

**IMPLEMENTASI *TRANSFER LEARNING* ARSITEKTUR  
MOBILENETV2 UNTUK IDENTIFIKASI  
PENYAKIT DAUN KENTANG**



**TESIS**

Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Magister Ilmu Komputer (M.Kom)

TIKA ADILAH M

14002364

Program Studi Ilmu Komputer (S2)

Fakultas Teknologi Informasi

Universitas Nusa Mandiri

2021

**IMPLEMENTASI *TRANSFER LEARNING* ARSITEKTUR  
MOBILENETV2 UNTUK IDENTIFIKASI  
PENYAKIT DAUN KENTANG**



**TESIS**

Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Magister Ilmu Komputer (M.Kom)

TIKA ADILAH M

14002364

Program Studi Ilmu Komputer (S2)

Fakultas Teknologi Informasi

Universitas Nusa Mandiri

2021

## SURAT PERNYATAAN ORISINALITAS DAN BEBAS PLAGIARISME

Yang bertanda tangan di bawah ini :

Nama : Tika Adilah M  
NIM : 14002364  
Program Studi : Ilmu Komputer  
Fakultas : Teknologi Informasi  
Jenjang : Strata Dua (S2)  
Konsentrasi : *Image Processing*

Dengan ini menyatakan bahwa tesis yang telah saya buat dengan judul: “Implementasi *Transfer Learning* Arsitektur MobileNetV2 Untuk Identifikasi Penyakit Daun Kentang” adalah hasil karya sendiri, dan semua sumber baik yang kutip maupun yang dirujuk telah saya nyatakan dengan benar dan tesis belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tesis yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Studi Ilmu Komputer (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri dicabut/dibatalkan.

Jakarta, 30 Juli 2021  
Yang menyatakan,



Tika Adilah M

## HALAMAN PERSETUJUAN DAN PENGESAHAN TESIS

Tesis ini diajukan oleh:

Nama : Tika Adilah M  
NIM : 14002364  
Program Studi : Ilmu Komputer  
Jenjang : Strata Dua (S2)  
Konsentrasi : *Image Processing*  
Judul Tesis : IMPLEMENTASI TRANSFER LEARNING ARSITEKTUR  
MOBILENETV2 UNTUK IDENTIFIKASI PENYAKIT  
DAUN KENTANG

Telah dipertahankan pada periode 2021-1 dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Ilmu Komputer (M.Kom) pada Program Studi Ilmu Komputer (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri.

Jakarta, 16 Agustus 2021

### PEMBIMBING TESIS

Pembimbing I : Dr. Yan Rianto, M.Eng



### DEWAN PENGUJI

Penguji I : Dr. Agus Subekti, M.T



Penguji II : Dr. Hilman Ferdinandus Pardede,  
S.T, M.EICT



Penguji III /  
Pembimbing I : Dr. Yan Rianto, M.Eng





## LEMBAR BIMBINGAN TESIS

UNIVERSITAS NUSA MANDIRI

NIM : 14002364  
Nama Lengkap : Tika Adilah M  
Dosen Pembimbing : Dr. Yan Rianto, M.Eng  
Judul Tesis : Implementasi *Transfer Learning* Arsitektur  
MobileNetV2 Untuk Identifikasi Penyakit Daun  
Kentang

No	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1.	21 April 2021	Bimbingan Pedana+Acc Judul Tesis	
2.	25 April 2021	Konsultasi Metode	
3.	3 Mei 2021	Acc Metode + Konsultasi Aplikasi	
4.	20 Mei 2021	Acc Aplikasi	
5.	26 Mei 2021	Konsultasi File PPT	
6.	10 Juli 2021	Acc PPT	
7.	19 Juli 2021	Konsultasi BAB I-V	
8.	30 Juli 2021	Acc BAB I-V	

Catatan untuk Dosen Pembimbing.

Bimbingan Tesis

- Dimulai pada tanggal : 21 April 2021
- Diakhiri pada tanggal : 30 Juli 2021
- Jumlah pertemuan bimbingan : 8 (Delapan) kali bimbingan

Disetujui oleh,  
**Dosen Pembimbing**

Dr. Yan Rianto, M.Eng

## KATA PENGANTAR

Puji syukur Alhamdulillah, penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan tesis ini tepat pada waktunya. Dimana laporan tesis ini penulis sajikan dalam bentuk buku yang sederhana.

Tujuan penulisan laporan tesis ini adalah sebagai salah satu syarat untuk memperoleh gelar Magister Ilmu Komputer (M.Kom) pada Program Studi Ilmu Komputer (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri.

Tesis ini diambil berdasarkan hasil penelitian mengenai penyakit daun kentang dengan menggunakan 2152 citra, mulai dari pengumpulan *dataset*, augmentasi, ekstraksi fitur hingga klasifikasi. Dalam penulisannya, penulis mencari dan menganalisa berbagai macam sumber referensi, baik dalam bentuk jurnal ilmiah, buku-buku literatur, internet dan lain-lain yang berkaitan dengan pembahasan pada tesis ini.

Penulis menyadari bahwa tanpa bimbingan dan dukungan dari semua pihak dalam pembuatan laporan tesis ini, maka penulis tidak dapat menyelesaikan laporan tesis ini tepat pada waktunya. Untuk itu izinkanlah penulis pada kesempatan ini untuk mengucapkan terima kasih kepada:

1. Rektor Universitas Nusa Mandiri
2. Wakil Rektor Universitas Nusa Mandiri
3. Dekan Fakultas Teknologi Informasi Universitas Nusa Mandiri
4. Ketua Program Studi Ilmu Komputer
5. Bapak Dr. Yan Rianto, M.Eng selaku pembimbing yang telah meluangkan waktu, tenaga dan pikiran guna memberikan bimbingan bagi penulis selama penyusunan tesis.
6. Seluruh dosen Program Studi Ilmu Komputer (S2) Universitas Nusa Mandiri yang telah memberikan ilmu, pengetahuan dan motivasi selama penulis menempuh studi S2.
7. Yayasan Bina Sarana Informatika yang telah memberikan beasiswa kepada penulis sehingga penulis dapat menempuh pendidikan tinggi hingga S2.

8. Orang tua, keluarga dan sahabat yang telah memberikan dukungan baik secara moral dan materiil serta do'a yang tak henti.
9. Yusup Rizal Muttaqin yang selalu ada, couldn't have done it without you.
10. Teman-teman seperjuangan yang telah memberikan dukungan moral.

Serta semua pihak yang tidak dapat penulis sebutkan satu persatu, sehingga atas bantuan semua pihak tersebut penulis dapat menyelesaikan penulisan tesis ini. Penulis menyadari bahwa dalam penulisan tesis ini masih terdapat banyak kekurangan dan kekeliruan, untuk itu penulis mengharapkan kritik dan saran membangun agar penulisan karya ilmiah yang penulis hasilkan di masa yang akan datang menjadi semakin lebih baik.

Akhir kata semoga laporan tesis ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 30 Juli 2021



Tika Adilah M  
Penulis

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya :

Nama : Tika Adilah M  
NIM : 14002364  
Program Studi : Ilmu Komputer  
Fakultas : Teknologi Informasi  
Jenjang : Strata Dua (S2)  
Konsentrasi : *Image Processing*  
Jenis Karya : Tesis

Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan ijin kepada pihak Program Studi Ilmu Komputer (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri **Hak Bebas Royalti Non-Eksklusif (*Non-exclusive Royalti-Free Right*)** atas karya ilmiah kami yang berjudul : “Implementasi *Transfer Learning* Arsitektur MobileNetV2 Untuk Identifikasi Penyakit Daun Kentang” beserta perangkat yang diperlukan (apabila ada).

Dengan **Hak Bebas Royalti Non-Eksklusif** ini pihak Universitas Nusa Mandiri berhak menyimpan, mengalih-media atau *bentuk*-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Nusa Mandiri, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 30 Juli 2021  
Yang Menyatakan



Tika Adilah M

## ABSTRAK

Nama : Tika Adilah M  
NIM : 14002364  
Program Studi : Ilmu Komputer  
Jenjang : Strata Dua (S2)  
Konsentrasi : *Image Processing*  
Judul : Implementasi *Transfer Learning* Arsitektur MobileNetV2 Untuk Identifikasi Penyakit Daun Kentang

Kentang termasuk salah satu tanaman pangan paling penting ketiga di dunia. Pertanian kentang memiliki masalah berupa penyakit yang menyerang pada daun. Penyakit tersebut dapat memengaruhi kualitas tanaman kentang, sehingga mengakibatkan gagal panen. *Digital image processing* merupakan metode yang dapat digunakan untuk membantu petani dalam mengidentifikasi penyakit daun kentang. Pengembangan *digital image processing* telah banyak dilakukan salah satunya dengan menggunakan algoritma *Convolutional Neural Network* (CNN). CNN memerlukan data yang besar. Arsitektur CNN akan mengalami *overfitting* jika menggunakan data yang sedikit, dimana model klasifikasi memiliki akurasi yang tinggi pada data latih tetapi akurasinya menjadi buruk pada data uji. Penelitian ini memanfaatkan metode *Transfer Learning* dan Augmentasi untuk menghindari *overfitting* pada data yang terlalu sedikit. Metode *Transfer Learning* yang digunakan dalam penelitian ini adalah MobileNetV2. Hasil dari uji coba pada penelitian ini menunjukkan bahwa metode *Transfer Learning* MobileNetV2 memiliki hasil performa klasifikasi yang baik dan menghasilkan nilai akurasi tinggi sebesar 99,6%.

Kata Kunci : Klasifikasi penyakit daun, *Transfer Learning*, MobileNetV2, Penyakit Daun Kentang

## ***ABSTRACT***

Name : Tika Adilah M  
NIM : 14002364  
Study Program : Ilmu Komputer  
Levels : Strata Dua (S2)  
Concentration : Image Processing  
Title : Implementation of Transfer Learning MobileNetV2 Architecture for Identification of Potato Leaf Disease

*Potatoes are one of the third most important food crops in the world. Potato farming has problems in the form of diseases that attack the leaves. These diseases can affect the quality of potato plants, resulting in crop failure. Digital image processing is a method that can be used to assist farmers in identifying potato leaf diseases. The development of digital image processing has been carried out, one of which is by using the Convolutional Neural Network (CNN) algorithm. CNN requires big data. CNN architecture will experience overfitting if it uses little data, where the classification model has high accuracy on training data but poor accuracy on test data. This research utilizes Transfer Learning and Augmentation methods to avoid overfitting on too little data. Transfer Learning method used in this research is MobileNetV2. The results of the trials in this study indicate that the MobileNetV2 Transfer Learning method has good classification performance results and produces a high accuracy value of 99.6%.*

*Keywords: Leaf disease classification, Transfer Learning, MobileNetV2, Potato Leaf Disease.*

## DAFTAR ISI

<b>HALAMAN SAMPUL</b> .....	<b>i</b>
<b>HALAMAN JUDUL</b> .....	<b>ii</b>
<b>SURAT PERNYATAAN ORISINALITAS DAN BEBAS PLAGIARISME</b> .....	<b>iii</b>
<b>HALAMAN PERSETUJUAN DAN PENGESAHAN TESIS</b> .....	<b>iv</b>
<b>LEMBAR BIMBINGAN TESIS</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS</b> .....	<b>viii</b>
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang Penulisan .....	1
1.2. Identifikasi Masalah.....	4
1.3. Tujuan Penelitian .....	4
1.4. Ruang Lingkup .....	4
1.5. Hipotesis .....	4
1.6. Sistematika Penulisan .....	5
<b>BAB II LANDASAN PEMIKIRAN</b> .....	<b>6</b>
2.1. Tinjauan Pustaka.....	6
2.1.1. <i>Digital Image Processing</i> .....	6
2.1.2. <i>Augmentasi Citra</i> .....	6
2.1.3. <i>Deep Learning</i> .....	7
2.1.4. <i>Convolutional Neural Network (CNN)</i> .....	7
2.1.4.1. <i>Convolutional Layer</i> .....	8
2.1.4.2. <i>Pooling Layer</i> .....	8
2.1.4.3. <i>Aktivasi Function</i> .....	9
2.1.4.4. <i>Global Average Pooling</i> .....	10
2.1.5. <i>Transfer Learning</i> .....	11
2.1.6. <i>MobileNetV2</i> .....	12
2.1.7. <i>Penyakit daun Kentang</i> .....	13
2.2. Tinjauan Studi.....	14
2.3. Objek Penelitian .....	18
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>19</b>
3.1 <i>Data Collection</i> .....	20

3.2. <i>Augmentation</i> .....	21
3.3. <i>Pre-processing</i> .....	21
3.4. <i>Split Data</i> .....	21
3.5. <i>Image Classifier</i> .....	22
3.6. <i>FlatBuffer File Converter</i> .....	25
3.7. Evaluasi dan Validasi.....	26
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>27</b>
4.1. Hasil Penelitian .....	27
4.1.1. <i>Augmentation</i> .....	27
4.1.2. <i>Pre-Processing</i> .....	28
4.1.3. <i>Image Classifier</i> .....	29
4.1.3.1. Pengujian Model dengan <i>Transfer Learning</i> MobileNetV2 .	29
4.1.3.2. Pengujian Model dengan <i>Transfer Learning</i> InceptionResNetV2, VGG16 dan Inception V3.....	32
4.2. Implementasi Sistem .....	36
4.2.1. Tampilan Home .....	36
4.2.2. Halaman Deteksi dari Galeri.....	36
4.2.3. Halaman Deteksi dengan Kamera.....	37
4.2.4. Halaman Tentang .....	38
4.2.5. Halaman Petunjuk.....	39
4.3. Evaluasi dan Validasi Hasil .....	39
<b>BAB V PENUTUP.....</b>	<b>41</b>
5.1. Kesimpulan.....	41
5.2. Saran .....	41
<b>DAFTAR PUSTAKA .....</b>	<b>43</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>46</b>
<b>LAMPIRAN.....</b>	<b>47</b>

## DAFTAR TABEL

Halaman

Tabel 2.1. Memilih Pendekatan Transfer Learning .....	12
Tabel 2.2. Penelitian Sebelumnya .....	16
Tabel 3.1. Dataset.....	20
Tabel 3.2. Data Train, Valid dan Test .....	22
Tabel 3.3. Bottleneck Resudial Block.....	23
Tabel 3.4. Arsitektur MobileNetV2 .....	24
Tabel 3.5. Hyperparameter yang digunakan .....	25
Tabel 4.1. Dataset Hasil Augmentasi .....	28
Tabel 4.2. Hasil Konversi BGR ke RGB .....	29
Tabel 4.3. Confusion Matrix MobileNetV2 Tanpa Augmentasi.....	31
Tabel 4.4. Confusion Matrix MobileNetV2 dengan Augmentasi .....	32
Tabel 4.5. Perbandingan Kinerja MobileNetV2 dengan distribusi data asli dan data hasil augmentasi .....	32
Tabel 4.6. Confusion Matrix Model InceptionResNetV2.....	33
Tabel 4.7. Confusion Matrix Model VGG16 .....	34
Tabel 4.8. Confusion Matrix Model InceptionV3.....	35
Tabel 4.9. Perbandingan Hasil Pengujian .....	35
Tabel 4.10. Penjabaran Citra Teridentifikasi .....	39

## DAFTAR GAMBAR

	Halaman
Gambar 2.1. Arsitektur Convolutional Neural Network .....	8
Gambar 2.2. Convolutional Layer .....	8
Gambar 2.3. Operasi max-pooling .....	9
Gambar 2.4. ReLU .....	9
Gambar 2.5. Global Average Pooling .....	11
Gambar 2.6. Transfer Learning .....	11
Gambar 2.7. MobileNetV2 .....	13
Gambar 2.8. Late Blight (a), Early Blight (b) .....	14
Gambar 3.1. Model Desain Penelitian .....	19
Gambar 3.2. Citra Asli (a) Early Blight, (b) Healthy dan (c) Late Blight .....	20
Gambar 3.3. Proses Image Classifier .....	23
Gambar 3.4. Proses Konversi Tensorflow Lite .....	25
Gambar 4.1. Hasil Augmentasi .....	28
Gambar 4.2. Hasil dari model klasifikasi MobileNetV2 tanpa Augmentasi (a) loss (b) accuracy .....	30
Gambar 4.3. Hasil dari model klasifikasi MobileNetV2 dengan Augmentasi (a) loss (b) accuracy .....	31
Gambar 4.4. Hasil dari model klasifikasi transfer learning InceptionResNetV2 (a) loss (b) accuracy .....	33
Gambar 4.5. Hasil dari model klasifikasi VGG16 (a) loss (b) accuracy .....	34
Gambar 4.6. Hasil dari model klasifikasi InceptionV3 (a) loss (b) accuracy .....	35
Gambar 4.7. Tampilan Home .....	36
Gambar 4.8. Halaman Deteksi dari Galeri .....	37
Gambar 4.9. Halaman Deteksi dengan Kamera .....	38
Gambar 4.10. Halaman Tentang .....	38
Gambar 4.11. Halaman Petunjuk .....	39

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Contoh Dataset Citra Daun Kentang .....	47
Lampiran 2. Citra Data Test.....	48

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Penulisan

Kentang (*Solanum tuberosum L.*) merupakan salah satu tanaman pangan paling penting ketiga di dunia yang berasal dari daerah dataran tinggi Andes, Amerika Selatan [1]. Kentang sangat menarik dalam sistem produksi pertanian karena mengkonsolidasikan potensi hasil yang luar biasa tinggi dengan nilai gizi yang tinggi. Kentang ditanam di lebih dari 125 negara dan dikonsumsi hampir di setiap bagian dunia [2]. Di Indonesia sendiri kentang menjadi makanan pokok yang populer. Kentang tumbuh dengan baik di dataran tinggi atau daerah pegunungan dengan tingkat kemiringan 800 – 1.500 m di atas permukaan laut (*dpl*).

Kentang merupakan produk pertanian yang telah berkembang pesat di dekade ini. Setiap tahun, jumlah produksi hampir mencapai 850.000 ton. Jumlah produksi tersebut dihasilkan dari area dengan luas sekitar 60.000 hektar. Area penanaman serta produksi kentang telah meningkat sekitar 10% per tahun, hal ini membuat Indonesia menjadi negara penghasil kentang terbesar di Asia Tenggara. Namun, kentang bisa terkena penyakit yang mempengaruhi tahap tanaman sebelum dan sesudah panen [3].

Penyakit tanaman adalah penyebab umum hilangnya produksi tanaman yang dapat memiliki dampak ekonomi yang signifikan yang mengakibatkan penurunan pendapatan bagi produsen dan distributor dan kenaikan harga bagi konsumen. Penyakit yang sering menyerang tanaman kentang diantaranya adalah penyakit bercak kering (*Early Blight*) dan penyakit hawar daun atau busuk daun (*Late Blight*). Berdasarkan pengalaman para petani kentang, penyakit tersebut muncul saat tanaman kentang berumur 5-6 minggu setelah ditanam [4]. Serangan penyakit ini sangat berpotensi menyebar ke bagian lain dari tanaman kentang, seperti batang, tangkai dan umbi kentang. Oleh sebab itu, petani kentang harus memangkas secara dini daun yang sudah terinfeksi penyakit agar penyakit tidak menyebar dan petani bisa mendapatkan hasil panen yang baik [5].

Identifikasi dini penyakit daun kentang sangat penting, untuk itu metode baru untuk identifikasi penyakit kentang diperlukan untuk mencegah hilangnya produksi dan peningkatan kualitas. Identifikasi dini ini menjadi tantangan bagi para petani, hal

ini dikarenakan mereka harus menghadapi berbagai permasalahan penyakit yang tidak dapat diatasi dengan prediksi manual. Jika mereka dapat mengetahui penyakit sejak dini, itu akan membantu para petani membuat keputusan yang akan meningkatkan produksi kentang. Identifikasi manual penyakit kentang tidak hanya sulit dan rawan kesalahan tetapi juga memakan waktu sementara sistem berbasis teknologi lebih efisien dan hemat biaya[2]. Perkembangan revolusi industri 4.0 ditekankan di berbagai bidang salah satunya dalam bidang pertanian [6]. Dalam bidang pertanian, teknologi digital bisa dimanfaatkan salah satunya untuk deteksi otomatis penyakit pada daun tanaman kentang.

Sistem *Deep Learning* (DL) saat ini banyak digunakan untuk mengotomatiskan proses. *Deep Learning* pencarian pola penyakit merupakan salah satu solusi yang dapat diberikan dan mudah diakses oleh petani kecil [7]. Dengan diberikan cukup data citra tanaman yang terinfeksi, maka bisa melatih sistem *deep learning* yang mampu mengidentifikasi penyakit yang diberikan sesuai data.

Dalam penelitian ini memiliki tujuan yaitu melakukan klasifikasi jenis penyakit pada daun tanaman kentang. Untuk memastikan jenis penyakit yang tepat, saat ini penggunaan *deep learning* menjadi jalan yang optimal dengan tujuan untuk mendapatkan algoritma terlatih yang dapat mengklasifikasikan jenis penyakit daun pada tanaman kentang [8]. Metode yang akan digunakan yaitu *Convolutiona Neural Network* yang nantinya akan menunjukkan keakuratan dari hasil klasifikasi penyakit daun tanaman kentang. Model CNN adalah metode pengenalan modern dan efisien. Berbeda dengan metode klasifikasi citra tradisional [9], *Convolutiona Neural Network* menggunakan konvolusi multilayer yang secara otomatis mengekstrak dan menggabungkan fitur. Algoritma ini dirancang untuk dilakukan secara independen dan dilatih untuk menyelesaikan tugas tertentu [10]. CNN sangat cocok untuk klasifikasi pada jumlah data yang besar. Namun data yang didapatkan untuk digunakan dalam melakukan klasifikasi citra tidak semuanya selalu memiliki jumlah data yang besar, data yang didapatkan bisa saja memiliki proporsi yang tidak seimbang untuk setiap kelasnya dan ini mempengaruhi pada tingkat akurasi yang dihasilkan pada proses klasifikasi. Solusi dalam mengatasi data yang sedikit dapat digunakan teknik augmentasi [11] dan pada proses klasifikasinya dapat menggunakan model *transfer learning* pada arsitektur CNN, model transfer learning atau biasa disebut juga

*pretrained model* adalah arsitektur CNN yang sudah dilatih dengan suatu dataset sebelumnya yang nantinya bisa digunakan untuk melatih dataset lain [12].

Penelitian mengenai deteksi penyakit tanaman telah banyak dilakukan. Augmentasi penting dilakukan seperti penelitian Utpal terhadap citra daun kentang bertujuan untuk menatasi *imbalance* data, penelitian Utpal mendapatkan akurasi tertinggi 97% [13]. Jumlah dataset citra daun kentang yang sedikit membuat faranbee dkk melakukan penelitian identifikasi penyakit daun tanaman kentang menggunakan metode CNN dengan *transfer learning* dan mendapat tingkat akurasi 99.43%, metode *transfer learning* merupakan metode yang menjanjikan dan dapat diterapkan dalam aplikasi seluler [9].

Penelitian ini melakukan klasifikasi penyakit pada daun kentang dengan menggunakan arsitektur *Convolutional Neural Network* (CNN). Data yang digunakan pada penelitian ini memiliki jumlah yang *imbalance* dan sedikit, permasalahan yang bisa muncul ketika menggunakan data yang kurang banyak adalah terjadinya *overfitting* [14], sehingga pada penelitian ini menggunakan teknik augmentasi data untuk mengatasi *imbalance* data dan menggunakan model *transfer learning* arsitektur MobileNetV2 untuk mengatasi masalah data yang sedikit. Kelebihan dari metode *Transfer Learning* adalah kemampuannya dalam menghasilkan informasi yang lebih banyak untuk klasifikasi karena memiliki lapisan yang lebih mendalam dibandingkan CNN sederhana.

Di antara banyak cara untuk menerapkan model *deep learning*, salah satu cara termudah adalah dengan menerapkannya di perangkat seluler. Keuntungannya adalah perangkat seluler populer dan mudah digunakan. Pengguna bisa mendapatkan jawaban dengan beberapa sentuhan. Apalagi model *deep learning* bisa menerima jumlah yang besar data secara real time berkat kamera ponsel. Saat menerapkan model *deep learning* di perangkat seluler, dua aspek harus dipertimbangkan: ukuran file model dan kecepatan proses. Jika ukurannya terlalu besar, tidak mungkin menerapkan model pada perangkat seluler. Jika prosesnya lambat, itu akan menyebabkan ketidaknyamanan bagi pengguna. Maka pada penelitian ini mengusulkan teknik pengolohan citra penyakit daun kentang dengan menggunakan *transfer learning* arsitektur MobileNetV2 untuk klasifikasi pada citra penyakit daun kentang.

## 1.2. Identifikasi Masalah

Agar tujuan terfokus pada permasalahan yang ada maka terdapat beberapa identifikasi masalah, diantaranya:

1. Apakah metode Augmentasi dapat meningkatkan kinerja model?
2. Apakah metode *transfer learning* MobilenetV2 dapat mengklasifikasi penyakit daun kentang?
3. Apakah Nilai akurasi klasifikasi jenis penyakit daun kentang menggunakan metode *transfer learning* MobilenetV2 cukup tinggi?
4. Bagaimana pengimplementasian metode *transfer learning* MobilenetV2 kedalam aplikasi android?

## 1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Melakukan Augmentasi untuk meningkatkan kinerja model.
2. Mengklasifikasi daun kentang menggunakan metode *transfer learning* MobilenetV2.
3. Mengetahui akurasi yang dihasilkan dalam proses klasifikasi penyakit daun kentang menggunakan metode *transfer learning* MobilenetV2.
4. Mengimplementasikan penelitian dalam bentuk aplikasi android.

## 1.4. Ruang Lingkup

Ruang lingkup penelitian digunakan untuk membatasi pokok bahasan penelitian agar terfokus dan sesuai dengan tujuan penelitian. Ruang lingkup masalah pada penelitian dibatasi pada proses pengklasifikasian daun tanaman kentang menjadi tiga kelas yakni, Early blight, Healty dan Late Blight menggunakan metode *transfer learning* MobilenetV2. Dataset yang digunakan yaitu berasal dari dataset penyakit daun kentang yang diambil dari: <https://www.kaggle.com/sohaibalam67/potato-disease>. Aplikasi dikembangkan menggunakan Android Studio.

## 1.5. Hipotesis

Hipotesis merupakan dugaan awal terhadap sebuah penelitian yang dilakukan. Dalam penelitian ini terdapat hipotesis yang ditentukan, yaitu: Hipotesis umum yang dijadikan aspek penelitian adalah diduga metode aumentasi dapat

digunakan untuk mengatasi imbalance data dan diduga metode *transfer learning* MobilenetV2 dapat digunakan dalam mengkalsifikasikan jenis penyakit pada daun kentang.

## **1.6. Sistematika Penulisan**

Sistematika penulisan merupakan gambaran secara umum dari masing-masing bab, yang terdiri dari pendahuluan, landasan pemikiran, metode penelitian, hasil penelitian dan pembahasan serta kesimpulan dan saran.

### **BAB I PENDAHULUAN**

Bab pendahuluan membahas tentang latar belakang dari penelitian dan melakukan perumusan masalah yang diangkat dalam penelitian, selain itu dibahas juga mengenai tujuan dilakukannya penelitian, ruang lingkup, kontribusi penelitian, serta manfaat dari penelitian.

### **BAB II LANDASAN PEMIKIRAN**

Bab landasan pemikiran mengupas tentang tinjauan studi yaitu membahas penelitian sebelumnya yang digunakan untuk mengetahui sejauh mana penelitian sebelumnya dan membahas tentang landasan teori terkait penelitian yang dilakukan.

### **BAB III METODE PENELITIAN**

Bab Metode Penelitian ini berisi tentang penjelasan mengenai perancangan penelitian diantaranya adalah metode pengumpulan data, metode yang diusulkan, eksperimen dan pengujian metode serta evaluasi dan validasi hasil.

### **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

Bab hasil penelitian dan pembahasan membahas tentang hasil penelitian berupa hasil dari klasifikasi daun kentang.

### **BAB V PENUTUP**

Bab kesimpulan dan saran berisi tentang ringkasan akhir dari penelitian serta saran-saran yang diberikan untuk mengembangkan penelitian selanjutnya.

## **BAB II**

### **LANDASAN PEMIKIRAN**

#### **2.1. Tinjauan Pustaka**

Bagian ini merupakan penjelasan mengenai teori-teori yang digunakan pada penelitian yang dilakukan, serta dijelaskan juga metode-metode yang digunakan dalam penelitian.

##### **2.1.1. *Digital Image Processing***

*Digital Image processing* merupakan metode yang digunakan dalam memproses atau memanipulasi citra 2 dimensi. *Digital image processing* dapat juga dikatakan sebagai operasi untuk menganalisa, mengubah atau memperbaiki suatu citra.

Konsep dasar pengolahan citra dalam pemrosesan suatu objek pada gambar diambil dari kemampuan indera penglihatan manusia dan selanjutnya dihubungkan dengan kemampuan otak manusia. Pengolahan citra telah banyak diaplikasikan dalam banyak bidang dengan tingkat keberhasilan yang cukup besar. Pengolahan citra menyangkut menyangkut berbagai cabang ilmu, seperti matematika, fotografi, elektronika dan teknologi komputer [15].

##### **2.1.2. Augmentasi Citra**

Augmentasi citra adalah teknik untuk menangani data yang sedikit. Augmentasi sering digunakan untuk menanggapi overfitting karena menghasilkan data pelatihan yang lebih banyak. Tujuan dari augmentasi adalah agar model tidak melihat citra yang sama persis dua kali [16]. Augmentasi biasanya terdiri dari operasi transformasi sebagai berikut:

- a. *Vertical\_flip* : membalik secara horizontal.
- b. *Horizontal\_flip* : membalik secara horizontal.
- c. *Width\_shift and height\_shift* : menerapkan pergeseran gambar.
- d. *Zoom\_range* : menerapkan transformasi memperbesar gambar
- e. *Shear\_range* : menerapkan transformasi geser secara acak
- f. *rotation\_range* : menerapkan rotasi gambar dengan skala 0-180 derajat.

### 2.1.3. *Deep Learning*

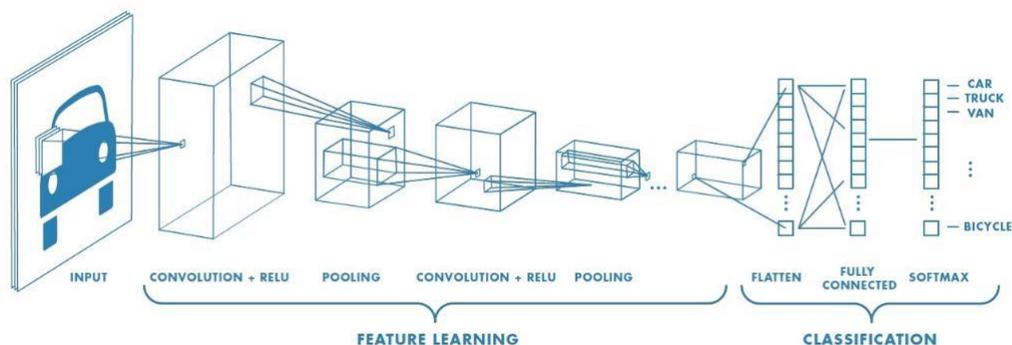
*Deep Learning* merupakan bagian dari *Machine Learning* yang memanfaatkan *artificial neural network* dalam mengatasi permasalahan dengan jumlah data yang besar. Teknik *deep learning* bekerja dengan menemukan struktur yang rumit pada kumpulan data yang besar menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah parameter internal yang akan digunakan untuk menghitung representasi disetiap lapisan serta representasi dilapisan sebelumnya [17].

Konsep *deep artificial neural network* dapat digunakan pada algoritma *machine learning* yang sudah ada sehingga komputer sekarang mampu belajar dengan skala, kecepatan dan akurasi yang besar. *Deep Learning* terus berkembang dan sering digunakan oleh peneliti serta industri dalam membantu memecahkan banyak masalah dalam data besar seperti misalnya *Computer vision* [18], *Speech recognition* [19], dan *Natural Language Processing*[20].

### 2.1.4. *Convolutional Neural Network (CNN)*

*Convolutional Neural Network* adalah suatu metode yang termasuk jenis *Deep Neural Network* yang didesain untuk mengolah data citra dua dimensi. CNN memiliki kedalaman jaringan yang tinggi dan sering diaplikasikan pada data citra. Penemuan CNN ini didasari pada penelitian yang dilakukan pertama kali oleh Hubel dan Wiesel [21] mengenai *visual cortex* pada indera penglihatan kucing. CNN secara teknis merupakan arsitektur yang terdiri dari beberapa tahap dan dapat dilatih. Masukan dan keluaran pada setiap tahap terdiri dari beberapa *array* atau sering disebut *feature map*. Pada setiap tahapannya terdiri atas tiga *layer* yakni konvolusi *layer*, aktivasi *layer* serta pooling *layer*.

*CNN* menggabungkan tiga pokok arsitektur, yaitu *Local Connection*, *Shared Weight* berupa *filter*, *Spatial Subsampling* berupa *Pooling* [22]. *Pooling layer* berfungsi untuk menggabungkan fitur yang hampir sama menjadi satu kemudian mengurangi dimensi representasi dan juga mencegah *overfitting* [23]. Fungsi dari konvolusi *layer* adalah matriks untuk melakukan *filtering*. Gambar 2.1 menunjukkan jaringan arsitektur CNN:

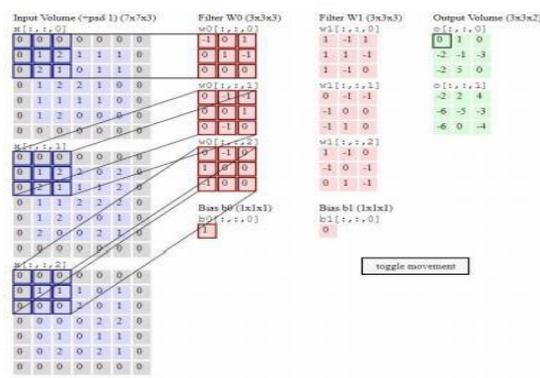


Gambar 2.1. Arsitektur *Convolutional Neural Network*

Sumber: [22]

#### 2.1.4.1. *Convolutional Layer*

*Convolution layer* adalah salah satu bagian dalam arsitektur CNN. Tujuan utama konvolusi dalam kaitannya dengan *ConvNet* adalah untuk mengekstraksi fitur gambar yang dimasukkan [24]. Pada tahapan ini dilakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi merupakan istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi output sebagai *feature map* dari input citra. Input dan output ini dapat dilihat sebagai dua argumen bernilai nyata.. Gambar 2.2 merupakan ilustrasi dari konvolusi layer.



Gambar 2.2. *Convolutional Layer*

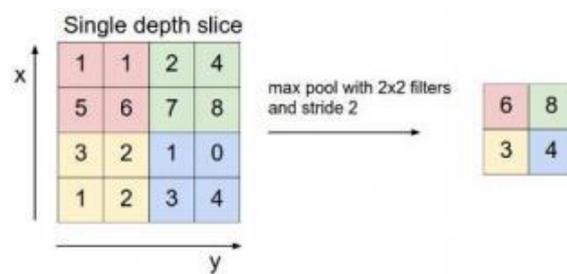
Sumber: [24]

#### 2.1.4.2. *Pooling Layer*

*Pooling layer* digunakan untuk mengurangi ukuran matriks menggunakan operasi *pooling*. *Pooling layer* ini berada setelah *Convolutional Layer* [25]. *Pooling*

layer pada dasarnya terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Terdapat dua jenis *pooling layer* yakni *max-pooling* serta *average pooling*. Pada *max-pooling* Nilai yang diambil merupakan nilai maksimal, sedangkan pada *average pooling* seperti namanya, nilai yang diambil adalah nilai rata-rata.

Gambar 2.3 merupakan gambaran dari proses *max-pooling*. Keluaran dari proses ini adalah sebuah matriks dengan ukuran dimensi lebih kecil dari citra awal. pada contoh dibawah ini operasi *max-pooling* menggunakan ukuran filter  $2 \times 2$ . *Input* berukuran  $4 \times 4$  ini diambil nilai maksimalnya dan dilanjutkan dengan membuat ukuran baru menjadi  $2 \times 2$ .



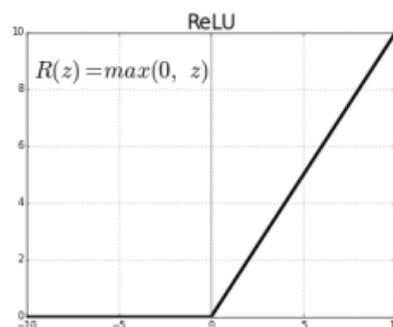
Gambar 2.3. Operasi *max-pooling*

Sumber: [25]

### 2.1.4.3. Aktivasi Function

fungsi dari *Activation Function* yaitu menentukan apakah suatu *neuron* harus “aktif” atau tidak berdasarkan dari *weighted sum* dari *input*. Terdapat dua jenis *activation function* yaitu *Linear* dan *Non-Linear activation function*.

#### A. ReLU (Non Linier)



Gambar 2.4. *ReLU*

Sumber: [26]

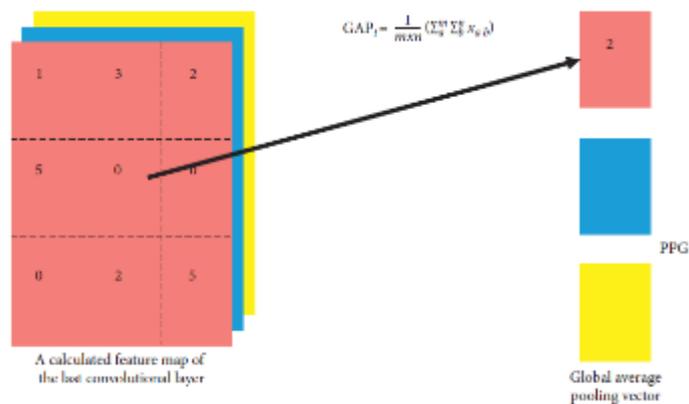
Pada dasarnya *ReLU* melakukan “*threshold*” dari 0 hingga *infinity*. *ReLU* juga dapat menutupi kelemahan yang dimiliki oleh *Sigmoid* dan *Tanh*[26].

#### B. *Softmax*

*Softmax Classifier* merupakan fungsi yang digunakan untuk klasifikasi lebih dari dua kelas. Fungsi aktivasi ini biasanya digunakan pada *output layer*. *softmax* merupakan perhitungan kemungkinan dari masing-masing kelas target atau semua kelas target yang memungkinkan untuk menentukan kelas target untuk input yang diberikan. Keuntungan menggunakan *softmax* yaitu nilai rentang probabilitas yang dihasilkan dari 0 hingga 1, dan jumlah semua kemungkinan sama dengan satu. Ketika *softmax* digunakan untuk model klasifikasi multi, maka akan mengembalikan peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas lebih tinggi dari kelas yang lain.

#### 2.1.4.4. *Global Average Pooling*

*Global average pooling* merupakan operasi yang mereduksi suatu ukuran matriks menjadi matriks yang kecil [27]. Contohnya matriks  $7 \times 7$  diubah menjadi matriks  $1 \times 1$  dengan cara mengambil nilai rata-rata pada setiap matriks tersebut seperti yang ditunjukkan oleh Gambar 2.5. GAP adalah vektor dari nilai rata-rata feature maps dari lapisan konvolusional terakhir.  $GAP_i$  menunjukkan sebuah elemen dari vektor.  $M$  adalah jumlah baris dalam peta fitur dan  $n$  adalah jumlah kolom di peta fitur. Arti dari istilah kiri adalah menjumlahkan semua nilai dalam fitur tersebut memetakannya dan kemudian membaginya dengan  $m$  dikalikan dengan  $n$ . tujuannya adalah untuk mendapatkan nilai rata-rata dari feature maps. GAP menghitung rata-rata feature maps yang merupakan hasil dari proses konvolusional. Selanjutnya membuat vektor yang terdiri dari nilai rata-rata. GAP memiliki keunggulan dibandingkan fully connected layer. Pertama, komputasi biaya dapat dikurangi dengan mengurangi jumlah parameter yang akan ditangani (hyperparameters). Kedua, beberapa parameter model dapat dihilangkan untuk mengurangi overfitting. Oleh karena itu, tidak perlu bergantung pada Dropout. Konsep dari GAP ditunjukkan pada gambar 2.5.

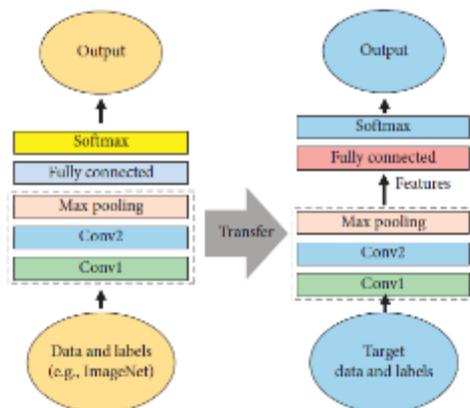


Gambar 2.5. Global Average Pooling

Sumber: [27]

**2.1.5. Transfer Learning**

*Transfer learning* merupakan metode menggunakan *neural network* yang telah dilatih sebelumnya kemudian mengurangi jumlah parameter dengan cara mengambil beberapa bagian dari model yang sudah dilatih sebelumnya untuk digunakan dalam mengenali model baru [28]. Gambar 2.6. menunjukkan empat langkah membuat model lengkap menggunakan transfer learning. Pertama, membangun arsitektur membuat model dan melatihnya dalam kumpulan data yang besar. Kedua, menghapus lapisan terakhir. Ketiga, mengganti dengan layer lain yang tugasnya menyelesaikan spesifik tugas. Keempat, melatih model baru dengan ukuran dataset yang relatif kecil yang sesuai untuk tujuan tersebut.



Gambar 2.6. Transfer Learning

Sumber: [28]

*Transfer learning* didasari fakta bahwa manusia dapat menerapkan pengetahuan yang sebelumnya telah dipelajari untuk memecahkan masalah baru

dengan lebih cepat serta dengan solusi yang lebih baik. Untuk mencapai kinerja yang lebih tinggi *neural network* sangat bergantung pada jumlah data. Berikut merupakan beberapa masalah yang dapat diatasi menggunakan *transfer learning*:

- A. Data, untuk mendapatkan hasil yang lebih baik *deep learning* membutuhkan banyak data. Untuk mendapatkan data berlabel akan membutuhkan waktu yang banyak.
- B. Komputasi, ketika kita memiliki data ribuan, untuk melatih jaringan saraf yang dalam menggunakan data yang besar akan sangat mahal dan membutuhkan waktu sehari-hari. Selain itu juga memerlukan proses berulang untuk mendapatkan hasil yang memuaskan.

Menurut [29], terdapat tiga pendekatan utama *transfer learning*. diantaranya Pretrained as a classifier, Pretrained as feature extractor dan fine-tuning. Pada dasarnya lapisan konvolusional pada lapisan bawah/awal merupakan proses ekstraksi fitur-fitur umum [29] Semakin dalam jaringan, maka fitur ekstraksi akan semakin detail. Dua faktor penting yang harus diperhatikan dalam penerapan transfer learning diantaranya:

- A. Ukuran dataset target (sedikit atau banyak).
- B. Kesamaan konteks dataset antara sumber dan target.

Tabel 2.1 menunjukkan aturan umum untuk melakukan *transfer learning*.

**Tabel 2.1. Memilih Pendekatan *Transfer Learning***

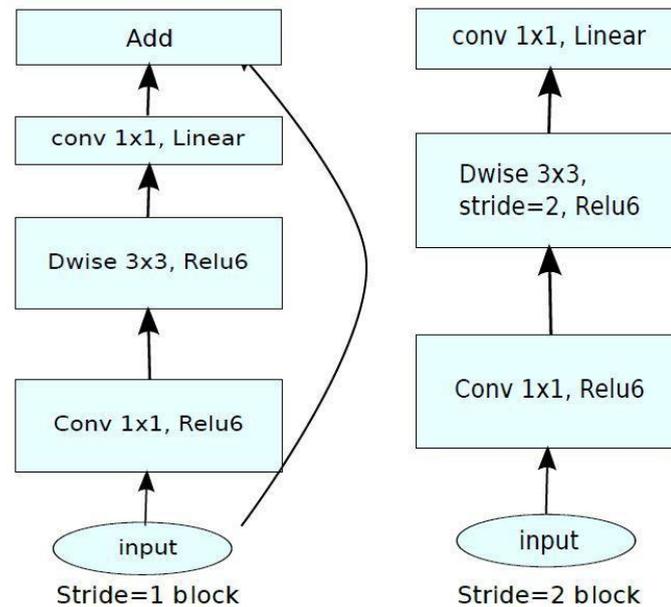
	Dataset Sangat Mirip	Dataset sangat Berbeda
Dataset Kecil	<i>Pre-trained as a feature extractor</i>	<i>Fine tune from activations earlier in the network</i>
Dataset Besar	<i>Fine tune through the full network</i>	<i>Fine tune through the entire network</i>

Sumber: [29]

### 2.1.6. MobileNetV2

MobileNetV2 merupakan model pengembangan dari MobileNetV1, pada percobaan klasifikasi gambar menggunakan ImageNet, MobileNetV2 menunjukkan hasil akurasi lebih baik dibandingkan MobileNetV1 dengan jumlah parameter yang

lebih sedikit [30]. Pada MobileNetV2, terdapat dua jenis blok seperti yang tercantum pada gambar 2.7 di bawah ini yakni blok residual dengan stride 1 dan dengan stride 2, kedua blok tersebut disusun membentuk arsitektur MobileNetV2.



Gambar 2.7. MobileNetV2

Sumber: [30]

### 2.1.7. Penyakit daun Kentang

Pada penelitian ini penyakit daun kentang dibagi menjadi dua kelas, yaitu busuk daun (*late blight*) dan bercak kering (*early blight*). Pada gambar 2.10 merupakan contoh dari penyakit busuk daun. Penyakit ini disebabkan oleh cendawan *Phytophthora infestans*. Patogen ditularkan melalui udara dan air. Gejala awal berupa bercak kebasah-basahan pada bagian tepi atau tengah daun. Bercak selanjutnya melebar dan terbentuk daerah nekrotik yang berwarna coklat. Bercak dikelilingi oleh masa sporangium yang berwarna putih dengan latar belakang hijau kelabu. Serangan dapat menyebar ke batang, tangkai, umbi dan buah. Serangan penyakit ini dapat berkembang dengan cepat pada musim hujan dengan kelembaban di sekitar kanopi >95% dengan suhu sekitar 20°C. Penyakit yang kedua adalah bercak daun atau penyakit bercak kering yang dapat dilihat pada Gambar 2.10. disebabkan oleh cendawan *Alternaria sp.* Patogen ditularkan melalui udara. Gejala awal timbulnya

bercak kecil di daun-daun bagian bawah, kemudian berkembang dengan diameter mencapai  $\pm 15$  mm. Warna bercak coklat dengan lingkaran-lingkaran sepusat. Masa konidia yang berwarna kelabu sampai hitam terlihat di atas bercak. Suhu optimum untuk perkembangan penyakit ini berkisar antara 28-30°C dengan kelembaban tinggi [31].



(a) (b)  
Gambar 2.8. *Late Blight* (a), *Early Blight* (b)

Sumber: [31]

## 2.2. Tinjauan Studi

Penelitian tentang daun kentang sudah banyak dilakukan, Berikut ini beberapa penelitian terkait dengan klasifikasi penyakit daun kentang:

1. Pada penelitian Tiwari dkk, sebanyak 2152 gambar daun kentang diambil dari dataset plant village yang terdiri dari 1000 gambar penyakit hawar awal, 1000 gambar penyakit busuk daun, dan 152 gambar daun kentang sehat. Set data dibagi menjadi dua bagian: bagian pelatihan terdiri dari 1700 gambar (70%) dan bagian pengujian berisi 452 gambar (30%). Berbagai model pra-terlatih seperti inceptionV3, VGG16, dan VGG19 digunakan untuk ekstraksi fitur di antaranya VGG19 memberikan hasil optimal. Pengklasifikasi berganda yaitu KNN, SVM, Neural Network, dan regresi logistik digunakan untuk klasifikasi. Diantaranya regresi logistik memberikan solusi mutakhir dengan akurasi klasifikasi 97,8% [32].
2. Pada penelitian Lee dkk, mengusulkan arsitektur CNN (convolutional neural network) yang sangat efisien yang cocok untuk deteksi penyakit kentang. Database dibuat untuk set pelatihan menggunakan pemrosesan gambar. Adam digunakan sebagai pengoptimal dan cross-entropy digunakan untuk analisis model. Softmax digunakan sebagai fungsi penilaian akhir. Lapisan konvolusi

dan sumber daya diminimalkan tetapi akurasi tetap terjaga. Hasil eksperimen menunjukkan bahwa model yang diusulkan mendeteksi penyakit tanaman dengan akurasi 99,53% dan mengurangi penggunaan parameter rata-rata 99,39% [33].

3. Pada penelitian Barman dkk, mengusulkan CNN (SBCNN) yang dibuat sendiri untuk deteksi penyakit daun kentang. SBCNN diterapkan secara terpisah di dataset gambar daun kentang ditambah dan tidak ditambah. Algoritma ini digunakan untuk melatih dan menguji citra daun kentang. Akurasi validasi terbaik dari SBCNN di non-augmented dan augmentasi data yang ditambah adalah 96,98% dan 96,75% dengan with akurasi pelatihan masing-masing 99,71% dan 98,75%. kesalahan pelatihan dan validasi dilaporkan di setiap zaman. Model SBCNN dilakukan dengan baik dalam kumpulan data yang diperbesar tanpa overfitting dalam model. Modelnya juga dibandingkan dengan kinerja arsitektur MobileNet untuk pengembangan aplikasi smartphone. Akhirnya, SBCNN (Augmented) dipilih sebagai model terbaik dibandingkan ke SBCNN (non-augmented) dan MobileNet. Modelnya adalah dikerahkan dalam aplikasi android untuk pengujian waktu nyata penyakit daun kentang dan dapat dianggap sebagai replika dari laboratorium patologi pertanian [13].
4. Penelitian Rozaki dkk, menggunakan metode *Transfer Learning* untuk menghindari model overfit ketika data yang digunakan tidak ideal atau terlalu sedikit. *Transfer Learning* merupakan metode yang menggunakan arsitektur CNN yang telah dilatih oleh data lain sebelumnya yang kemudian digunakan untuk klasifikasi citra pada data baru. Tujuan dari penelitian ini adalah menggunakan metode *Transfer Learning* pada arsitektur CNN untuk mengklasifikasikan citra daun kentang dalam mengidentifikasi penyakit daun kentang. Penelitian ini membandingkan metode Transfer Learning yang digunakan untuk menemukan metode terbaik. Hasil percobaan dalam penelitian ini menunjukkan bahwa *Transfer Learning* VGG-16 memiliki hasil performansi klasifikasi yang paling baik, metode ini menghasilkan nilai akurasi tertinggi yaitu 95% [34].

5. Pada penelitian Lee dkk, mengusulkan CNN (Convolutional Neural Network) arsitektur yang cocok untuk deteksi penyakit kentang. Pada awalnya, kita akan membuat database untuk set pelatihan kami melalui pemrosesan gambar di CNNnya. Adam digunakan sebagai pengoptimal, dan entropi silang adalah digunakan sebagai dasar analisis model. Softmax digunakan sebagai final fungsi penilaian. Lapisan konvolusi dan sumber daya adalah meminimalkan jumlah penggunaan sambil mempertahankan akurasi yang tinggi. Itu hasil percobaan menunjukkan bahwa penggunaan parameter adalah 10.089.219 dan keakuratan penilaian penyakit dapat mencapai 99% di bawah model preset yang diusulkan dalam makalah ini [35].
6. Dalam penelitian yang dilakukan Islam dkk, mendemonstrasikan bahwa teknik transfer learning dapat digunakan untuk deteksi dini penyakit kentang ketika sulit untuk mengumpulkan ribuan gambar daun baru. Pembelajaran transfer menggunakan model pembelajaran mendalam yang sudah terlatih berat untuk memecahkan masalah baru. Eksperimen termasuk gambar dari 152 daun sehat, 1000 daun busuk daun, dan 1000 daun awal daun busuk. Program memprediksi dengan akurasi 99,43% dalam pengujian dengan 20% data uji dan 80% data latih. Penelitian ini juga membandingkan model deep learning sekuensial dengan beberapa model pembelajaran sebelumnya model menerapkan pembelajaran transfer dan menemukan bahwa transfer learning memberikan hasil terbaik sampai saat ini. Keluaran kami menunjukkan bahwa transfer learning mengungguli semua karya yang ada di deteksi penyakit kentang [2].

**Tabel 2.2. Penelitian Sebelumnya**

<b>Judul Penelitian</b>	<b>Tahun/ Quartile/ Akreditasi</b>	<b>Kelebihan</b>	<b>Perbedaan</b>
Potato Leaf Diseases Detection Using Deep Learning	2020, IEEE Access	model disajikan menggunakan model pre-trained seperti VGG19 untuk fine-tuning (Transfer	Tidak melakukan augmentasi citra

		Learning) untuk mengekstrak fitur yang relevan dari dataset. Kemudian diklasifikasi dengan beberapa metode machine learning.	
High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network	2021, Springer	mengusulkan arsitektur CNN (convolutional neural network) sendiri. Pre-processing dilakukan augmentasi. Hasilnya model yang diusulkan lebih unggul dibandingkan VGG16 dan VGG19	Menggunakan arsitektur sendiri
Comparative Assessment of Deep Learning to Detect the Leaf Diseases of Potato based on Data Augmentation	2020, IEEE	mengusulkan model CNN (SBCNN) yang dibuat sendiri untuk deteksi penyakit daun kentang. Penelitian dilakukan dengan menggunakan data yang di augmentasi dan tidak di augmentasi. Modelnya diimplementasikan dalam aplikasi android untuk pengujian secara <i>real-time</i> .	Menggunakan arsitektur sendiri
Implementation of Transfer Learning in the Convolutional	2021, Procedia of Engineering	menggunakan metode <i>Transfer Learning</i> VGG16 untuk menghindari model overfit ketika data yang	Hanya menggunakan sebagian dataset

Neural Network Algorithm for Identification of Potato Leaf Disease	and Life Science	digunakan tidak ideal atau terlalu sedikit.	(jauh lebih sedikit)
Health Detection for Potato Leaf with Convolutional Neural Network	2020, IEEE Access	mengusulkan arsitektur CNN (convolutional neural network) sendiri. Hasilnya model yang diusulkan lebih unggul dibandingkan VGG16 dan VGG19	Tidak dilakukan augmentasi
Application of Transfer Learning to Detect Potato Disease from Leaf Image	2019, IEEE Access	mendemonstrasikan bahwa teknik <i>transfer learning</i> dapat digunakan untuk deteksi dini penyakit kentang ketika sulit untuk mengumpulkan ribuan gambar daun baru.	Tidak dilakukan augmentasi citra.

Sumber: Hasil Penelitian

Pada Tabel 2.2 dijelaskan tentang penelitian sebelumnya yang sudah dilakukan, dan pada penelitian ini akan dilakukan klasifikasi daun kentang dengan melakukan beberapa tahap, tahap yang pertama, akan dilakukan augmentasi citra pada kelas healthy, tujuan dari tahapan ini adalah agar data menjadi seimbang setelah itu dilakukan resize kemudian dilakukan proses klasifikasi. Semua metode yang dipakai dalam penelitian ini belum diterapkan pada penelitian sebelumnya.

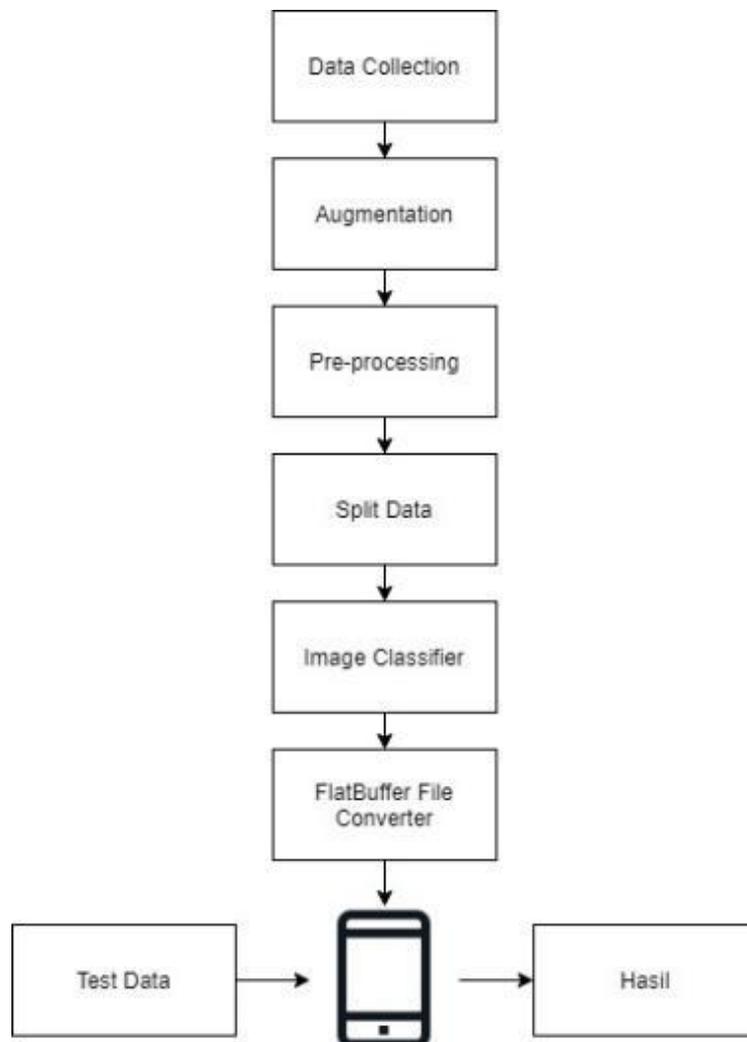
### 2.3. Objek Penelitian

Objek dari penelitian yang dilakukan adalah menggunakan data publik dari kaggle yang merupakan database yang tersedia untuk umum. Dataset ini berisi 2152 citra daun kentang dengan gambar berwarna. Resolusi citra adalah  $256 \times 256$  piksel. Dataset tersebut memiliki tiga kelas: Early Blight, Healty, Late Blight.

### BAB III

## METODOLOGI PENELITIAN

Penelitian merupakan kegiatan mencari, mencatat, merumuskan, dan menganalisis hingga menyusun laporan dan pada penelitian ini digunakan metode eksperimen. Penelitian eksperimen merupakan metode penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi yang terkendalikan [36]. Gambar 3.1 merupakan diagram model desain penelitian yang dilakukan dalam penelitian klasifikasi penyakit daun kentang:



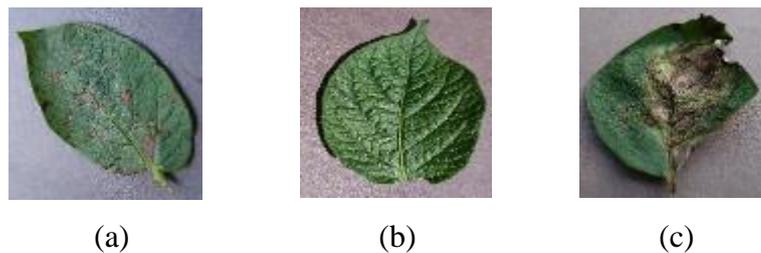
Gambar 3.1. Model Desain Penelitian

Sumber: Hasil Penelitian

Diagram pada Gambar 3.1 menunjukkan secara umum proses klasifikasi penyakit daun kentang. pada penelitian ini terdiri atas tujuh proses penting. Ketujuh proses penting itu yang pertama adalah pengumpulan dataset, selanjutnya adalah augmentasi citra, dilanjutkan *preprocessing*, kemudian dilanjutkan proses split data untuk menentukan data *training*, *validation* dan *testing*, selanjutnya adalah proses *image classifier*, setelah itu mengubah model menjadi file *FlatBuffer* (.tflite) dan terakhir hasilnya ditampilkan dalam perangkat seluler.

### 3.1. Data Collection

Pada penelitian ini digunakan citra dari data publik. citra daun kentang diperoleh dari kaggle yang merupakan database yang tersedia untuk umum. Dataset ini berisi 2152 gambar daun kentang gambar berwarna. Resolusi gambar adalah  $256 \times 256$  piksel.



Gambar 3.2. Citra Asli (a) *Early Blight*, (b) *Healthy* dan (c) *Late Blight*

Sumber: <https://www.kaggle.com/sohaibalam67/potato-disease>

Gambar 3.2. merupakan contoh citra asli daun kentang yang akan digunakan pada penelitian.

**Tabel 3.1. Dataset**

Label	Kategori	Jumlah
1	Early Blight	1000
2	Healthy	152
3	Late Blight	1000
Total		2152

Sumber: Hasil Penelitian

Tabel 3.1 menunjukkan terdapat 2152 dataset daun kentang yang terbagi menjadi 3 kelas yakni Early Blight, Healthy dan Late Blight.

### 3.2. *Augmentation*

Salah satu batasan terbesar dalam pengembangan model *deep learning* yaitu membutuhkan banyak himpunan data. Jutaan atau bahkan lebih banyak sampel data diperlukan untuk membuat model *deep learning* yang andal. Keterbatasan dapat diatasi dengan memanipulasi dan mentransformasikan sejumlah kecil data. Teknik ini disebut augmentasi data. Teknik augmentasi data telah digunakan di banyak penelitian [11],[12]. Teknik termasuk *random cropping*, *horizontal flipping*, modifikasi kecerahan, dan modifikasi kontras. Selain itu augmentasi juga biasanya digunakan dalam menangani *imbalance* data. Pada penelitian ini, citra yang ada pada kelas *early blight* dan *late bligh* lebih banyak dibandingkan dengan yang sehat. ini adalah salah satu jenis ketidakseimbangan himpunan data. Hal ni dapat menyebabkan model yang diterapkan menjadi *overfitting* atau *underfitting*. Untuk mengatasi masalah pada penelitian ini dilakukan augmentasi hanya pada kelas daun sehat.

### 3.3. *Pre-processing*

Setelah proses augmentasi selanjutnya dilakukan *pre-processing*. Semua citra daun kentang di dalam dataset merupakan citra berwarna dengan ukuran dimensi 256x256. Pada penelitian ini ukuran citra diubah menjadi dimensi baru 224x224. Tujuan dilakukan *resize* citra menjadi dimensi lebih kecil adalah untuk membantu kinerja dari model agar melakukan pekerjaan dengan tepat dengan waktu yang lebih rendah. Setelah dilakukan *resize*, selanjutnya dilakukan konversi BGR ke RGB. Konversi ini dilakukan karena OpenCV menggunakan format gambar BGR. sehingga, ketika kita membaca gambar menggunakan `cv2.imread()` maka itu akan menafsirkan dalam format BGR secara default. Kita dapat menggunakan metode `cvtColor(im, cv2.COLOR_BGR2RGB)` untuk mengonversi gambar BGR ke RGB dan sebaliknya. Setelah itu selanjutnya dilakukan *rescale* =  $1/255$ . Hal ini dilakukan untuk mengubah ukuran data piksel RGB citra (0-255) menjadi rentang angka (0-1) untuk memudahkan proses training data.

### 3.4. *Split Data*

Setelah dilakukan *rescale*, selanjutnya dilakukan *split* data. Pada penelitian ini dataset dibagi menjadi tiga bagian, yakni data *training*, validasi dan *testing*. Data *training* merupakan data yang digunakan untuk melatih algoritma klasifikasi, data

validasi digunakan untuk proses validasi model dan data *testing* digunakan untuk mengetahui performa dari algoritma klasifikasi yang sudah dilatih sebelumnya. Pada awalnya data dibagi menjadi 2 bagian, yaitu data train dan data test, dengan perbandingan 90% untuk data train dan 10% untuk data test. Setelah itu data train dibagi menjadi 2 bagian, yaitu data train dan data validasi, data validasi diambil sebanyak 300 citra dari data train atau sebanyak 10% dari jumlah data. Distribusi citra terdapat pada tabel 3.2.

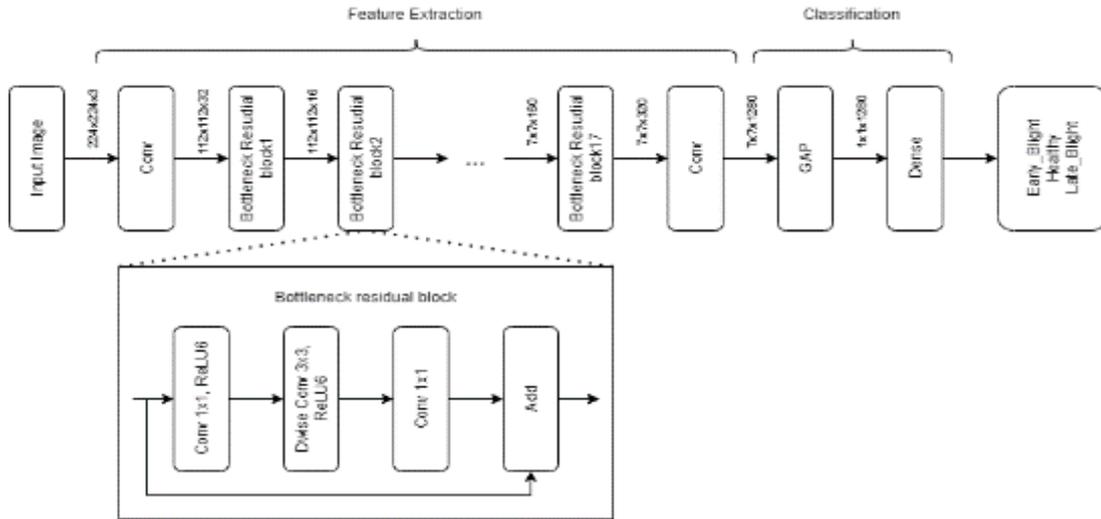
**Tabel 3.2. Data Train, Valid dan Test**

Dataset	Jumlah
Train	2400
Valid	300
Test	300

Sumber: Hasil Penelitian

### 3.5. *Image Classifier*

Setelah tahap *pre-processing*, selanjutnya adalah proses klasifikasi citra dengan menggunakan model *Transfer Learning* MobileNetV2, Pada tahap ini terdapat dua langkah, langkah pertama merupakan ekstraksi fitur dan langkah kedua merupakan proses klasifikasi. Untuk ekstraksi fitur yang dilakukan adalah mengimport model MobilenetV2 dari pustaka keras. Model yang diimport menggunakan lapisan konvolusional yang diinisialisasi dengan bobot yang telah ditetapkan dan dilatih sebelumnya menggunakan ImageNet. Selanjutnya, untuk klasifikasi digunakan *Global Average Pooling layer* dan *dense layer* dengan *SoftMax* ditambahkan setelah *convolutional layer*, bukan *fully connected layer*. Menurut [37] *fully connected layer* biasanya digunakan dalam model CNN tradisional dan cenderung overfitting meski menggunakan dropout. Maka dari itu pada penelitian ini digunakan *Global Average Pooling (GAP)* yang memasukkan nilai rata-rata setiap fitur dan memetakan ke dalam vektor dan menautkannya ke masukan lapisan *SoftMax* secara langsung. Gambar 3.3. Merupakan gambaran proses *image classifier*.



Gambar 3.3. Proses *Image Classifier*

Sumber: Hasil Penelitian

Fondasi utama dari arsitektur MobileNetV2 merupakan *bottleneck residual block* yang ditunjukkan pada tabel 3.3. dengan  $h, w, d, t$  dan  $s$  merupakan panjang, lebar, kedalaman, faktor ekspansi dan *stride* secara berurutan. Jika dibandingkan dengan residual block milik pada model ResNet [38], *bottleneck residual block* lebih efisien secara memori dan resistan terhadap hilangnya fitur penting selama proses pelatihan model (*vanishing gradient*) [39]. Jenis *convolution layer* yang digunakan dalam block ini adalah *depthwise (dwise) separable convolution layer* yang menggunakan 8 hingga 9 kali lebih sedikit komputasi dibandingkan *convolution layer* standar [30].

Tabel 3.3. *Bottleneck Resudial Block*

Input	Operator	Output
$h \times w \times d$	1 x 1 conv2d, ReLU6	$h \times w \times t*d$
$h \times w \times t*d$	3 x 3 dwise stride = s, ReLU6	$h/s \times w/s \times t*d$
$h/s \times w/s \times t*d$	Linear 1 x 1 conv2d	$h/s \times w/s \times d'$

Sumber: Hasil Penelitian

Arsitektur MobileNetV2 menggunakan resolusi input 224 x 224 pixel, ditunjukkan pada tabel 3.4. Resolusi ini dipilih karena 224 x 224 pixel merupakan resolusi standar dari model MobileNetV2 yang telah dilatih pada dataset ImageNet.

Setiap *layer bottleneck* pada arsitektur tersebut dibangun oleh *bottleneck residual block* sebanyak  $n$ . *Bottleneck residual block* pertama untuk setiap layer bottleneck memiliki stride dengan nilai  $s$ , dan selanjutnya menggunakan stride dengan nilai 1. Kanal/kedalaman dari output pada setiap layer dilambangkan dengan  $c$ . Dengan pengecualian untuk layer pertama, expansion rate ( $t$ ) dengan nilai 6 digunakan pada seluruh layer. Dengan model ini tiap citra daun kentang akan direpresentasikan menjadi 1280 fitur pada *global average pooling*. 1280 fitur ini akan menjadi input pada layer output/*classifier*. pada layer classifier digunakan fungsi aktivasi softmax yang menghasilkan kelas objek yang akan diklasifikasi.

**Tabel 3.4. Arsitektur MobileNetV2**

Input	Layer	$t$	$c$	$N$	$s$
224 x 224 x 3	Conv2d	-	32	1	2
112 x 112 x 32	Bottleneck	1	16	1	1
112 x 112 x 16	Bottleneck	6	24	2	2
56 x 56 x 24	Bottleneck	6	32	3	2
28 x 28 x 32	Bottleneck	6	64	4	2
14 x 14 x 64	Bottleneck	6	96	3	1
14 x 14 x 96	Bottleneck	6	160	3	2
7 x 7 x 160	Bottleneck	6	320	1	1
7 x 7 x 320	Conv2d 1x1	-	1280	1	1
7 x 7 x 1280	GlobalAvgPool	-	1280	1	-
1280	Output/Classifier	-	3	-	-

Sumber: Hasil Penelitian

*Hyperparameter* yang akan digunakan dalam pelatihan model MobileNetV2 dapat dilihat pada tabel 3.5. Model MobileNetV2 dilatih dengan *epoch* berjumlah 30 dan *batch size* sebesar 32. *Optimizer* yang dipilih adalah Adam. “categorical\_crossentropy” digunakan sebagai *loss function* Pada setiap akhir *epoch*, nilai akurasi dari hasil klasifikasi akan dihitung dan bobot dari MobileNetV2 akan disimpan sebagai *checkpoint* model apabila nilai akurasi yang didapatkan lebih tinggi dibandingkan *checkpoint* sebelumnya. Dengan skema pelatihan seperti ini, akan

didapatkan model yang dapat memberikan nilai akurasi tertinggi pada proses pelatihan.

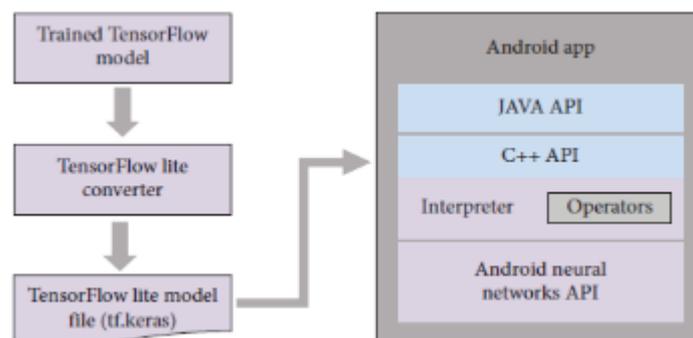
**Tabel 3.5. Hyperparameter yang digunakan**

Hyperparameter	Nilai/Jenis yang Digunakan
Jumlah Epoch	30
Batch Size	32
Optimizer	Adam
Loss Function	categorical_crossentropy
Model Checkpoint	Model dengan akurasi tertinggi

Sumber: Hasil Penelitian

### 3.6. FlatBuffer File Converter

Untuk mengembangkan aplikasi, TensorFlow Lite menyediakan sebuah metode yang mengubah model yang dihasilkan menjadi Tensor- File berformat Flow Lite FlatBuffer (.tflite), yang dapat digunakan di perangkat seluler. FlatBuffer adalah sebuah *open-source cross-platform serialization library* yang membuat serial data secara efisien. TensorFlow Lite mendukung konversi file yang dibuat oleh TensorFlow, fungsi konkret, dan Keras [40]. Pada penelitian ini file model yang diubah ini dimasukkan kedalam proyek demo yang disediakan oleh TensorFlow Lite. Setelah langkah ini, selanjutnya membuat android package file (APK) dan menginstal aplikasi di perangkat seluler. Gambar 3.4. menunjukkan keseluruhan proses.



**Gambar 3.4. Proses Konversi Tensorflow Lite**

Sumber: Hasil Penelitian

### **3.7. Evaluasi dan Validasi**

Pada tahap ini dilakukan proses pengujian metode yang diusulkan dengan melakukan deteksi penyakit daun kentang pada citra baru, sehingga diketahui seberapa tepat metode yang diusulkan untuk melakukan identifikasi penyakit daun kentang.

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

#### **4.1. Hasil Penelitian**

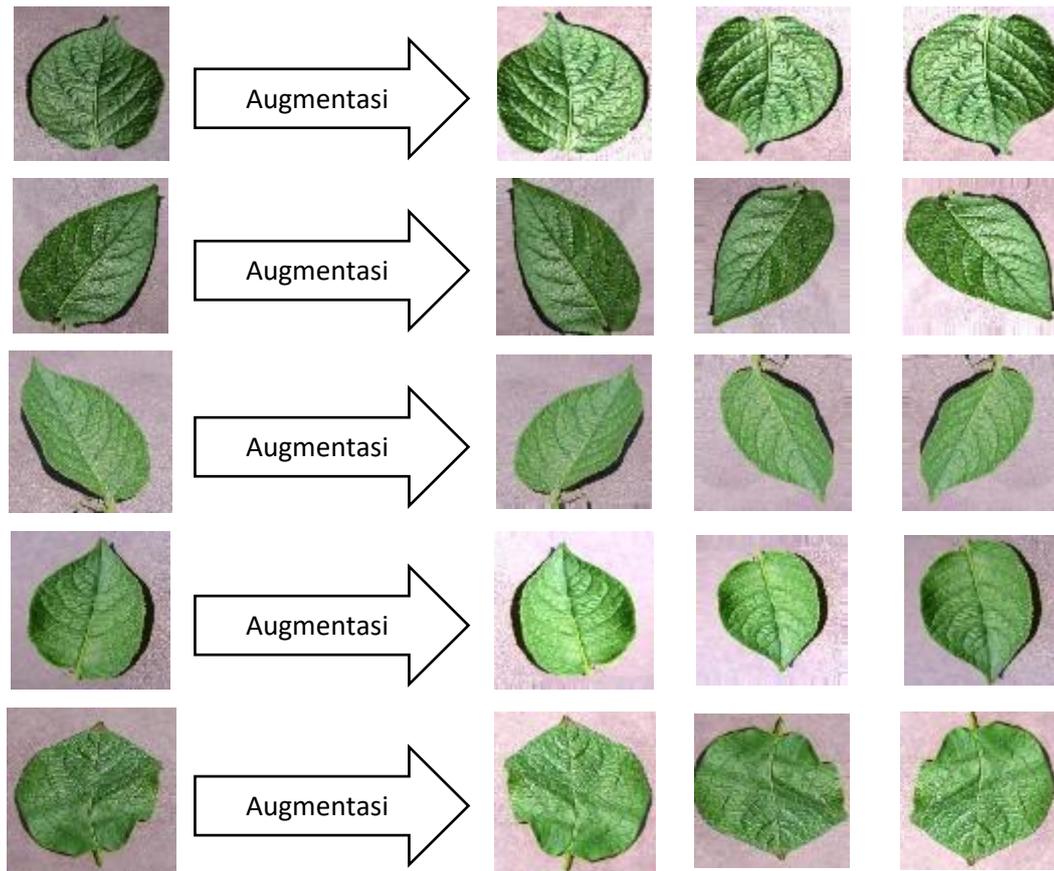
Pada tahapan penelitian ini membahas hasil dari penerapan metode-metode yang diusulkan untuk mengkasifikasi penyakit daun kentang, sampai kepada implementasi antar muka menggunakan aplikasi Android.

##### **4.1.1. *Augmentation***

Pada penelitian ini augmentasi citra dilakukan menggunakan enam teknik, proses augmentasi ini tidak dilakukan pada penelian sebelumnya yaitu penelitian [2][41][33], hasil dari augmentasi pada penelitian ini digunakan untuk mengatasi ketidakseimbangan data. Jumlah citra daun sehat yang semula hanya 152 dilakukan augmentasi sehingga jumlahnya menjadi seimbang dengan 2 kelas lainnya yakni 1000 citra. Dataset daun kentang sehat diaugmentasi dengan menggunakan:

- horizontal flip = true. Berfungsi untuk membalik citra secara horizontal secara acak.
- vertical flip = true. Berfungsi untuk membalik citra secara vertikal secara acak.
- zoom\_range = 0.2. Berfungsi untuk membesarkan citra gambar, 0.2 menunjukkan intensitas pembesaran pada citra.
- fill\_mode = 'nearest'. Berfungsi untuk mengisi area kosong. "nearest" artinya hanya menggantikan area kosong dengan nilai piksel terdekat.
- brightness\_range = [0.5, 1.5]. Berfungsi untuk mengubah kecerahan secara acak. [0.5, 1.5] merupakan nilai pergeseran kecerahan. Nilai kurang dari 1,0 digunakan untuk menggelapkan gambar, sedangkan nilai di atas 1,0 digunakan untuk mencerahkan gambar.

Hasil augmentasi ditunjukkan pada Gambar 4.1.



Gambar 4.1. Hasil Augmentasi

Sumber: Hasil Penelitian

Tabel 4.1. menunjukkan jumlah dataset setelah dilakukan proses augmentasi.

**Tabel 4.1. Dataset Hasil Augmentasi**

Label	Kategori	Jumlah
1	Early Blight	1000
2	Healthy	1000
3	Late Blight	1000
Total		3000

Sumber: Hasil Penelitian

#### 4.1.2. Pre-Processing

Tahapan *pre-processing* dilakukan dengan *resize*, *rescale* dan mengkonversi citra BGR menjadi RGB. Beberapa citra hasil konversi BGR ke RGB dapat dilihat pada tabel 4.2.

Tabel 4.2. Hasil Konversi BGR ke RGB

Citra Awal	BGR	RGB
		
		
		
		
		

Sumber: Hasil Penelitian

Tabel 4.2. menunjukkan hasil konversi citra dari RGB ke RGB, terlihat perubahan warna dari citra tersebut.

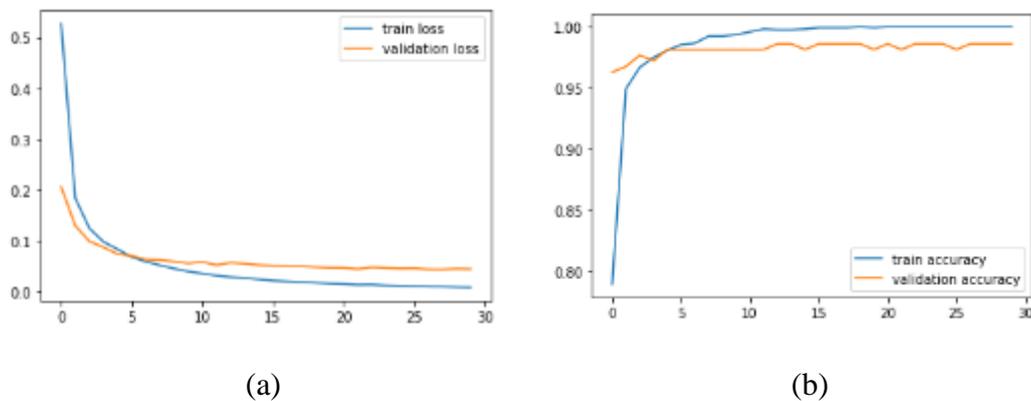
#### 4.1.3. *Image Classifier*

Penelitian ini melakukan beberapa percobaan sesuai dengan tahapan penelitian, dan percobaan menggunakan beberapa metode *Transfer Learning* diantaranya MobileNetV2, InceptionResNetV2, VGG16 dan Inception V3.

##### 4.1.3.1. Pengujian Model dengan *Transfer Learning* MobileNetV2

Pengujian Model *transfer learning* MobileNetV2 dilakukan dengan 2 percobaan. Percobaan pertama dilakukan menggunakan metode *transfer learning*

MobileNetV2 dengan dataset asli atau tanpa dilakukan augmentasi, percobaan kedua menggunakan metode *transfer learning* MobileNetV2 dengan dataset yang telah dilakukan augmentasi pada kelas daun sehat. Gambar 4.2. menunjukkan grafik performa hasil *loss* dan *accuracy* dari arsitektur MobileNetv2 dengan dataset asli atau tidak dilakukan augmentasi, dimana garis biru menunjukkan data *training* dan warna oranye menunjukkan data *validation*. Pada pengujian ini, model MobileNetV2 telah dilatih menggunakan dataset daun kentang. Hasil klasifikasi pada data *training* dan data *validation* menggunakan model *transfer learning* MobileNetV2 menunjukkan performa yang baik. Jika dilihat pada grafik nilai *loss* (a), *training loss* dan *validation loss* menurun secara stabil. Begitu pula jika dilihat pada grafik nilai *accuracy* (b), nilai akurasi pada data *training* dan data *validation* meningkat secara stabil.



Gambar 4.2. Hasil dari model klasifikasi MobileNetV2 tanpa Augmentasi (a) *loss* (b) *accuracy*

Sumber: Hasil Penelitian

Dengan pertimbangan nilai *validation loss* yang stabil setelah *epoch* ke-5 dan tidak mengalami penurunan hingga *epoch* ke-30 serta nilai akurasi tertinggi yang didapatkan ada pada *epoch* ke-14, maka akan dipilih jumlah *epoch* maksimal sebesar 30. *Batch size* yang digunakan adalah 32, dengan optimizer adam, *loss function* yang digunakan adalah *categorical cross entropy* dan model *checkpoint* menggunakan *save best only*.

Tabel 4.3. menunjukkan hasil evaluasi dan *confusion matrix* prediksi model MobileNetV2 pada data *testing*. Hasil prediksi model MobileNetV2 pada data *testing* mendapatkan nilai akurasi 97,6%. Dari tabel 4.3 dapat dilihat bahwa model memberikan prediksi yang baik pada kelas Early blight. Sedangkan Pada kelas late

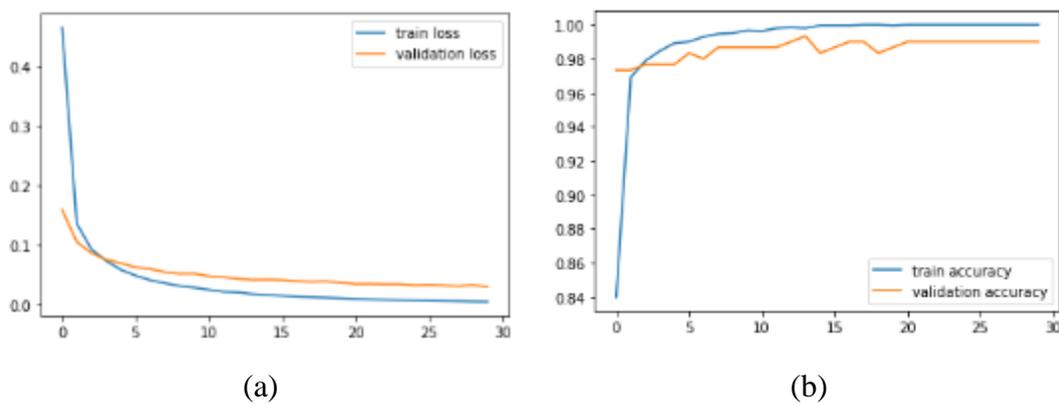
bligh terdapat 1 kesalahan yakni 1 citra terdeteksi sehat dan pada kelas sehat sebanyak 3 citra terdeteksi Late Blight.

**Tabel 4.3. Confusion Matrix MobileNetV2 Tanpa Augmentasi**

	Early Blight	Late Blight	Healthy
Early Blight	103	0	0
Late Blight	1	99	1
Healthy	0	3	9

Sumber: Hasil Penelitian

Dari hasil konfusi matrix yang ditunjukkan pada Tabel 4.3 tersebut dapat disimpulkan bahwa dataset training untuk kelas sehat sangat kurang sehingga banyak kesalahan dalam deteksi. Maka dari itu dilakukan augmentasi pada kelas sehat untuk menangani terjadinya imbalance data. Gambar 4.3 menunjukkan grafik performa hasil *loss* dan *accuracy* dari arsitektur MobileNetv2 yang dilatih dengan dataset yang sudah di aumentasi, Hasil klasifikasi pada data *training* dan data *validation* menggunakan model *transfer learning* MobileNetV2+augmentasi menunjukkan performa yang baik. Jika dilihat pada grafik nilai *loss* (a), *training loss* dan *validation loss* menurun secara stabil. Begitu pula jika dilihat pada grafik nilai *accuracy* (b), nilai akurasi pada data *training* dan data *validation* meningkat secara stabil.



Gambar 4.3. Hasil dari model klasifikasi MobileNetV2 dengan Augmentasi (a) *loss* (b) *accuracy*

Sumber: Hasil Penelitian

Tabel IV.3. menunjukkan hasil evaluasi dan *confusion matrix* prediksi model MobileNetV2 pada data *testing*. Hasil prediksi model MobileNetV2 dengan dataset yang telah dilakukan augmentasi mendapatkan nilai akurasi 99,6%. Dari tabel 4.4

dapat dilihat bahwa model memberikan prediksi yang baik pada kelas Early blight dan *Healthy*. Hanya terdapat satu kesalahan saja yakni satu citra late Blight terdeteksi sehat.

**Tabel 4.4. Confusion Matrix MobileNetV2 dengan Augmentasi**

	Early Blight	Late Blight	Healthy
Early Blight	107	0	0
Late Blight	0	99	1
Healthy	0	0	93

Sumber: Hasil Penelitian

Pada tabel 4.4 dijelaskan hasil perbandingan dari nilai akurasi yang didapatkan dengan menggunakan *Confusion Matrix*. Akurasi dari klasifikasi dengan model *transfer learning* MobileNetV2 menggunakan dataset asli menghasilkan 97,6%, sedangkan model *transfer learning* MobileNetV2 menggunakan dataset yang telah dilakukan augmentasi pada kelas sehat mendapat akurasi sebesar 99,6%.

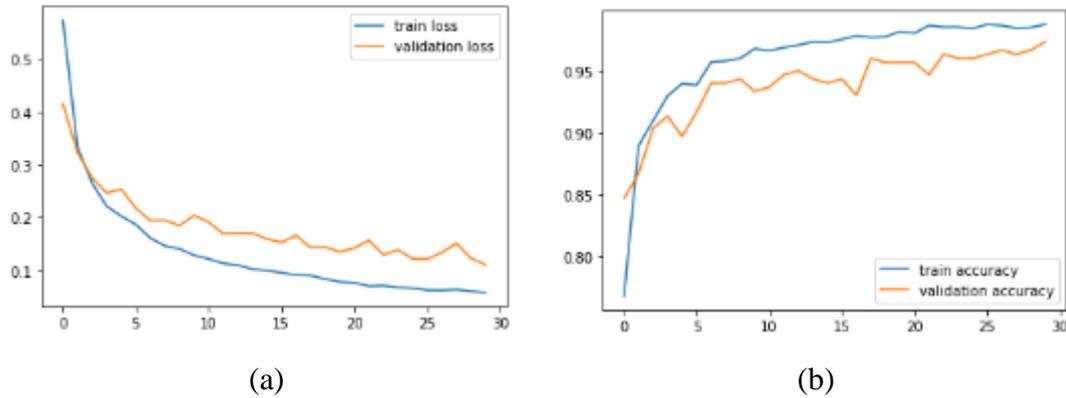
**Tabel 4.5. Perbandingan Kinerja MobileNetV2 dengan distribusi data asli dan data hasil augmentasi**

Distribusi Data	Prediksi Benar	Prediksi Salah	Akurasi
Asli	211	5	97,6%
Hasil Augmentasi	299	1	99.6%

Sumber: Hasil Penelitian

#### **4.1.3.2. Pengujian Model dengan *Transfer Learning* InceptionResNetV2, VGG16 dan Inception V3**

Percobaan selanjutnya dengan melakukan klasifikasi dengan menggunakan model *Transfer Learning* InceptionResNetV2, VGG16 dan Inception V3 dengan menggunakan *hyperparameter* yang sama dengan yang diterapkan pada percobaan menggunakan *transfer learning* MobileNetV2.



Gambar 4.4. Hasil dari model klasifikasi transfer learning InceptionResNetV2 (a) *loss* (b) *accuracy*

Sumber: Hasil Penelitian

Gambar 4.4. menunjukkan grafik performa hasil *loss* dan *accuracy* dari arsitektur InceptionResNetV2, Jika dilihat pada grafik nilai *loss* (a), *training loss* dan *validation loss* menurun kurang stabil. Begitu pula jika dilihat pada grafik nilai *accuracy* (b), nilai akurasi pada data *training* dan data *validation* meningkat kurang stabil.

Tabel 4.6 menunjukkan hasil evaluasi dan *confusion matrix* prediksi model InceptionResnetV2. Hasil prediksi memperoleh nilai akurasi 96,6%. Dari tabel 4.6 dapat dilihat bahwa model memberikan prediksi yang kurang baik. pada kelas Early blight terdapat satu citra terdeteksi kelas Late Blight. Kemudian pada kelas late blight ada enam citra yang tidak tepat deteksinya yakni tiga citra terdeteksi early blight dan tiga citra terdeteksi sehat. Kemudian pada kelas sehat terdapat tiga citra yang salah deteksi, yakni satu citra terdeteksi early blight dan 2 kelas terdeteksi late blight.

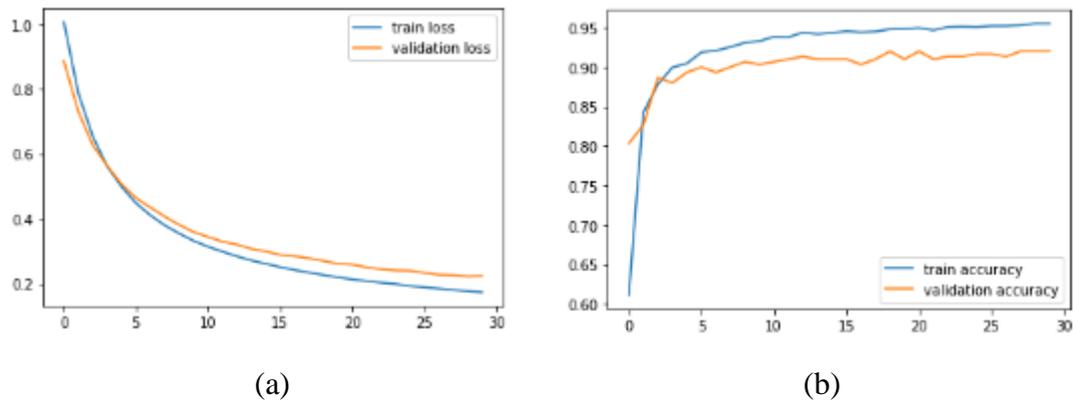
**Tabel 4.6. Confusion Matrix Model InceptionResNetV2**

	Early Blight	Late Blight	Healthy
Early Blight	106	1	0
Late Blight	3	94	3
Healthy	1	2	90

Sumber: Hasil Penelitian

Percobaan selanjutnya dengan melakukan klasifikasi dengan menggunakan model *Transfer Learnig* VGG16. Gambar 4.5 menunjukkan grafik performa hasil *loss* dan *accuracy* dari arsitektur VGG16, Jika dilihat pada grafik nilai *loss* (a), *training loss* dan *validation loss* menurun sangat stabil. Begitu pula jika dilihat pada grafik

nilai accuracy (b), nilai akurasi pada data *training* dan data *validation* meningkat stabil.



Gambar 4.5. Hasil dari model klasifikasi VGG16 (a) loss (b) accuracy

Sumber: Hasil Penelitian

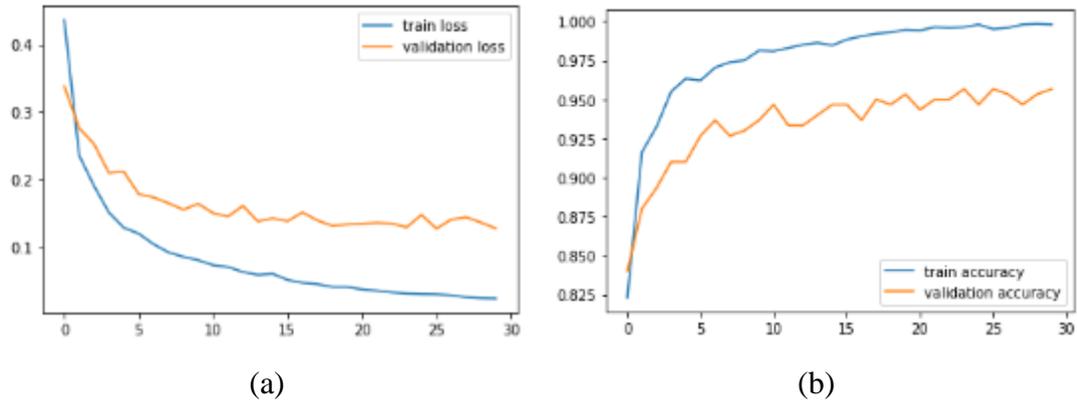
Tabel 4.7 menunjukkan hasil evaluasi dan *confusion matrix* prediksi model VGG16. Hasil prediksi memperoleh nilai akurasi 96,3%. Dari tabel 4.7 dapat dilihat bahwa model memberikan prediksi yang kurang baik.

**Tabel 4.7. Confusion Matrix Model VGG16**

	Early Blight	Late Blight	Healthy
Early Blight	103	4	0
Late Blight	4	94	2
Healthy	0	1	92

Sumber: Hasil Penelitian

Percobaan selanjutnya dengan melakukan klasifikasi dengan menggunakan model *Transfer Learning* InceptionV3. Gambar 4.6 menunjukkan grafik performa hasil *loss* dan *accuracy* dari arsitektur InceptionV3, Jika dilihat pada grafik nilai *loss* (a), *training loss* dan *validation loss* memiliki perbedaan yang sangat jauh. Begitu pula jika dilihat pada grafik nilai accuracy (b). Perbedaan nilai yang lebar antara nilai akurasi dan *loss* merupakan bukti bahwa performa klasifikasi mengalami *overfitting* dan ini adalah hasil yang kurang baik dari model klasifikasi.



Gambar 4.6. Hasil dari model klasifikasi InceptionV3 (a) loss (b) accuracy

Sumber: Hasil Penelitian

Tabel 4.8 menunjukkan hasil evaluasi dan *confusion matrix* prediksi model InceptionV3. Hasil prediksi memperoleh nilai akurasi 95,6%. Dari tabel 4.8 dapat dilihat bahwa model memberikan prediksi yang kurang baik.

**Tabel 4.8. Confusion Matrix Model InceptionV3**

	Early Blight	Late Blight	Healthy
Early Blight	102	4	1
Late Blight	3	95	2
Healthy	0	3	90

Sumber: Hasil Penelitian

Pada tabel 4.9 dijelaskan hasil perbandingan dari nilai akurasi yang didapatkan dengan menggunakan *Confusion Matrix*. Akurasi dari klasifikasi dengan model *transfer learning* MobileNetV2 menghasilkan akurasi paling tinggi yaitu 99,6% dengan waktu training paling cepat yaitu 5 menit.

**Tabel 4.9. Perbandingan hasil pengujian**

Model Klasifikasi	Prediksi Benar	Prediksi Salah	Akurasi	Waktu
MobileNetV2	299	1	99,6%	5m
InceptionResNetV2	292	8	96.6%	25m
VGG16	289	11	96,3%	14m
InceptionV3	287	13	95%	10m

Sumber: Hasil Penelitian

## 4.2. Implementasi Sistem

Implementasi antar muka ialah penerapan model yang digunakan untuk mengklasifikasi daun kentang lalu diterapkan pada aplikasi android. *Flatform* android dipilih untuk memudahkan dalam mendapatkan citra gambar, karena pada android dapat langsung menggunakan kamera untuk mendapatkan citra daun kentang.

### 4.2.1. Tampilan Home

Tampilan Home merupakan halaman pertama yang akan diakses saat sistem dijalankan. Pada menu utama ini terdapat judul dari sistem yang telah dibuat serta terdapat beberapa informasi mengenai penyakit daun kentang.

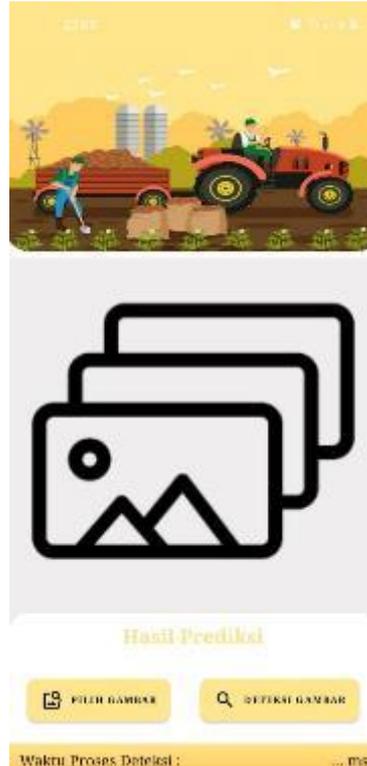


Gambar 4.7. Tampilan *Home*

Sumber: Hasil Penelitian

### 4.2.2. Halaman Deteksi dari Galeri

Pada halaman Deteksi dari galeri inilah citra uji dapat dideteksi dengan cara mengambil gambar dari galeri, halaman ini berisi 2 tombol dan tampilan untuk menampilkan hasil waktu pemrosesan, dapat dilihat pada gambar dibawah ini.



Gambar 4.8. Halaman Deteksi dari Galeri

Sumber: Hasil Penelitian

Pada Gambar 4.8 dapat dilihat bahwa halaman deteksi dari galeri berisi tombol untuk memilih gambar dari galeri ini berfungsi untuk menginput citra uji kedalam sistem, lalu tombol deteksi gambar digunakan untuk memulai mendeteksi gambar setelah di tekan tombol tersebut ditekan maka akan menampilkan hasil deteksi dari citra uji tersebut.

#### 4.2.3. Halaman Deteksi dengan Kamera

Halaman Deteksi dengan Kamera merupakan halaman yang digunakan untuk deteksi penyakit daun kentang *secara real-time* dan hasilnya akan muncul ketika kita klik tombol capture.

Pada menu deksi dengan kamera flash diatur secara otomatis agar perubahan cahaya tidak berbeda secara signifikan saat berada diluar ruangan yang banyak cahaya maupun berada di dalam ruangan yang kurang cahaya. Flash akan menyala dan mati otomatis berdasarkan intensitas cahaya yang ada.



Gambar 4.9. Halaman Deteksi dengan Kamera

Sumber: Hasil Penelitian

Pada Gambar 4.9 dapat terlihat pada halaman ini menampilkan 3 hasil prediksi dari deteksi citra uji dan juga menampilkan waktu proses deteksi.

#### 4.2.4. Halaman Tentang

Halaman Tentang merupakan halaman yang memberikan informasi terkait Aplikasi.



Gambar 4.10. Halaman Tentang

Sumber: Hasil Penelitian

#### 4.2.5. Halaman Petunjuk

Halaman Petunjuk merupakan halaman yang memberikan informasi terkait penggunaan aplikasi deteksi daun kentang.



Gambar 4.11. Halaman Petunjuk

Sumber: Hasil Penelitian

#### 4.3. Evaluasi dan Validasi Hasil

Pada tahapan ini dijelaskan mengenai proses evaluasi dari pengujian, seberapa baik deteksi dengan aplikasi android yang telah dibuat. Pada Tabel 4.10 merupakan penjabaran dari citra yang diuji.

Tabel 4.10. Penjabaran Citra Teridentifikasi

Citra	Kelas	Model Prediksi	Kesimpulan
	Early Blight	91.1%	Sesuai

	Late Blight	97,3%	Sesuai
	Healthy	99.9%	Sesuai

Sumber: Hasil Penelitian

Dari pembahasan di atas dapat diketahui jika augmentasi yang dilakukan, lalu Ekstraksi fitur menggunakan metode *transfer learning* MobileNetV2 dan klasifikasi penyakit daun kentang menggunakan lapisan *global average pooling* dan *Dense* dapat dilakukan pada citra daun kentang dengan baik jika dibandingkan pada penelitian [2], dan dapat dirancang aplikasi android yang dapat mendeteksi daun kentang secara otomatis.

## **BAB V**

### **PENUTUP**

Kesimpulan yang dapat diambil berdasarkan hasil penelitian yang telah dilakukan disajikan dengan singkat serta pada bagian akhir terdapat saran yang dapat menjadi masukan dalam penelitian selanjutnya.

#### **5.1. Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Metode Augmentasi dapat digunakan untuk menangani imbalance data dan hal ini terbukti dengan meningkatnya kinerja model .
2. Metode klasifikasi yang digunakan melakukan pendekatan *transfer learning*. Tahap pelatihan model dilakukan dengan membekukan *base model* serta menambahkan beberapa *layer* tambahan. Dalam penelitian ini menggunakan 4 jenis model *transfer learning* berbeda untuk melakukan klasifikasi penyakit daun kentang. Keempat model tersebut adalah MobileNetV2, InceptionResNetV2, VGG16 dan Inception V3.
3. Evaluasi dilakukan dengan memprediksi kelas dari data *testing* menggunakan *confusion matrices* untuk mendapat perhitungan dari penilaian model yaitu akurasi. Model dengan performa terbaik dalam melakukan klasifikasi dicapai oleh model *transfer learning* MobileNetV2 dengan menggunakan dataset yang telah dilakukan augmentasi pada kelas sehat yakni dengan nilai *accuracy* 99,6%.
4. Implementasi antar muka dengan menerapkan metode-metode yang digunakan untuk deteksi penyakit daun kentang dapat diterapkan pada Aplikasi android. Deteksi penyakit daun kentang dapat dilakukan secara otomatis.

#### **5.2. Saran**

Berikut merupakan beberapa saran yang dapat digunakan untuk perbaikan pada penelitian selanjutnya

1. Pada proses augmentasi, untuk penelitian selanjutnya diharapkan dapat diterapkan pada semua kelas. Agar himpunan data menjadi banyak agar dapat menghasilkan performa model yang lebih baik.
2. Dilakukan *fine-tuning* dalam melakukan pelatihan untuk mencegah *overfitting*.
3. Pada aplikasi diharapkan dalam pengujiannya ditambahkan proses segmentasi. Sehingga meminimalisir latar belakang citra ikut terbaca.

## DAFTAR PUSTAKA

- [1] Kementrian Pertanian, “Kentang Sangkuriang Agrihorti Tahan Penyakit Busuk Daun,” 2019. [Online]. Available: <https://www.litbang.pertanian.go.id/info-teknologi/3836/>.
- [2] F. Islam, M. N. Hoq, and C. M. Rahman, “Application of Transfer Learning to Detect Potato Disease from Leaf Image,” *2019 IEEE Int. Conf. Robot. Autom. Artif. Internet-of-Things, RAAICON 2019*, pp. 127–130, 2019.
- [3] R. A. Sholihati, I. A. Sulistijono, A. Risnumawan, and E. Kusumawati, “Potato Leaf Disease Classification Using Deep Learning Approach,” *IES 2020 - Int. Electron. Symp. Role Auton. Intell. Syst. Hum. Life Comf.*, pp. 392–397, 2020.
- [4] H. P. Susetyo, “Penyakit Busuk Daun Kentang,” 2017. [Online]. Available: <http://hortikultura.pertanian.go.id/?p=2025>.
- [5] P. U. Rakhmawati, Y. M. Pranoto, and E. Setyati, “Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur Dan Fitur Warna Menggunakan Support Vector Machine,” *Semin. Nas. Teknol. dan Rekayasa 2018*, pp. 1–8, 2018.
- [6] R. D. Puspitasari, “Pertanian Berkelanjutan Berbasis Revolusi Industri 4.0,” *J. Layanan Masy. (Journal Public Serv.*, vol. 3, no. 1, p. 26, 2020.
- [7] H. F. Pardede *et al.*, “Plant diseases detection with low resolution data using nested skip connections,” *J. Big Data*, vol. 7, no. 1, 2020.
- [8] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Comput. Electron. Agric.*, vol. 145, no. September 2017, pp. 311–318, 2018.
- [9] D. Choe, E. Choi, and D. K. Kim, “The Real-Time Mobile Application for Classifying of Endangered Parrot Species Using the CNN Models Based on Transfer Learning,” *Mob. Inf. Syst.*, vol. 2020, 2020.
- [10] Z. Huang, Z. Pan, and B. Lei, “Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data,” *Remote Sens.*, vol. 9, no. 9, pp. 1–21, 2017.
- [11] B. Liu, C. Tan, S. Li, J. He, and H. Wang, “A Data Augmentation Method Based on Generative Adversarial Networks for Grape Leaf Disease Identification,” *IEEE Access*, vol. 8, pp. 102188–102198, 2020.
- [12] R. Hu, S. Zhang, P. Wang, G. Xu, D. Wang, and Y. Qian, “The identification of corn leaf diseases based on transfer learning and data augmentation,” *ACM Int. Conf. Proceeding Ser.*, pp. 58–65, 2020.
- [13] U. Barman, D. Sahu, G. G. Barman, and J. Das, “Comparative Assessment of Deep Learning to Detect the Leaf Diseases of Potato based on Data Augmentation,” *2020 Int. Conf. Comput. Perform. Eval. ComPE 2020*, pp. 682–687, 2020.
- [14] G. Geetharamani and A. P. J., “Identification of plant leaf diseases using a nine-layer deep convolutional neural network,” *Comput. Electr. Eng.*, vol. 76, pp. 323–338, 2019.
- [15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, vol. 3rd Editio, no. 3. Prentice hall, 2008.
- [16] F. Chollet, *Introduction to Keras*. 2017.
- [17] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [18] M. Hassaballah and A. I. Awad, *Deep Learning in Computer Vision: Principles*

- and Applications*. CRC Press, 2020.
- [19] Z. Song, “English speech recognition based on deep learning with multiple features,” *Computing*, vol. 102, no. 3, pp. 663–682, 2020.
  - [20] L. Lausen, “GluonCV and GluonNLP : Deep Learning in,” vol. 21, pp. 1–7, 2020.
  - [21] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.
  - [22] Y. Altuntaş, Z. Cömert, and A. F. Kocamaz, “Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach,” *Comput. Electron. Agric.*, vol. 163, no. 40, pp. 1–11, 2019.
  - [23] Y. Yu, H. Lin, J. Meng, X. Wei, H. Guo, and Z. Zhao, “Deep transfer learning for modality classification of medical images,” *Inf.*, vol. 8, no. 3, 2017.
  - [24] M. R. Karim and P. Pujari, *Practical Convolutional Neural Networks: Implement advanced deep learning models using Python*. 2018.
  - [25] Y. S. Tan, K. M. Lim, C. Tee, C. P. Lee, and C. Y. Low, “Convolutional neural network with spatial pyramid pooling for hand gesture recognition,” *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5339–5351, 2021.
  - [26] Y. Wang, Y. Li, Y. Song, and X. Rong, “The influence of the activation function in a convolution neural network model of facial expression recognition,” *Appl. Sci.*, vol. 10, no. 5, 2020.
  - [27] Y. Chen, C. Li, L. Gong, X. Wen, Y. Zhang, and W. Shi, “A deep neural network compression algorithm based on knowledge transfer for edge devices,” *Comput. Commun.*, vol. 163, no. August, pp. 186–194, 2020.
  - [28] M. A. H. Abas, N. Ismail, A. I. M. Yassin, and M. N. Taib, “VGG16 for plant image classification with transfer learning and data augmentation,” *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 90–94, 2018.
  - [29] M. Elgendy, *Deep Learning for Vision Systems*. 2019.
  - [30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018.
  - [31] Pusat Penelitian dan Pengembangan Hortikultura, “Pengenalan Penyakit yang Menyerang Pada Tanaman Kentang,” pp. 1–8.
  - [32] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, “Potato Leaf Diseases Detection Using Deep Learning,” *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. Iccics, pp. 461–466, 2020.
  - [33] T.-Y. Lee, I.-A. Lin, J.-Y. Yu, J. Yang, and Y.-C. Chang, “High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network,” *SN Comput. Sci.*, vol. 2, no. 4, 2021.
  - [34] J. Rozaqi, A. Sunyoto, and R. Arief, “Implementation of Transfer Learning in the Convolutional Neural Network Algorithm for Identification of Potato Leaf Disease,” vol. 1, no. 1, 2021.
  - [35] T. Y. Lee, J. Y. Yu, Y. C. Chang, and J. M. Yang, “Health Detection for Potato Leaf with Convolutional Neural Network,” *Indo - Taiwan 2nd Int. Conf. Comput. Anal. Networks, Indo-Taiwan ICAN 2020 - Proc.*, pp. 289–293, 2020.
  - [36] C. Narbuko and H. A. Achmadi, *Metodologi Penelitian*, no. January. Jakarta: PT Bumi Aksara, 2007.
  - [37] M. Lin, Q. Chen, and S. Yan, “Network in network,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.
  - [38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image

- recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.
- [39] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017.
- [40] Tensorflow, “TensorFlow lite converter,” 2019. .
- [41] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, “Potato Leaf Diseases Detection Using Deep Learning,” *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. January, pp. 461–466, 2020.

## DAFTAR RIWAYAT HIDUP

### A. Biodata Mahasiswa

Nama Lengkap : Tika Adilah M  
NIM : 14002364  
Tempat & Tanggal Lahir : Bandung, 30 Juni 1997  
Agama : Islam  
Alamat Lengkap : Kp. Bojong Monyet RT 02/ RW 06 Desa Bojong  
Salam Kecamatan Rancaekek, Kabupaten Bandung,  
40394.  
Nomor Telepon : 082218367840

### B. Riwayat Pendidikan Formal dan Non Formal

1. SD Negeri 1 Bojong Salam	Bandung	Lulus Tahun 2009
2. SMP Negeri 2 Rancaekek	Bandung	Lulus Tahun 2012
3. SMK Negeri 1 Rancaekek	Bandung	Lulus Tahun 2015
4. Universitas Bina Sarana Informatika	Bandung	Lulus Tahun 2018

### C. Riwayat Pengalaman Berorganisasi / Pekerjaan

1. Ketua OSIS SMP Negeri 2 Rancaekek	Tahun 2011 - 2012
2. Lurah Paskibra SMK Negeri 1 Rancaekek	Tahun 2014 - 2015
3. Universitas Bina Sarana Informatika	Tahun 2019 - Sekarang

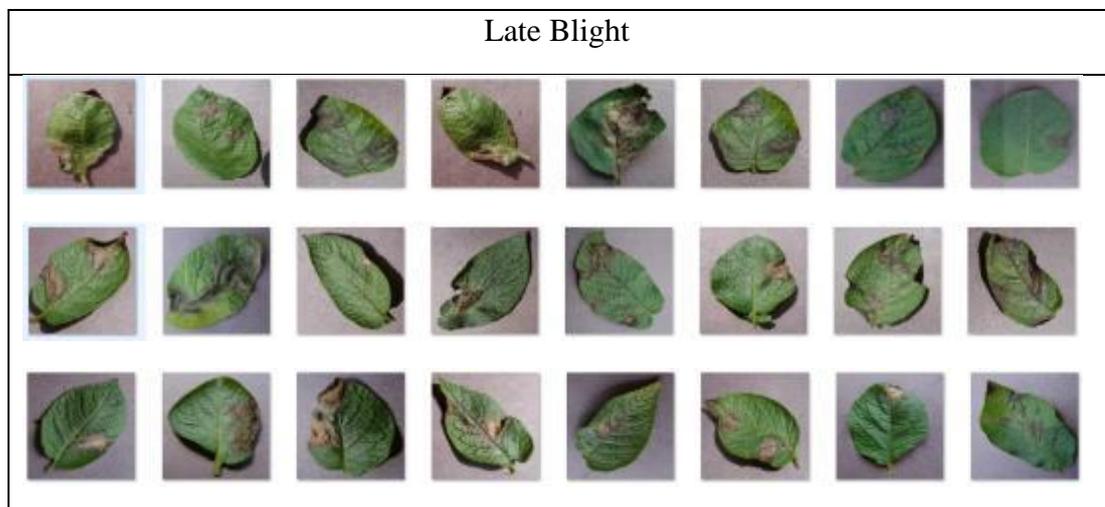
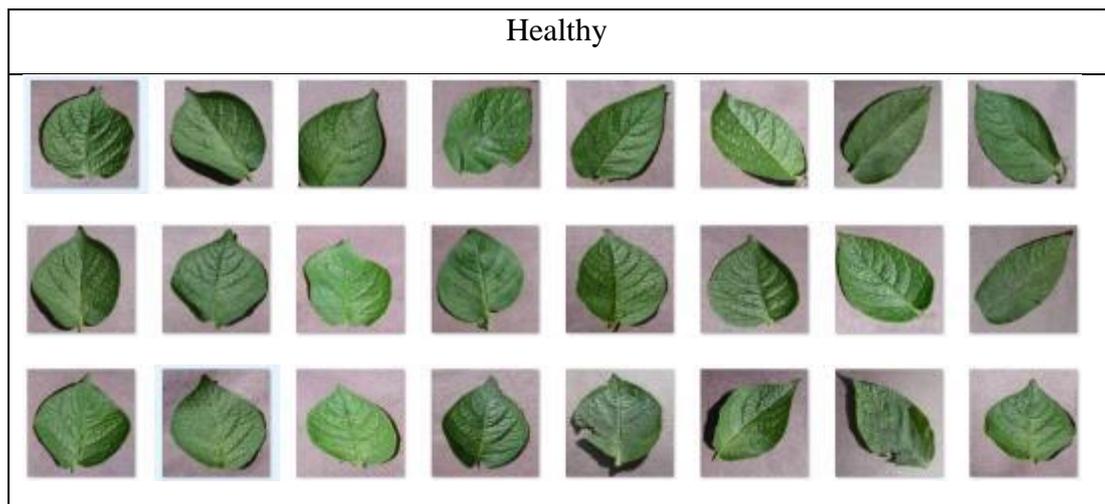
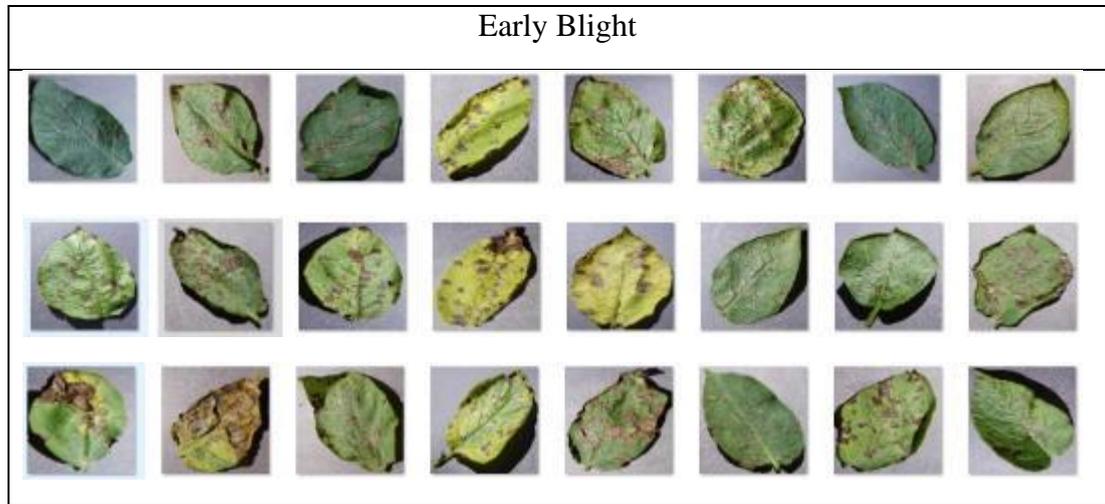


Jakarta, 22 Agustus 2021

Tika Adilah M

## LAMPIRAN-LAMPIRAN

### Lampiran 1. Contoh Dataset Citra Daun Kentang



## Lampiran 2. Citra Data Test

Early Blight	Healthy	Late Blight
		