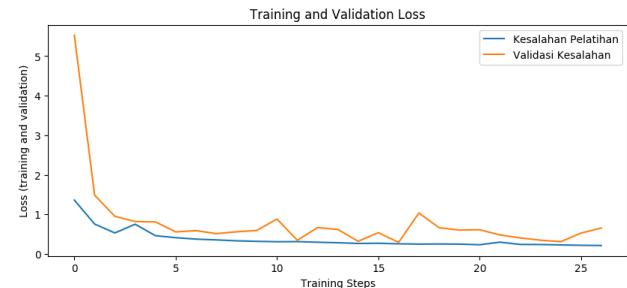
Pelatihan dan pengujian *dataset* menggunakan arsitektur *Inception* Versi 3 menggunakan gambar berukuran $75 \times 75 \times 3$ *pixel*. Menggunakan *dataset train* Sebanyak 19.052, *dataset validasi* sebanyak 2.858, dan *dataset test* sebanyak 1.906.



Gambar-29 Kurva akurasi training dan validasi Menggunakan Arsitektur InceptionV3

Pada gambar 29 diatas memperlihatkan kondisi kurva akurasi pelatihan yang tidak stabil seiring bertambahnya epoch, terutama pada akurasi validasinya. Sehingga diperoleh nilai akurasi pada *epoch* terakhir (*epoch* ke-27) yaitu nilai akurasi pelatihan dan akurasi validasi masing-masing adalah acc: 0.9230 dan 0.7706.

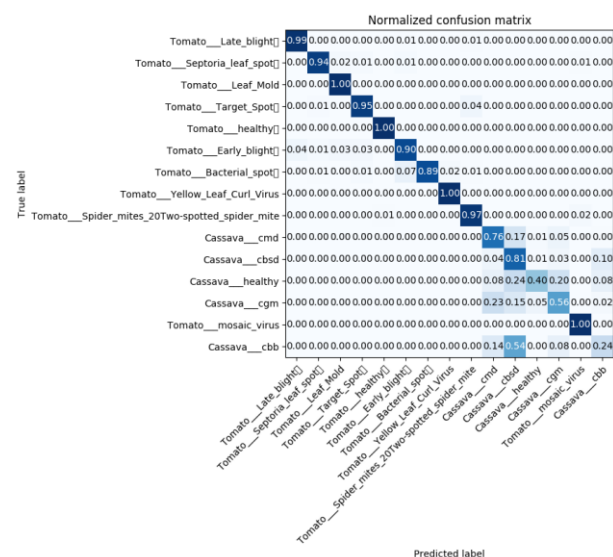
Adapun ditampilkan hasil pelatihan pada dataset pelatihan dan validasi dalam kegagalannya pada Gambar 30 dibawah ini.



Gambar-30 Kurva Loss Training dan validasi Menggunakan arsitektur InceptionV3

Pada gambar 30 diatas memperlihatkan kondisi kurva kegagalan/kesalahan pelatihan yang tidak stabil, bahkan pada *epoch* terakhir mengalami kenaikan *loss* pada data validasi. Nilai kegagalan pada epoch terakhir pada kegagalan pelatihan sebesar 0.2117, sedangkan pada kegagalan validasi sebesar 0,629.

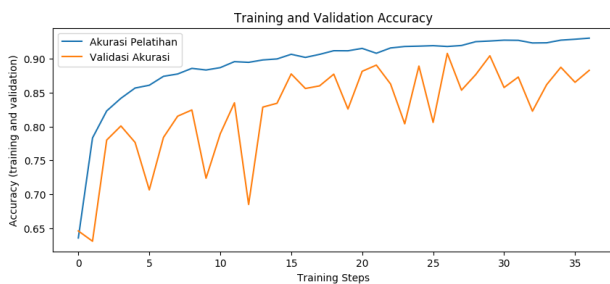
Kemudian terdapat *confusion matrix* yang digunakan untuk mengetahui tingkat keberhasilan dan kegagalan pada pengujian yang dilakukan. Dalam *confusion matrix* ini terdapat label benar dan label prediksi yang ditampilkan dengan masing-masing label terdapat nama penyakit-penyakit pada daun tomat dan daun singkong.



Gambar-31. Confusion Matrix menggunakan Asritektur InceptionV3

3. ARSITEKTUR *MOBILENET*

Pelatihan dan pengujian *dataset* menggunakan arsitektur *MobileNetV1* menggunakan gambar berukuran 64x64x3 *pixel*. Menggunakan *dataset train* Sebanyak 19.052, *dataset validasi* sebanyak 2.858, dan *dataset test* sebanyak 1.906.



Gambar-32 Kurva Akurasi Training Dan Validasi Menggunakan Arsitektur *MobileNetV1*

Pada gambar 32 diatas memperlihatkan kondisi kurva akurasi pelatihan yang mendekati nilai 1 di seiring bertambahnya epoch, meskipun pada akurasi validasi tidak stabil. Diperoleh nilai akurasi pada epoch terakhir (epoch ke-37) yaitu nilai akurasi pelatihan dan akurasi validasi masing-masing adalah 0.9303 dan 0.8828.

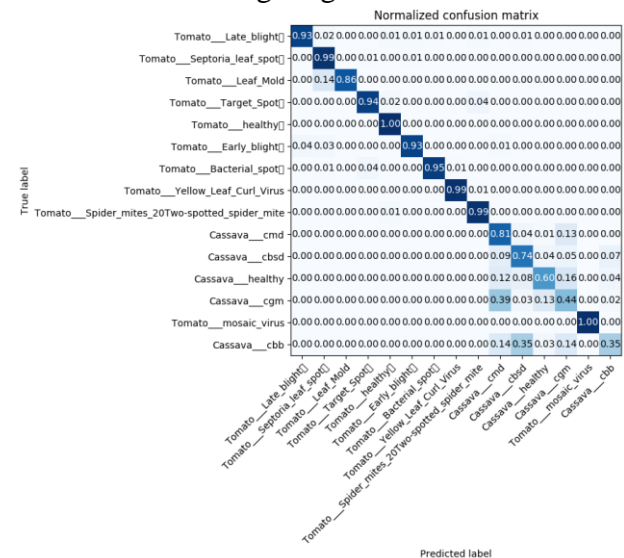
Adapun ditampilkan hasil pelatihan pada dataset pelatihan dan validasi dalam kegagalannya pada Gambar 33 dibawah ini.



Gambar-33 Kurva Loss Training dan Validasi menggunakan Arsitektur *MobileNetV1*

Pada gambar 33 diatas memperlihatkan kondisi kurva kegagalan/kesalahan pelatihan yang mendekati nilai 0 seiring bertambahnya epoch, baik pada dataset pelatihan maupun validasinya. Nilai kegagalan pada epoch terakhir pada kegagalan pelatihan sebesar 0.2074, sedangkan pada kegagalan validasi sebesar 0.3919.

Kemudian terdapat confusion matrix yang digunakan untuk mengetahui tingkat keberhasilan dan kegagalan pada pengujian yang dilakukan. Dalam confusion matrix ini terdapat label benar dan label prediksi yang ditampilkan dengan masing-masing label terdapat nama penyakit-penyakit pada daun tomat dan daun singkong.

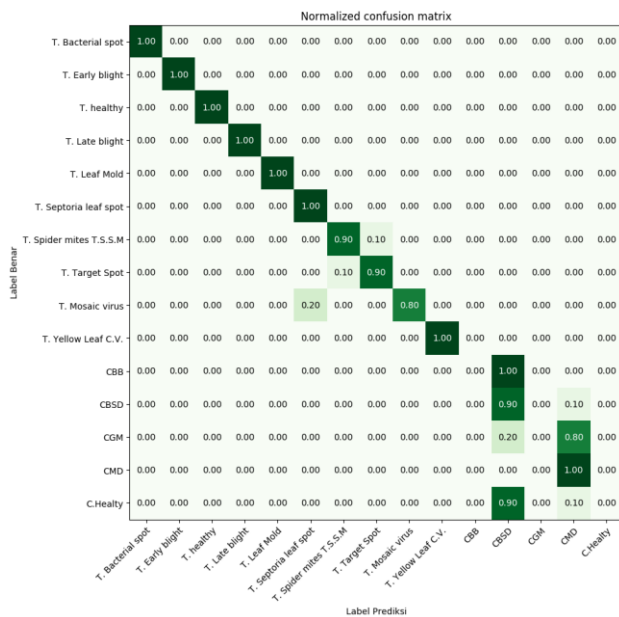


Gambar-34 Confusion Matrix Menggunakan Arsitektur *MobileNetV1*

C. PENGUJIAN PADA ANDROID

1. PENGUJIAN PADA ARSITEKTUR *VGG16*

Pengujian dilakukan dengan memasukan gambar dengan cara memotret langsung atau melalui *import galeri* pada aplikasi *Plant Diseases*. Data yang di gunakan sebanyak 10 pada setiap penyakit.



Gambar-35. Confusion Matrix Pengujian Pada Android Menggunakan Arsitektur VGG16

Dari gambar pengujian tersebut dapat dilihat bahwa hasil prediksi menggunakan arsitektur VGG16 kurang cocok untuk digunakan pada android. Meskipun memiliki kurva yang baik pada saat proses pelatihan dan validasi namun VGG16 kurang cocok untuk diaplikasikan kedalam android.

$$\%akurasi = \frac{jumlahprediksibenar}{jumlahdatapengujian} \times 100\%$$

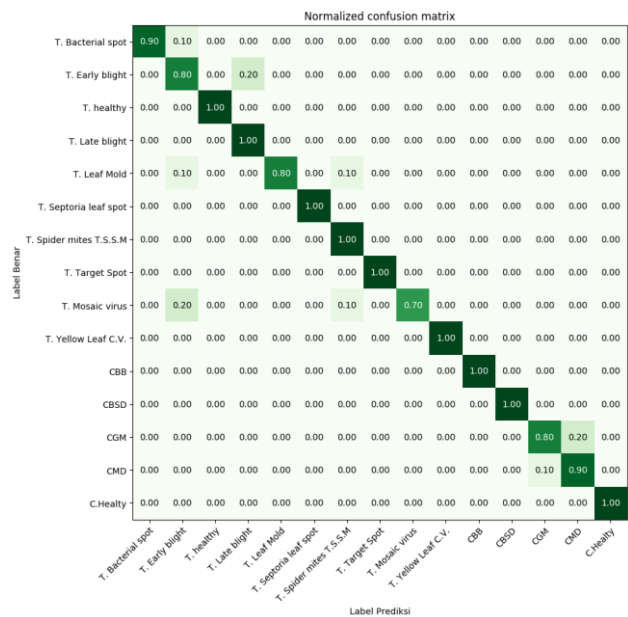
$$\%akurasi = \frac{115}{150} \times 100\% = 76,67\%$$

$$\%kesalahan = \frac{jumlahprediksialah}{jumlahdatapengujian} \times 100\%$$

$$\%kesalahan = \frac{35}{150} \times 100\% = 23,33\%$$

2. PENGUJIAN PADA ARSITEKTUR INCEPTION - V3

Pengujian dilakukan dengan memasukan gambar dengan cara memotret langsung atau melalui *import* galeri pada aplikasi *Plant Diseases*. Data yang di gunakan sebanyak 10 pada setiap penyakit.



Gambar-36 Confusion Matrix Pengujian Pada Android Menggunakan Arsitektur InceptionV3

Dari gambar pengujian tersebut dapat dilihat bahwa hasil prediksi *relative* bagus, hanya beberapa kesalahan prediksi yang terjadi pada beberapa penyakit. Hal ini dikarenakan jumlah *dataset* yang tidak seimbang dan penggunaan arsitektur, sehingga mempengaruhi proses pelatihan dan validasi.

$$\%akurasi = \frac{jumlahprediksibenar}{jumlahdatapengujian} \times 100\%$$

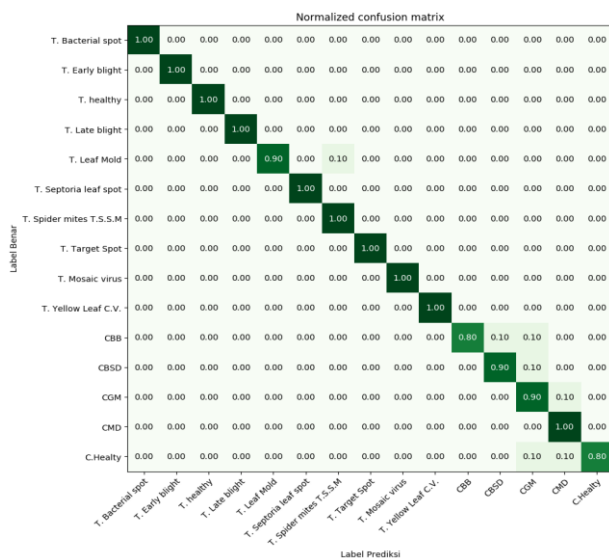
$$\%akurasi = \frac{139}{150} \times 100\% = 92,67\%$$

$$\%kesalahan = \frac{jumlahprediksialah}{jumlahdatapengujian} \times 100\%$$

$$\%kesalahan = \frac{11}{150} \times 100\% = 7,33\%$$

3. PENGUJIAN PADA ARSITEKTUR MOBILENET

Pengujian dilakukan dengan memasukan gambar dengan cara memotret langsung atau melalui *import* galeri pada aplikasi *Plant Diseases*. Data yang di gunakan sebanyak 10 pada setiap penyakit.



Gambar-37 Confusion Matrix Pengujian Pada Android Menggunakan Arsitektur MobileNetV1

Dari gambar 37 pengujian tersebut dapat dilihat bahwa hasil prediksi untuk daun tomat relative bagus hanya ada satu kesalahan prediksi pada penyakit *leaf mold*. Selanjutnya pada daun singkong terdapat beberapa kesalahan yang terjadi. Hal ini dikarenakan jumlah *dataset* yang tidak seimbang. Sehingga mempengaruhi proses pelatihan dan validasi.

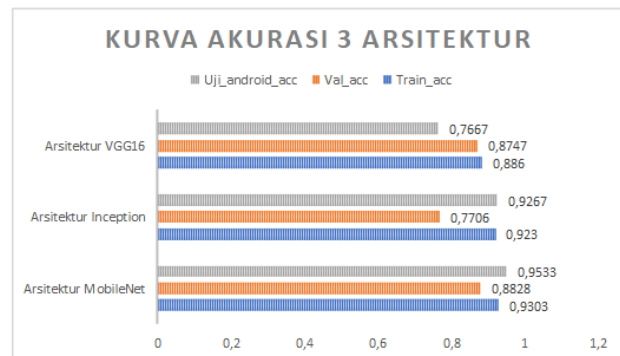
$$\%akurasi = \frac{\text{jumlahprediksibenar}}{\text{jumlahdatapengujian}} \times 100\%$$

$$\%akurasi = \frac{143}{150} \times 100\% = 95,33\%$$

$$\%kesalahan = \frac{\text{jumlahprediksialah}}{\text{jumlahdatapengujian}} \times 100\%$$

$$\%kesalahan = \frac{7}{150} \times 100\% = 4,67\%$$

Berdasarkan dari hasil penelitian ketiga arsitektur yang digunakan memiliki perbedaan masing-masing. Adapun perbedaannya sebagai berikut.

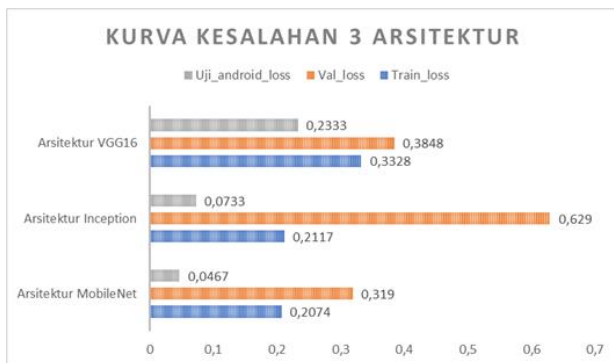


Gambar-38 Kurva Akurasi Perbandingan Arsitektur VGG16, InceptionV3 dan MobileNetV1

Kurva diatas menunjukkan perbandingan nilai akurasi antara tiga arsitektur yaitu VGG16, *InceptionV3* dan *MobilNetV1*. Dimana ketiganya memiliki perbedaan pada nilai akurasinya. Pada pengujian android *MobileNetV1* lebih unggul daripada *Inceptionv3* dan VGG16 dengan akurasi untuk *MobileNetV1* yaitu sebesar 0.9533, sedangkan pada arsitektur *InceptionV3* sebesar 0.927 dan pada arsitektur VGG16 memiliki nilai akurasi sebesar 0.7667.

Pada proses *training* dan *validation* ketiganya memiliki nilai akurasi dan kesalahan yang hampir sama. Nilai akurasi *training* dan *validation* pada arsitektur VGG16 adalah 0.8860 dan 0.8747. Nilai akurasi *training* dan *validation* pada arsitektur InceptionV3 adalah 0.9230 dan 0.7706, Nilai akurasi pada arsitektur *MobileNetV1* pada proses *training* dan *validation* adalah 0.933 dan 0.8828.

Pada ketiga arsitektur yang digunakan memiliki kekurangan yaitu berupa nilai kesalahan pada masing-masing proses, baik pada proses *training*, *validation*, dan pengujian pada android. Berikut kurva kesalahan yang terjadi pada tiga arsitektur ini.



Gambar-39 Kurva kesalahan Perbandingan Arsitektur VGG16, InceptionV3 dan MobileNetV1

Pada gambar 4.14 diatas menunjukan kurva kesalahan tiga arsitektur (VGG16, InceptionV3, dan MobileNetV1) yang terjadi pada proses *training*, *validation*, dan pengujian android. Pada pengujian android nilai kesalahan terkecil terdapat pada arsitektur *MobileNetV1* yaitu sebesar 0.0467, selanjutnya pada arsitektur *InceptionV3* yaitu sebesar 0.0733, dan terakhir pada arsitektur VGG16 yaitu sebesar 0.233.

Kemudian pada proses *training* dan *validation*, arsitektur VGG16 memiliki nilai kesalahan yaitu sebesar 0.3328 dan 0.3848. pada arsitektur *InceptionV3* memiliki nilai kesalahan sebesar 0.2117 dan 0.629. Pada Arsitektur *MobileNetV1* memiliki nilai kesalahan sebesar 0.2074 dan 0.319.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. *Deep learning* dapat mengklasifikasikan sesuatu secara otomatis. Sehingga tidak harus memberikan fitur-fitur yang diperlukan untuk mengenali sesuatu tersebut.
2. Jumlah dan kualitas *dataset* memberi pengaruh besar terhadap proses *training* dan *validation*. Jumlah *dataset* yang tidak seimbang pada setiap jenis klasifikasi dapat menjadikan proses *training* dan *validation* tidak stabil.
3. Arsitektur VGG16 memiliki kurva *training* dan *validation* yang lebih baik di bandingkan arsitektur *MobileNet* dan *Inception*.

4. Identifikasi penyakit tanaman menggunakan *google colab* untuk melakukan proses *training* dan *validation*, lalu kemudian mengkonversi dalam bentuk *tensorflow lite* agar bisa diaplikasikan pada android.
5. Pengujian identifikasi penyakit tanaman dilakukan pada aplikasi *Plant Diseases* yang terinstal pada *smartphone*, dengan cara memotret langsung obyek atau dengan cara *import Gallery*.
6. Arsitektur *MobileNet* memiliki nilai akurasi lebih tinggi yaitu sebesar 95,33% dibandingkan dengan arsitektur *Inception* (92,67%) dan VGG16 (76,67%) saat pengujian melalui aplikasi android.
7. Respon deteksi *MobileNet* lebih cepat dibandingkan dengan respon deteksi arsitektur lain. Dapat dilihat pada perhitungan nilai rata-rata respon deteksi yang dibutuhkan pada *MobileNet* yaitu sebesar 134,66, sedangkan pada arsitektur *Inception* dan VGG16 masing masing sebesar 643,6 dan 789,03

B. Saran

1. Menambah jumlah *dataset* sehingga jumlah *dataset* pada setiap jenis penyakit tanaman tidak jauh berbeda.
2. Mengembangkan arsitektur agar memperoleh nilai akurasi yang baik dan mengurangi jumlah *loss* pada proses *training* maupun *validation*.
3. Melakukan modifikasi untuk klasifikasi obyek lain agar penggunaan metode *Deep learning* dapat diaplikasikan lebih luas lagi.

DAFTAR PUSTAKA

- [1] Apriyantono, A., D. Fardiaz, N. L. Puspitasari, Sedamawati dan S. Budiyo, 1989. Analisis Pangan. PAU Pangan dan Gizi. IPB Press.
- [2] Departemen Kesehatan RI. 2003. Pedoman Penanggulangan Masalah Gizi dalam Keadaan Darurat. Direktorat Bina Gizi Masyarakat.
- [3] Badan Pusat Statistik Jakarta Pusat, 2012. Statistik Indonesia Tahun 2012. Jakarta Pusat : Badan Pusat Statistik.
- [4] J. Pertanian Agros. “Penyakit Tanaman”. Universitas Janabadra Yogyakarta.
- [5] Chollet, François. 2018. *Deep Learning with Python*. Shelter Island: Manning Publications Co.
- [6] Sena, Samuel. 2017. “Pengenal *Deep Learning Part 7 : Convolutional Neural Network* (CNN)”. [Daring]. Tersedia pada: <https://medium.com/@samuelsena/pengenal-an-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>. [Diakses: 10-Nov-2019].
- [7] Munir, Rinaldi. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung : Informatika
- [8] XenonStack. 2017. “Log Analytics With Deep Learning And Machine Learning”. [Daring]. Tersedia pada <https://medium.com/@xenonstack/log-analytics-with-deep-learning-and-machine-learning-20a1891ff70e> [Diakses Pada : 10-Nov-2019]
- [9] Tandungan, Sofyan. 2019. “Pengenal *Convolutional Neural Network – Part 1*”. [daring] tersedia pada: <http://sofyantandungan.com/pengenal-convolutional-neural-network-part-1/> [Diakses Pada : 10-Nov-2019]
- [10] K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. arXiv:1512.03385 [cs], Dec 2015. arXiv: 1512.03385
- [11] Adam, Rian. 2019. “Mengenal Google Colab”. [Daring]. Tersedia pada <https://structilmy.com/2019/05/mengenal-google-colab/> [Diakses Pada : 17-Nov-2019]
- [12] Moroney, Laurence. 2018. “Using TensorFlow Lite on Android”. [Daring] tersedia pada : <https://medium.com/tensorflow/using-tensorflow-lite-on-android-9bbc9cb7d69d> [Diakses Pada 17-Nov-2019]
- [13] Google. “Mengenal Android Studio” . [Daring] tersedia Pada : <https://developer.android.com/studio/intro/?hl=id> [Diakses Pada 17-Nov-2019]
- [14] Anonim. 2016. “Pengertian Penyakit tanaman”. [Daring] tersedia pada : <https://www.sampulpertanian.com/2016/10/pengertian-penyakit-tanaman.html> [Diakses Pada 17-Nov-2019]
- [15] Layanan Informasi Desa. 2018. “Penyakit pada Tanaman Tomat”. [Daring] tersedia pada : <https://8villages.com/full/petani/article/id/5b5af73bec9e5e7b5cd05e28> [Diakses Pada 17-Nov-2019]
- [16] Clemson University. 2018. “*Tomato Diseases & Disorders*”. [Daring] tersedia pada : <https://hgic.clemson.edu/factsheet/tomato-diseases-disorders/> [Diakses Pada 17-Nov-2019]
- [17] Jackson, Grahame. 2010. “*Pacific Pests and Pathogens - Fact Sheets*”. [Daring] tersedia pada : http://www.pestnet.org/fact_sheets/cucumber_target_spot_189.htm [Diakses pada : 17-Nov-19]
- [18] R. Hazzard, UMass Extension. 2013. “Two-spotted Spider Mite”. [Daring] tersedia pada : <https://ag.umass.edu/vegetable/fact-sheets/two-spotted-spider-mite> [diakses pada : 17-Nov-19]
- [19] Badan Penelitian Dan Pengembangan Pertanian. 2013. “Hama, Penyakit, dan Gulma pada Tanaman Ubi Kayu Identifikasi dan Pengendaliannya”. Jakarta : IAARD Press