



**IMPLEMENTASI METODE CONVOLUTIONAL NEURAL  
NETWORK UNTUK KLASIFIKASI PENYAKIT DAUN  
TANAMAN KENTANG MENGGUNAKAN ARSITEKTUR  
INCEPTION RESNET-V2**

**SKRIPSI**

Oleh:

**Pradana Rizki Susanto**

**182410103020**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER**

**2024**



**IMPLEMENTASI METODE CONVOLUTIONAL NEURAL  
NETWORK UNTUK KLASIFIKASI PENYAKIT DAUN  
TANAMAN KENTANG MENGGUNAKAN ARSITEKTUR  
INCEPTION RESNET-V2**

**SKRIPSI**

*diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana, pada  
program studi Informatika*

Oleh:

**Pradana Rizki Susanto**

**182410103020**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER**

**2024**

## **PERSEMBAHAN**

Alhamdulillah, puji syukur atas kehadiran Allah SWT yang selalu memberikan rahmat serta hidayah-Nya sehingga skripsi ini dapat diselesaikan. Dengan penuh kerendahan hati, skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa selalu memberikan rahmat dan petunjuk-Nya dalam setiap Langkah yang saya ambil dalam penelitian ini sehingga berhasil diselesaikan.
2. Orang tua saya Bapak Hendrik Susanto S.H dan Ibu Sudi Hidayati S.H yang telah memberi dukungan penuh semangat, motivasi, dan doa bagi kesuksesan penelitian ini dan juga kepada seluruh keluarga besar yang senantiasa selalu memberi doa terbaiknya.
3. Bapak Dwi Retno Istiyadi ST., M.Kom selaku Dosen Pembimbing Utama dan Muhammad Arief Hidayat S.Kom., M.Kom selaku Dosen Pembimbing Pendamping.
4. Seluruh teman saya yang telah memberikan dukungan dan semangat selama proses penelitian ini.

## **MOTTO**

*“Melihat mimpi kita terwujud itu memang menyenangkan, tapi bisa hidup untuk hari ini pun itu sudah cukup”*

**-Potgas D Ace One Piece**

## **PERNYATAAN ORISINALITAS**

Saya yang bertanda tangan dibawah ini:

Nama : Pradana Rizki Susanto

Nim : 182410103020

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul *“Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Tanaman Kentang Menggunakan Arsitektur Inception Resnet-V2”* adalah benar-benar hasil karya saya sendiri, kecuali jika ada pengutipan subtransi disebutkan sumbernya, belum pernah diajukan pada instansi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember,.....

Yang menyatakan

Pradana Rizki Susanto

NIM: 182410103020

## HALAMAN PENGESAHAN

Skripsi yang berjudul “Implementasi Metode *Convolutional Neural Network* Untuk Klasifikasi Penyakit Daun Tanaman Kentang Menggunakan Arsitektur *Inception Resnet-V2*” oleh Pradana Rizki Susanto NIM 182410103020 telah di uji dan disahkan pada :

Hari :

Tanggal :

Tempat : Fakultas Ilmu Komputer Universitas Jember

### Pembimbing

### Tanda Tangan

#### 1. Pembimbing Utama

Nama : Dr. Dwiretno Istiyadi S,ST., M.Kom  
NIP : 197803302003121003

(.....)

#### 2. Pembimbing Anggota

Nama : Muhammad Arief Hidayat,S.Kom.,  
M.Kom

(.....)

NIP : 198101232010121003

### Penguji

#### 1. Penguji Utama

Nama : Nelly Oktavia Adiwijaya S.Si., M.Kom  
NIP : 198410242009122008

(.....)

#### 2. Penguji Anggota

Nama : Maliatul Fitriyasari, ST., M.Sc  
NIP : 199503152023212038

(.....)

## **ABSTRAK**

Kentang (*Solanum Tuberosum L.*) menduduki peringkat keempat sebagai sumber pangan utama global setelah jagung, gandum, dan beras. Produksi kentang di Indonesia pada 2022 mencapai 1,42 juta ton, meningkat 4,21% dari tahun sebelumnya. Penyakit tanaman seperti busuk daun dan bercak kering daun dapat mengancam produksi kentang. Untuk mengatasi hal ini, penelitian menggabungkan kecerdasan buatan dan computer vision dengan menggunakan Convolutional Neural Network (CNN) Inception Resnet-V2 untuk mengklasifikasikan kondisi daun kentang. Penelitian sebelumnya mencapai tingkat akurasi hingga 95%. Penggunaan teknologi AI dalam pertanian menjanjikan solusi untuk mendeteksi dan mengatasi penyakit tanaman, potensial meningkatkan hasil pertanian dan mengurangi risiko kerugian.

**Kata kunci- CNN, Inception Resnet-V2, Daun Tanaman Kentang**

## **ABSTRACT**

*Potatoes (*Solanum Tuberosum L.*) are ranked fourth as the main global food source after corn, wheat and rice. Potato production in Indonesia in 2022 will reach 1.42 million tonnes, an increase of 4.21% from the previous year. Plant diseases such as late blight and dry leaf spots can threaten potato production. To overcome this, the research combines artificial intelligence and computer vision by using Convolutional Neural Network (CNN) Inception Resnet-V2 to classify the condition of potato leaves. Previous research achieved an accuracy rate of up to 95%. The use of AI technology in agriculture promises solutions for detecting and treating plant diseases, potentially increasing agricultural yields and reducing the risk of losses.*

**Keywords- CNN, Inception Resnet-V2, Daun Tanaman Kentang**

## RINGKASAN

Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Tanaman Kentang Menggunakan Arsitektur Inception Resnet-V2; Pradana Rizki Susanto;182410103020; 35 Halaman; Program Studi Informatika Fakultas Ilmu Komputer Universitas Jember.

Kentang (*Solanum Tuberosum L*) merupakan tanaman penting dalam sektor pertanian global, menduduki peringkat keempat setelah jagung, gandum, dan beras sebagai sumber pangan utama di dunia. Produksi kentang di Indonesia meningkat menjadi 1,42 juta ton pada 2022, mencatat pertumbuhan sebesar 4,21% dari tahun sebelumnya jawa timur menjadi produsen terbesar dengan 381.090 ton. Pencegahan dan pengendalian penyakit tanaman, seperti busuk daun dan bercak kering. Dalam upaya mengatasi penyakit tanaman terutama pada daun tanaman kentang, teknologi dan kecerdasan buatan(AI) telah menjadi solusi yang semakin penting. Penggunaan *Convolutional Neural Network* (CNN) khususnya dengan arsitektur Inception Resnet-V2 telah berhasil mendeteksi penyakit daun tanaman kentang dengan tingkat akurasi yang tinggi mencapai 95% dalam penelitian sebelumnya. Studi ini merencanakan pengembangan program komputer berbasis kecerdasan buatan untuk mengklasifikasikan kondisi daun kentang termasuk penyakit busuk daun, bercak kering dan daun sehat. Metode ini berfokus pendekatan CNN dengan struktur arsitektur Inception Resnet-V2. Harapannya, pendekatan ini dapat memberikan hasil akurasi maksimal dalam identifikasi penyakit daun tanaman kentang.

## PRAKATA

Puji syukur penulis panjatkan kehadiran Allah SWT atas segala limpahan Rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul *"Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Tanaman Kentang Menggunakan Arsitektur Inception Resnet-V2"* Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan Pendidikan strata satu (S1) di Program Studi Informatika Fakultas Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis menyampaikan terimakasih kepada :

1. Bapak Drs. Antonius Cahya Prihandoko, M.App.Sc, Ph.D Selaku Dekan Fakultas Ilmu Computer Universitas Jember
2. Ibu Nelly Oktavia A, S.Si, MT. Selaku dosen pembimbing akademik yang telah membimbing selama penulis menjadi mahasiswa.
3. Dr. Dwiretno Istiyadi Swasono, S.T., M.Kom. selaku dosen pembimbing utama dan Muhammad Arief Hidayat, S.Kom., M.Kom. selaku dosen pembimbing anggota dalam memberikan bimbingan dan arahan pada saat penyusunan skripsi.
4. Kedua orang tua saya Ayah dan Mama yang selalu mendoakan dan memberikan dukungan kepada saya baik secara moral maupun materi.
5. Teman-teman seperjuangan skripsi yang tidak dapat saya sebutkan satu persatu.

Penulis juga menyadari bahwa skripsi ini masih memiliki banyak kekurangan, oleh karena itu pembaca dapat memberikan kritik dan saran yang membangun untuk pengembangan skripsi ini menjadi lebih baik lagi. Semoga skripsi ini bermanfaat bagi semua.

Jember,.....

Pradana Rizki Susanto

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>PERSEMBAHAN.....</b>	<b>iii</b>
<b>MOTTO .....</b>	<b>iv</b>
<b>PERNYATAAN ORISINALITAS.....</b>	<b>v</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>RINGKASAN .....</b>	<b>viii</b>
<b>PRAKATA.....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xvi</b>
<b>BAB I. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	4
<b>BAB II. TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 Penelitian Terdahulu.....	5
2.2 Tanaman Kentang.....	7

2.3	Penyakit Tanaman Kentang.....	7
2.4	Computer Vision .....	9
2.5	Convolutional Neural Network (CNN) .....	9
2.6	Layer-layer pada CNN .....	10
2.7	Inception Resnet-V2.....	12
2.8	Transfer Learning .....	15
<b>BAB III. METODE PENELITIAN .....</b>		<b>17</b>
3.1	Tempat Penelitian.....	17
3.2	Jenis Penelitian .....	17
3.3	Tahapan Penelitian .....	17
3.3.1	Studi Literature.....	17
3.3.2	Pengumpulan dan Deskripsi Data .....	18
3.3.3	Preprocessing Data.....	<b>Error! Bookmark not defined.</b>
3.3.4	Pemodelan CNN .....	18
3.3.5	Evaluasi Hasil.....	19
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>21</b>
4.1	Pengambilan Dataset .....	21
4.2	Pemecahan Data .....	22
4.3	Augmentasi Citra.....	23
4.4	Ekstraksi Fitur .....	23
4.5	Implementasi kode model .....	26
4.6	Evaluasi Hasil Model .....	28
<b>BAB V. KESIMPULAN DAN SARAN .....</b>		<b>36</b>
5.1	Kesimpulan.....	36
5.2	Saran .....	36

<b>DAFTAR PUSTAKA .....</b>	<b>37</b>
-----------------------------	-----------

## DAFTAR TABEL

Tabel 2. 1 Perbandingan penelitian dengan metode CNN .....	6
Tabel 3. 1 Confusion Matrix .....	19
Tabel 4. 1 Penjelasan Citra yg digunakan dalam Menguji Sistem Inferensi .....	34

## DAFTAR GAMBAR

Gambar 2. 1 Busuk Daun ( <i>Late Blight</i> ) Sumber <a href="https://maulzxxx.wordpress.com/8">https://maulzxxx.wordpress.com/8</a>	
Gambar 2. 2 Bercak kering (Early Blight) Sumber : <a href="https://plantix.net/">https://plantix.net/</a> .....	8
Gambar 2. 3 Convolutional Neural Network (Sumber : <a href="https://www.trivusi.web.id">https://www.trivusi.web.id</a> .....	9
Gambar 2. 4 Operasi Konvolusi (Sumber : <a href="https://media.neliti.com">https://media.neliti.com</a> ).....	10
Gambar 2. 5 Max Pooling Sumber <a href="https://medium.com">https://medium.com</a> .....	11
Gambar 2. 6 Fully Connected Layer (Sumber : <a href="https://indiantechwarrior.com">https://indiantechwarrior.com</a> ) ..	11
Gambar 2. 7 Skema Inception Resnet-V2 (Sumber : <i>Ejournal Universitas majalengka</i> ).....	13
Gambar 2. 8 Stem Blok (Sumber : <i>Ejournal Universitas majalengka</i> ) .....	13
Gambar 2. 9 Inception Resnet A (Sumber : <i>Ejournal Universitas majalengka</i> )	13
Gambar 2. 10 Inception Resnet B (Sumber : <i>Ejournal Universitas majalengka</i> )	14
Gambar 2. 11 Inception Resnet C (Sumber : <i>Ejournal Universitas majalengka</i> )	14
Gambar 2. 12 Reduction A (Sumber : <i>Ejournal Universitas majalengka</i> ).....	15
Gambar 2. 13 Reduction B (Sumber : <i>Ejournal Universitas majalengka</i> ).....	15
Gambar 3. 1 Tahapan Penelitian .....	17
Gambar 3. 2 Pemodelan CNN.....	19
Gambar 4. 1 Label Dataset.....	21
Gambar 4. 2 Implementasi Pseudocode dalam proses pemecahan dataset.....	22
Gambar 4. 3 Implementasi Pseudocode dalam proses augmentasi citra .....	23
Gambar 4. 4 Implementasi Pseudocode pada Inception Resnet-V2 Transfer Learning Model .....	25
Gambar 4. 5 Implementasi Pseudocode untuk menampilkan Confusion Matrix.....	27
Gambar 4. 6 Implementasi Pseudocode akurasi .....	27
Gambar 4. 7 Implementasi Pseudocode dalam menampilkan Precision, Recall, dan F1-Score.....	27
Gambar 4. 8 Confusion Matrix Arsitektur Inception ResnetV2 .....	29
Gambar 4. 9 Confusion Matrix Arsitektur Inception ResnetV2 .....	30
Gambar 4. 10 Confusion Matrix Arsitektur Inception ResnetV2 .....	31

Gambar 4. 11 Classification Report Arsitektur Inception ResnetV2.....	32
Gambar 4. 12 Classification Report Arsitektur Inception ResnetV2.....	32
Gambar 4. 13 Classification Report Arsitektur Inception ResnetV2.....	33

## DAFTAR LAMPIRAN

Lampiran 1 Arsitektur Inception ResnetV2 Transfer Learning .....	39
--	----

## **BAB I. PENDAHULUAN**

Bab pendahuluan menarasikan tentang langkah awal sebuah penelitian. Bab pendahuluan untuk skripsi terdiri dari beberapa sub-bab yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan manfaat penelitian.

### **1.1 Latar Belakang**

Kentang (*Solamun Tuberosum L.*) merupakan tanaman yang memiliki daya tarik tinggi di sektor pertanian global. Setelah tanaman jagung, gandum, dan beras, kentang menempati peringkat keempat sebagai salah satu sumber pangan utama yang paling signifikan di dunia. (Putri et al. 2021).

Badan Pusat Statistik (BPS) mencatat, produksi kentang di Indonesia mencapai 1,42 juta ton pada 2022. Jumlah tersebut meningkat 4,21% dibandingkan pada tahun sebelumnya yang sebanyak 1,36 juta ton. Melihat trennya, produksi kentang di Indonesia berfluktuasi cenderung meningkat. Produksi pada tahun lalu merupakan rekor tertinggi dalam satu dekade terakhir. Terdapat 21 provinsi di Indonesia yang merupakan produsen kentang pada 2022. Produksi terbanyak berasal dari Jawa Timur yang mencapai 381.090 ton (DataIndonesia.id).

Untuk menjamin kualitas kentang sebagai salah satu sumber pangan yang penting pencegahan dan pengendalian penyakit tanaman efektif untuk dilakukan. Terjadinya penyakit tanaman akan berdampak buruk pada produksi pertanian. Apabila penyakit tanaman kentang tidak dideteksi tepat waktu maka akan terjadi peningkatan kerawanan pada tanaman kentang. Secara umum, daun tanaman merupakan sumber informasi pertama untuk mengidentifikasi penyakit tanaman dan sebagian besar gejala penyakit dapat muncul pada daun. Salah satu jenis penyakit yang sering ditemukan pada tanaman kentang adalah busuk daun (*late blight*), dan juga terdapat penyakit bercak kering daun (*early blight*). Aspek-aspek seperti kondisi lingkungan yang sejuk dan lembab merupakan salah satu pemicu munculnya penyakit daun yang membusuk. (Rozaqi, Sunyoto, dan Arief, n.d.2021).

Dalam mengatasi isu terkait penyakit daun kentang ini, telah banyak usaha dilakukan bukan hanya di sektor pertanian, melainkan juga di bidang teknologi, contohnya menggunakan aspek informatika untuk mengenali gangguan pada tanaman

kentang. Ancaman hama dan penyakit ini membuat petani merasa cemas terhadap potensi kerugian akibat kegagalan panen (Cucu Oktaviana et.al,2017).

Kemajuan teknologi dalam ranah Kecerdasan Buatan (*Artificial Intelligence* atau AI) memiliki dampak signifikan dalam sektor kesehatan, peternakan, pertanian, dan berbagai bidang lainnya. Salah satu penelitian yang memiliki pengaruh besar adalah *deep learning*, yang telah mengalami kemajuan yang sangat cepat dalam berbagai hal seperti penggambaran data, pengestrakan fitur, dan interaksi fitur. (Chen, 2021)

Metode *Convolutional Neural Network (CNN)* merupakan salah satu mode *deep learning* yang sering digunakan dalam penelitian. CNN sendiri memiliki cara kerja dengan menggunakan dataset gambar atau video sebagai masukannya (input). Dalam dunia *deep learning*, *CNN* sering digunakan untuk penelitian khususnya untuk mengklasifikasikan suatu studi kasus dengan objek gambar atau citra gambar. *CNN* menggunakan *Multi Layer Perceptron (MLP)* untuk memproses dan mengklasifikasikan citra gambar (Prakosa et al. 2023).

Pada penelitian sebelumnya mendeteksi penyakit pada daun tanaman kentang menggunakan pengolahan citra dengan Metode *Convolutional Neural Network* Metode ini telah digunakan pada penelitian terdahulu untuk mendeteksi penyakit pada daun tanaman kentang menggunakan metode *Convolutional Neural Network*. Hasil dari penelitian ini diperoleh 95% nilai akurasi dan 94% nilai akurasi validasi (Abdul Jalil Rozaqi et al. 2021). Penelitian lain menggunakan arsitektur CNN Inception Resnet-V2 untuk perbandingan antara berbagai model, yang dimana pada penelitian ini mendapatkan tingkat akurasi yang tinggi sebesar 92% (Fitriana Masrurroh et al. 2023).

Dalam studi ini, direncanakan membuat sebuah program komputer yang bertujuan untuk mengategorikan kondisi penyakit pada daun tanaman kentang, termasuk di antaranya penyakit busuk daun (*late blight*), penyakit bercak kering (*early blight*), dan keadaan daun yang sehat (*Healthy*). Untuk mencapai tujuan tersebut, pendekatan berbasis kecerdasan buatan akan diterapkan, dengan fokus utama pada teknik berbasis (*artificial intelligence*), guna menyederhanakan proses klasifikasi gangguan pada daun tanaman kentang. Metode yang akan dipakai melibatkan konsep *computer vision*, dan akan menggunakan pendekatan *Convolutional Neural Network*

(CNN) dengan struktur arsitektur inception resnet-V2. Harapannya, metode ini akan memberikan hasil akurasi yang maksimal.

### **1.2 Rumusan Masalah**

Berkaitan dengan uraian diatas, maka permasalahan yang dapat dirumuskan dalam penelitian yaitu :

1. Bagaimana menerapkan arsitektur Inception Resnet-V2 pada klasifikasi penyakit daun tanaman kentang menggunakan metode *Convolutional Neural Network* (CNN)?
2. Bagaimana performa metode *Convolutional Neural Network* (CNN) dengan arsitektur Inception Resnet-V2 yang digunakan pada klasifikasi penyakit daun tanaman kentang?

### **1.3 Tujuan Penelitian**

Berdasarkan rumusan diatas, maka tujuan yang ingin dicapai dalam penelitian sebagai berikut :

1. Untuk mengimplementasikan Inception Resnet-V2 pada klasifikasi penyakit daun tanaman kentang menggunakan metode *Convolutional Neural Network* (CNN).
2. Untuk mengetahui model yang baik untuk mengukur performa model *Convolutional Neural Network* (CNN) dalam mengklasifikasikan penyakit daun tanaman kentang.

### **1.4 Manfaat Penelitian**

Hasil dari penelitian ini dapat memberikan manfaat yaitu:

1. Bagi Penulis  
Memberi tambahan ilmu pengetahuan tentang metode *Convolutional Neural Network* (CNN) dan mampu menerapkan untuk klasifikasi gambar.
3. Bagi Lembaga Pendidikan  
Memberikan informasi kepada pembaca tentang klasifikasi dengan metode *Convolutional Neural Network* (CNN) dan juga sebagai bahan referensi bagi yang membutuhkan.

### **1.5 Batasan Masalah**

Batasan masalah dalam penelitian ini untuk menghindari perluasan topik sehingga penelitian ini menjadi lebih terarah dan pembahasan untuk mencapai tujuan penelitian menjadi lebih mudah. Berikut ini adalah beberapa faktor masalah dalam penelitian ini :

1. Dataset penyakit daun tanaman kentang yang dimanfaatkan dalam riset ini merupakan kumpulan data yang diperoleh dari platform Kaggle.
2. Penelitian ini lebih berfokus pada penyakit daun tanaman kentang dengan metode *CNN- Inception Resnet-V2*
3. Bahasa Pemrograman yang dipakai dalam penelitian ini yakni python.
4. Arsitektur model CNN yang digunakan adalah *Inception Resnet-V2*

## BAB II. TINJAUAN PUSTAKA

Pada Bab ini menjelaskan tentang teori-teori yang digunakan untuk penelitian, dan teori-teori ini diambil dari studi literatur:

### 2.1 Penelitian Terdahulu

Terdapat banyak penelitian sebelumnya yang berkaitan dengan penelitian yang akan dilaksanakan. Penelitian pertama oleh Evelyn Gabriela Winarto *et al* pada tahun 2021 berjudul “Penerapan arsitektur Inception Resnet-V2 pertama untuk klasifikasi mutu biji kakao”. Maksudnya adalah untuk mengategorikan mutu biji kakao, menggunakan struktur jaringan saraf tiruan konvolusi (CNN), model Inception Resnet-V2, guna mengklasifikasikan biji kakao. Penelitian ini berhasil mengenal pasti set data yang dipakai yaitu kumpulan citra biji kakao yang terdiri dari 614 berkas dengan enam kategori, yakni *whole beans*, *bean fraction cocoa*, *fermented cocoa*, *broken beans cocoa*, *unfermented cocoa*, dan *moldy cocoa*. Dari hasil percobaan dengan menjalankan 100 iterasi pelatihan, diperoleh tingkat ketepatan sebesar 89% dan nilai Area di Bawah Kurva (ROC) sebesar 97%.(Winarto, Rahmayati, dan Lawi 2021).

Kedua, studi yang dilakukan oleh Abdul Jalil Rozaqi dkk. pada tahun 2021 berjudul "Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra Dengan Metode *Convolution Neural Network*" mencatat bahwa hasil penelitian ini memperlihatkan performa yang mengesankan melalui penerapan model dan uji coba terhadap sampel daun kentang. Penelitian ini membagi dataset menjadi dua bagian, yakni 80% untuk pelatihan dan 20% untuk pengujian, dengan mengubah dimensi gambar menjadi 150x150 piksel. Pada *epoch* ke-10 dengan *batch\_size* 20, menggunakan total 922 gambar dalam tahap pelatihan dan 230 gambar pada tahap pengujian, berhasil mencapai akurasi sebesar 95%. Dalam hal validasi, model ini mampu mencapai tingkat akurasi 94% (Abdul Jalil Rozaqi *et al* 2021.).

Ketiga, penelitian yang dilakukan oleh Fitriana Masruroh *et.al* pada tahun 2023 berjudul “Perbandingan kinerja inception resnetV2,Xception,InceptionV3,

dan Resnet50 pada gambar bentuk wajah”. Penelitian ini bertujuan menerapkan metode *deep learning* untuk mengklasifikasikan berbagai bentuk wajah. Pendekatan ini membandingkan struktur arsitektur Inception ResnetV2, Xception, InceptionV3, dan Resnet50. Peningkatan dalam akurasi klasifikasi dapat dicapai melalui pemanfaatan berbagai teknik pengoptimalan dalam domain *deep learning*, serta dengan mengadopsi fungsi aktivasi Mish. Dari rangkaian percobaan yang dilaksanakan oleh peneliti, diketahui bahwa arsitektur Xception yang mengadopsi Mish dan Nadam menghasilkan tingkat akurasi sebesar 91,33%. Sementara itu, model Resnet50 yang diterapkan dengan Nadam dan Mish mencapai tingkat akurasi 89,80%, dan arsitektur InceptionV3 yang diperkuat oleh Nadam dan Mish mencapai akurasi 89,00%. Namun, pencapaian tertinggi terdapat pada arsitektur Inception ResNetV2 yang memadukan Nadam dan Mish, dengan tingkat akurasi mencapai 92,00%, melampaui ketiga arsitektur lainnya.(Fitriana Musaroh *et al.* 2023).

Tabel 2. 1 Perbandingan penelitian dengan metode CNN

No.	Penulis	Judul	Persamaan	Perbedaan
1	Evelyn Gabriela Winarto, dkk (2021)	Implementasi.Arsitektur Inception Resnet-V2 Untuk Klasifikasi Kualitas Biji Kakao	Sama-sama menggunakan metode CNN dengan Arsitektur Inception Resnet-V2	Objek yang digunakan berbeda
2	Abdul Jalil Rozaqi , dkk (2021)	Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra Dengan Metode <i>Convolutional Neural Network</i>	Sama-sama menggunakan metode CNN dan Objek yang sama	Tidak menggunakan Arsitektur

3.	Fitriana Musaroh, dkk (2023)	Perbandingan Inception V2,Xception, Resnet-V3 Dan Pada Gambar Wajah	Kinerja Resnet- Inception Resnet50 Bentuk	Sama-sama menggunakan Arsitektur Inception Resnet-V2	Objek yang berbeda
----	---------------------------------------	--	---	--	-----------------------

## 2.2 Tanaman Kentang

Kentang, yang secara ilmiah dikenal sebagai *Solanum Tuberosum L* merupakan jenis umbi-umbian yang memiliki nilai ekonomis tinggi. Hal ini menguntungkan petani karena harganya cenderung tetap stabil, dan umbi kentang juga memiliki kemampuan penyimpanan yang lebih lama dibandingkan dengan sebagian besar sayuran lainnya, seperti yang dinyatakan oleh (Ridwan, 2010). Umbi kentang dapat berperan sebagai sumber pangan karena mengandung jumlah karbohidrat, mineral, kalori, dan vitamin yang cukup tinggi. Kandungan ini memungkinkan penggantian sumber karbohidrat dari sumber-sumber seperti beras, gandum, atau jagung, yang telah menjadi populer dalam memenuhi kebutuhan pangan global. (Cahyono, 1996).

## 2.3 Penyakit Tanaman Kentang

Tanaman Kentang sangat rentan terhadap penyakit, apabila infeksi pada tanaman tersebut dibiarkan, akan mengakibatkan penurunan produksi makanan yang signifikan. Karena itu, deteksi penyakit pada tanaman perlu dilaksanakan pada waktu yang tepat agar penyebarannya bisa dikontrol dan dicegah dengan efektif. Salah satu penyakit yang secara umum menjangkiti tanaman kentang adalah busuk daun (*late blight*), juga terdapat penyakit bercak kering pada daun (*early blight*).

### a. Busuk Daun (*Late Blight*)

Busuk Daun (*Late Blight*) akan mengemuka dalam rentang waktu pertumbuhan antara minggu kelima dan minggu keenam. Tanda-tanda permulaan dari gangguan ini adalah adanya bercak lembab di pinggiran

daun atau bahkan di tengah daun. Selanjutnya, bercak ini akan merambat dan warna daun akan mengalami perubahan menjadi corak coklat atau abu-abu.



Gambar 2. 1 Busuk Daun (*Late Blight*) Sumber <https://maulzxxx.wordpress.com/>

Tanda-tanda pertama penyakit busuk daun kentang ialah timbulnya noda lembab di pinggir atau mungkin juga di tengah daun. Noda ini kemudian meluas menghasilkan area berwarna coklat / abu-abu dengan inti yang agak gelap dan sedikit lembab. Daerah yang terkena noda aktif ditutupi oleh massa *sporangium* menyerupai tepung putih dengan latar belakang hijau keabu-abuan. Biasanya, gejala pada daun tanaman kentang muncul setelah tanaman berusia lebih dari satu bulan.

b. Bercak Kering (*Early Blight*)

Penyakit (*Early Blight*) dicirikan oleh adanya bercak kering berbentuk lingkaran berwarna coklat pada bagian bawah daun.



Gambar 2. 2 Bercak kering (*Early Blight*)  
Sumber : <https://plantix.net/>

Gejala awal bercak kering pada daun bagian bawah, berwarna coklat berupa tanda khas lingkaran berpusat (seperti cincin) pada bercak tersebut, sporulasi tidak nampak seperti embun putih. Penyakit bercak

kering (*alternaria solani*) adalah penyakit pada kentang yang disebabkan oleh jamur *Alternaria solani*.

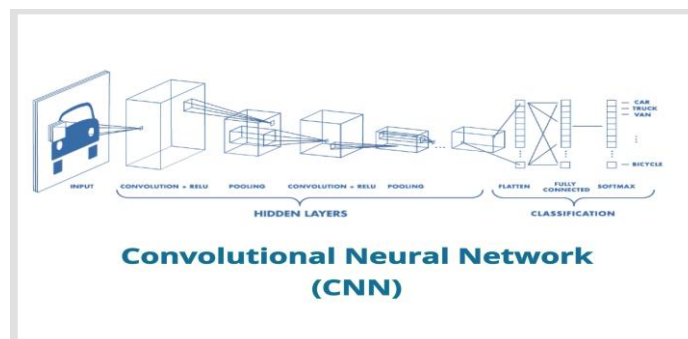
## 2.4 Computer Vision

Pengolahan citra merupakan manipulasi sebuah gambar dengan menggunakan komputer, dengan tujuan menghasilkan gambar yang berbeda. Kaitannya dengan *computer vision* adalah bahwa pengolahan citra berperan sebagai tahap awal (*preprocessing*) dalam *computer vision*. Teknologi *computer vision* bertujuan untuk mereplikasi metode pengamatan visual manusia.

*Computer Vision* adalah suatu proses otomatis yang menggabungkan beragam proses besar untuk interpretasi visual, termasuk perolehan gambar, pemrosesan gambar, klasifikasi, pengenalan, serta pengambilan keputusan. (Pradhitya 2015).

## 2.5 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) adalah jenis saraf tiruan (*artificial neural network*) yang dirancang khusus untuk memproses data berupa gambar dan data dengan struktur grid seperti citra atau video. CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi (*convolutional layer*), lapisan penggabungan (*pooling layer*) dan lapisan sepenuhnya terhubung (*fully connected layer*). Lapisan konvolusi adalah ini dari CNN yang menggunakan operasi konvolusi untuk menerapkan filter pada data input. Fiter ini berfungsi untuk mengekstraksi fitur-fitur penting dari data gambar.



Gambar 2.3 Convolutional Neural Network (Sumber : <https://www.trivusi.web.id>)

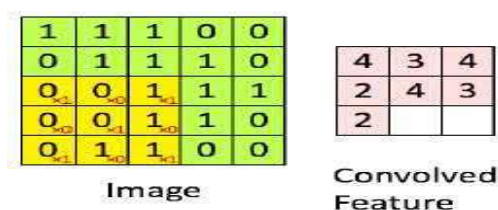
Struktur CNN memiliki komponen berupa input, tahap ekstraksi ciri, langkah klasifikasi, dan keluaran. Langkah ekstraksi pada CNN terdiri dari serangkaian lapisan tersembunyi, yakni lapisan konvolusi, fungsi aktivasi (ReLU), dan penggabungan informasi (*pooling*). Operasi CNN berlangsung dalam bentuk hirarki, sehingga hasil dari lapisan konvolusi awal diteruskan sebagai input bagi lapisan konvolusi berikutnya. Dalam tahap klasifikasi, terdapat tahap *fully-connected* dan penerapan fungsi aktivasi (*softmax*) yang menghasilkan luaran berupa hasil klasifikasi (Katole et al., 2015)

## 2.6 Layer-layer pada CNN

Ini telah mendorong pengembangan pembelajaran mesin, di mana model diberi lapisan data yang diubah sebelum diproses dengan metode klasifikasi. Hal ini mendorong pengembangan model jaringan saraf dengan lebih dari tiga lapisan. Namun, karena peranan layer awal sebagai metode ekstraksi fitur, tidak ada aturan umum untuk jumlah layer dalam sebuah CNN, akan berbeda-beda tergantung dari dataset yang digunakan. Oleh karena itu, jumlah lapisan dalam jaringan dan jumlah neuron per lapisan dianggap sebagai hyperparameter dan dioptimalkan menggunakan metode pencarian. CNN memiliki beberapa lapisan :

### 1. Convolution Layer

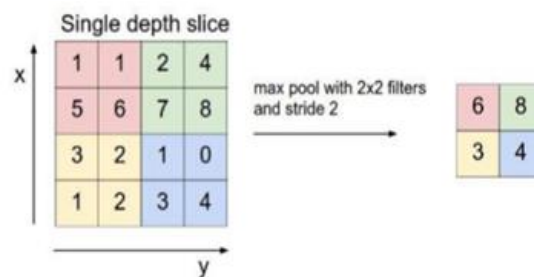
*Convolution layer* melakukan operasi konvolusi pada *output* dari layer sebelumnya. Maksud melakukan konvolusi terhadap data gambar ialah untuk menarik ciri dari gambar masukan. Proses konvolusi mampu menciptakan perubahan linier pada data masukan sesuai informasi istimewa yang terkandung dalam data tersebut. Parameter di lapisan ini menentukan kernel konvolusi yang diterapkan, sehingga kernel konvolusi dapat diadaptasi melalui pelatihan dengan input pada Jaringan Saraf Konvolusi (CNN).



Gambar 2. 4 Operasi Konvolusi (Sumber : <https://media.neliti.com>)

## 2. Pooling layer

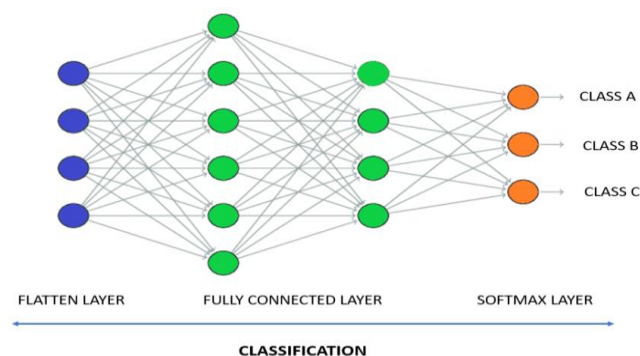
Lapisan Pooling adalah komponen yang digunakan untuk mengurangi dimensi gambar (*down sampling*) dengan tujuan mempermudah operasi konvolusi pada lapisan berikutnya. Reduksi ini dapat diterapkan melalui beberapa metode, termasuk *max pooling* (menggambil nilai maksimum dari subgambar), *average pooling* (menggambil nilai rata-rata dari subgambar). Ukuran subgambar yang digunakan adalah 2x2.



Gambar 2. 5 Max Pooling Sumber <https://medium.com>

## 3. Fully Connected Layer

Pada bagian ini, seluruh neuron terhubung secara penuh seperti dalam *multi-layer perceptron* (MLP), dengan bobot dan bias yang digunakan untuk menggambarkan data, mirip dengan cara yang diterapkan oleh MLP.



Gambar 2. 6 Fully Connected Layer (Sumber : <https://indiantechwarrior.com>)

#### 4. Relu

Fungsi pengelompokan ini bertujuan untuk mengubah nilai-nilai negatif menjadi nol, sehingga semua nilai dalam peta fitur memiliki nilai positif. Aktivasi Relu beroperasi dengan mengambil input dari peta fitur dan membatasi nilai-nilai negatif dengan mengganti bilangan-bilangan tersebut menjadi nol.

$$f(x) = \max(0, x)$$

Keterangan :

$X = \text{input pixel}$

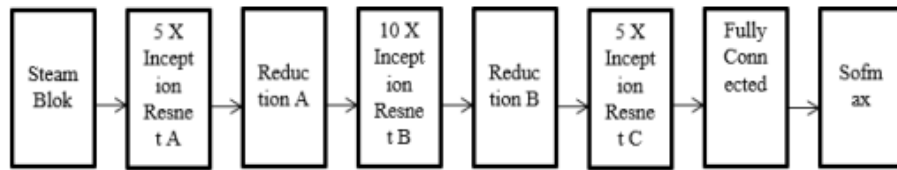
#### 5. Softmax

Fungsi aktivasi *softmax* digunakan khusus dalam *output layer* dari model klasifikasi multi kelas. Fungsi ini mengubah nilai *input* menjadi probabilitas. Ketika mendapatkan vektor *output* berada dalam rentang [0,1] dan total probabilitas dari semua elemen adalah satu (Patterson & Gibson, 2017)

### 2.7 Inception Resnet-V2

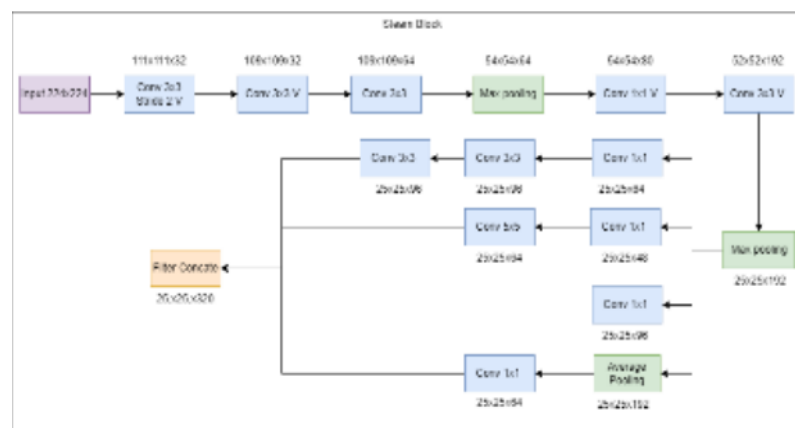
Inception resnet-V2 merupakan modifikasi dari inception V3 dan memiliki tingkat kompleksitas yang lebih dalam daripada inception V3 sebelumnya. Pada segmen konvolusi di Inception resnet-V2, terdapat struktur jaringan yang serupa di mana blok-blok residual yang berulang telah dikompres. Keberhasilan blok residual, yang awalnya diperkenalkan dalam arsitektur resnet dan menjadi pemenang dalam LSVRC 2015 (Kompetisi Pengenalan Visual Skala Besar *ImageNet*), telah mengatasi tantangan Vanishing Gradient yang muncul akibat banyaknya lapisan konvolusi.

Oleh sebab itu, diperlukanlah koneksi residual agar dapat melatih arsitektur yang sangat dalam. Inception Resnet-V2 merupakan suatu struktur CNN yang dibangun di atas keluarga arsitektur inception, namun mengintegrasikan koneksi residual (menggantikan tahap perangkat filter dari arsitektur inception). Inception Resnet-V2 memperoleh semua keuntungan dari pendekatan residual sambil tetap mempertahankan efisiensi perhitungannya. Ilustrasi Inception Resnet-V2 tersaji pada Gambar 2.7.



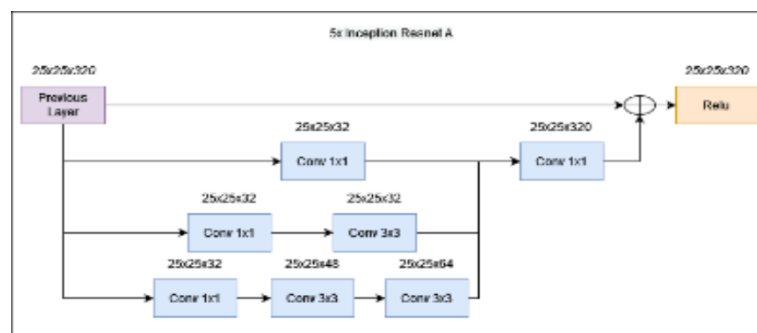
Gambar 2. 7 Skema Inception Resnet-V2 (Sumber : *Ejournal Universitas majalengka*)

Pada blok stem ini untuk memberi setup awal yang dilaksanakan sebelum menuju ke blok Inception Blok selanjutnya. Stem blok bisa dilihat pada Gambar 2.8.

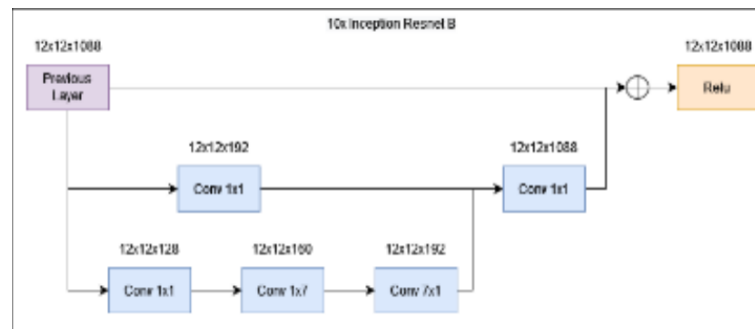


Gambar 2. 8 Stem Blok (Sumber : *Ejournal Universitas majalengka*)

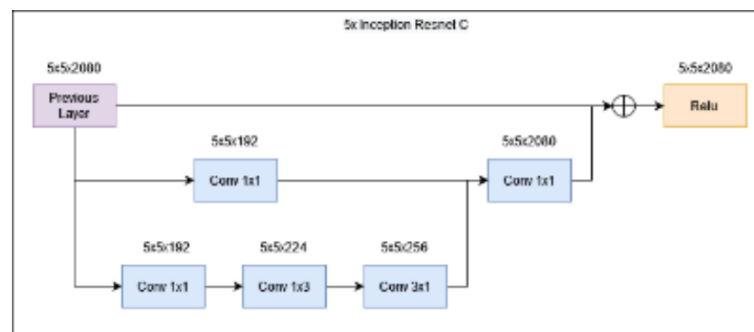
Pada blok inception resnet, diperkenalkanlah koneksi residual yang dibangun di atas struktur keluarga arsitektur inception. Namun, perbedaannya terletak pada penggabungan koneksi residual yang menggantikan langkah-langkah rangkaian filter dalam arsitektur inception. Blok inception resnet terdiri dari modul A, B, dan C, dan tampilan dari masing-masing modul ini dapat ditemukan dalam gambar 2.9, 2.10, dan 2.11



Gambar 2. 9 Inception Resnet A (Sumber : *Ejournal Universitas majalengka*)

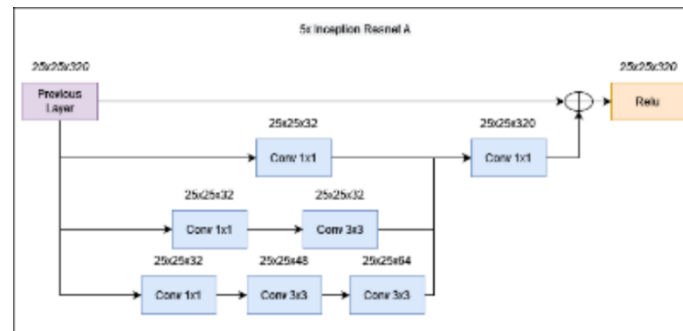


Gambar 2. 10 Inception Resnet B (Sumber : *Ejournal Universitas majalengka*)

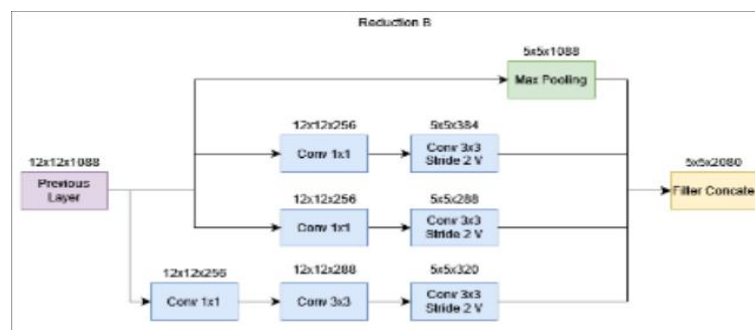


Gambar 2. 11 Inception Resnet C (Sumber : *Ejournal Universitas majalengka*)

Langkah berikutnya dalam proses ekstraksi fitur adalah memasuki blok reduksi, di mana dalam blok ini akan terjadi penyusutan ukuran fitur melalui penerapan operasi *max pooling* pada setiap blok reduksi, yang diikuti oleh penggabungan fitur atau penggabungan *Concatenate* yang berasal dari Reduksi A dan Reduksi B. Detail dari Blok Reduksi A dan Reduksi B dapat ditemukan pada gambar 2.12 dan 2.13.



Gambar 2. 12 Reduction A (Sumber : Ejournal Universitas majalengka)



Gambar 2. 13 Reduction B (Sumber : Ejournal Universitas majalengka)

## 2.8 Transfer Learning

*Transfer Learning* ialah pendekatan yang memanfaatkan jaringan yang telah diberi pelatihan sebelumnya dan memanfaatkannya sebagai pijakan awal dalam menguasai tugas-tugas baru. Melakukan penyempurnaan lanjutan pada jaringan melalui transfer learning jauh lebih cepat dan lebih sederhana daripada mengawali pelatihan jaringan dari awal dengan bobot-bobot yang diacak.

*Transfer learning* memungkinkan pemindahan pengetahuan dari fitur yang telah dipelajari ke dalam tugas baru dengan memanfaatkan dataset yang lebih terbatas dalam proses pelatihan, sehingga mempercepat prosesnya. Melakukan penyesuaian ulang pada jaringan biasanya memerlukan waktu yang lebih lama dan melibatkan usaha yang lebih besar daripada sekadar mengekstraksi fitur sederhana. Namun, dengan menerapkan *transfer learning*, jaringan yang telah dilatih sebelumnya dan telah memperoleh pemahaman tentang fitur-fitur yang beragam dapat memberikan kinerja yang lebih unggul daripada pendekatan ekstraksi fitur, meskipun hanya menggunakan dataset yang lebih terbatas. Hal

ini disebabkan jaringan yang baru telah menyerap pemahaman tentang fitur-fitur baru yang telah dipelajarinya sebelumnya. (Wonohadidjojo 2021).

## BAB III. METODE PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan dalam penelitian. Metode penelitian berfungsi sebagai gambaran tahapan penelitian untuk menganalisa data serta mengembangkan dan menerapkan sistem.

### 3.1 Tempat Penelitian

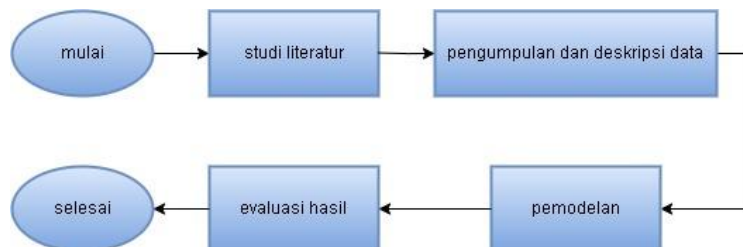
Penelitian ini akan dilaksanakan di Fakultas Ilmu Komputer Universitas Jember.

### 3.2 Jenis Penelitian

Tipe penelitian atau jenis penelitian yang digunakan adalah penelitian Terapan, dimana peneliti menerapkan metode CNN dengan arsitektur Inception Resnet-V2.

### 3.3 Tahapan Penelitian

Riset ini dijalankan melalui lima fase berbeda, dimulai dengan studi literatur, pengumpulan dan penjelasan data, pra-pemrosesan data, pemodelan, dan penilaian hasil. Tahapan-tahapan dari penelitian ini dapat ditemukan pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

#### 3.3.1 Studi Literature

Pada fase ini, peneliti menghimpun beragam referensi yang diperoleh dari sumber-sumber seperti buku dan jurnal yang relevan dengan topik penelitian ini. Landasan teori yang diterapkan bersumber dari karya-karya sebelumnya yang mendukung riset ini. *Fundamentally*, informasi yang terkumpul dalam tahap ini mencakup aspek gambar daun, pengolahan citra komputer, pendekatan *Convolutional Neural Network* (CNN), serta struktur arsitektur CNN Inception Resnet-V2 yang diimplementasikan menggunakan bahasa pemrograman *python*.

### 3.3.2 Pengumpulan dan Deskripsi Data

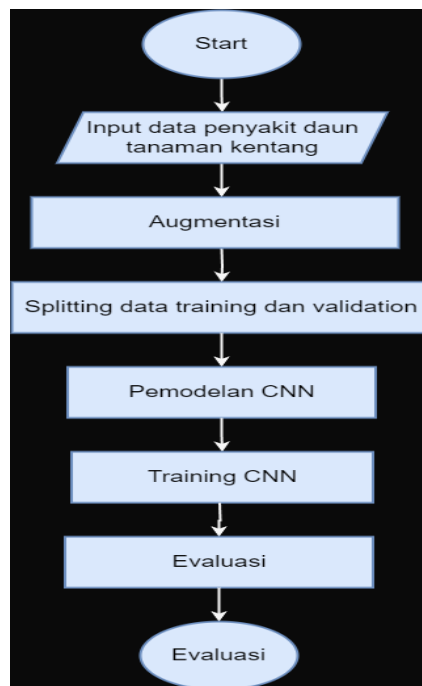
Tahapan penerimaan data merupakan fase dalam proses perolehan data dalam bentuk gambar digital atau foto yang diunduh dari platform Kaggle yang diperlukan untuk keperluan penelitian. Sumber data yang dimanfaatkan dalam studi ini ialah koleksi gambar daun kentang yang diperoleh dari Kaggle. Kumpulan data tersebut dapat diakses melalui tautan berikut: <https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld>. Sekumpulan data yang bakal digunakan dalam penelitian ini terdiri dari tiga kategori pengelompokan, yakni *Late Blight*, *Healty*, dan *Early Blight*, dengan jumlah total 4072 gambar. Jumlah tersebut mencakup 405 data uji, 3251 data latih, serta 416 data validasi.

### 3.3.3 Augmentasi

Augmentasi adalah proses memodifikasi suatu citra. Tujuan augmentasi ialah untuk menambah data citra pada proses pelatihan, misalnya citra di perkecil, atau di rotasi. Penelitian ini menggunakan augmentasi *rescale*, *horizontal flip* dan *rotation*.

### 3.3.4 Pemodelan CNN

Pada tahap ini arsitektur model CNN yang akan digunakan adalah arsitektur CNN Inception Resnet-V2, tahapan model ini dapat dilihat pada Gambar 3.2 berikut.



Gambar 3. 2 Pemodelan CNN

Langkah-langkah tersebut dapat diuraikan sebagai berikut: dalam proses pengembangan model atau pemodelan, terdapat beberapa tahap yang harus dilalui untuk mengembangkan model klasifikasi penyakit pada daun tanaman kentang, yang pertama yaitu dilakukan input citra daun tanaman kentang, kemudian dilakukan *Augmentasi* yaitu berupa *resize*, *rotation*, *rescale*, *horizontal flip*, selanjutnya dilakukan *splitting* data dengan 3 skenario data training dan data *validation* sebesar 90:10,80:20,70:30, kemudian dilakukan training CNN dan evaluasi.

### 3.3.5 Evaluasi Hasil

Dalam mengukur kinerja model dalam penelitian ini menggunakan *Precision*, *Recall* & *f1-Score* secara rata-rata. Dari hasil penilaian, pelatihan dengan metode *transfer learning* memiliki kinerja model yang lebih baik dalam melakukan klasifikasi.

Tabel 3. 1 Confusion Matrix

Aktual	Prediksi	
	+	-
+	<i>True positives</i> (A)	<i>False negatives</i> (B)
-	<i>False positives</i> (C)	<i>True negatives</i> (D)

Untuk dapat menghitung akurasi pada tabel *confusion matrix* dapat menggunakan rumus sebagai berikut :

$$\text{Akurasi} = \frac{(A+D)}{(A+B+C+D)} \quad (3.1)$$

Presisi ialah rasio item terkait yang terpilih untuk semua item yang dipilih. Oleh karena itu, ketepatan ini bisa diartikan sebagai kecocokan antara permintaan informasi dan tanggapan terhadap permintaan tersebut. Untuk dapat menghitung presisi dapat digunakan rumus sebagai berikut:

$$\text{Precision} = \frac{A}{(C+A)} \quad (3.2)$$

*Recall* merupakan rasio merupakan rasio dari item sesuai yang dipilih dari total jumlah item yang tersedia. *Recall* bisa dihitung dengan rumus sebagai berikut :

$$\text{Recall} = \frac{A}{(A+D)} \quad (3.3)$$

Presisi dan Recall bisa dievaluasi dengan perhitungan presentasi (1-100%) atau memakai angka antara 0-1. Skor F1 adalah perbandingan rata-rata presisi dan Recall (Powers & Ailab, 2011)

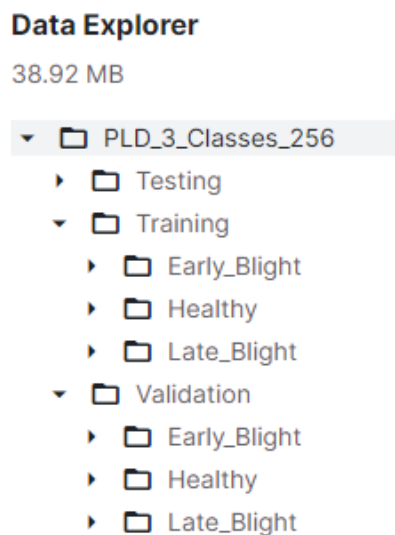
$$\text{F1Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (3.4)$$

## BAB IV HASIL DAN PEMBAHASAN

Pada Tahap ini membahas analisis dan hasil pengklasifikasian 3 kelas penyakit pada tanaman cabai berdasarkan citra daun. Metode yang digunakan adalah Convolutional Neural Network (CNN) dengan menggunakan Arsitektur Inception Resnet-V2. Tahapan yang dilakukan meliputi pengambilan data, pra-pemrosesan, perancangan model CNN, pengujian model, dan hasil klasifikasi.

### 4.1 Pengambilan Dataset

Dataset yang didapatkan ini merupakan data skunder, melalui penyedia data bersifat publik pada website Kaggle.com. Pada penelitian ini peneliti mempunyai 1 dataset (Bercak Kering (*Early Blight*)), Busuk Daun (*Late Blight*), dan Daun Sehat (*Healthy*), sehingga memiliki total class penyakit daun tanaman kentang didapatkan berjumlah 3 dengan total 4072 citra. Masing-masing label untuk *Early Blight* (bercak kering) 1303 file, *Late Blight* (busuk daun) 1132 file, *Healthy* (daun sehat) 816 file.



Gambar 4. 1 Label Dataset

Dataset yang didapatkan dari *Kaggle* dikumpulkan menjadi satu karena pada tahapan berikutnya akan dilakukan pemecahan data dalam beberapa rasio perbandingan data *training* dan validasi. Dataset dikumpulkan menjadi satu berdasarkan kategori atau label masing-masing

## 4.2 Pemecahan Data

Dalam tahapan ini dilakukan pembagian data menjadi 2 yaitu data *train* dan data *test*. Rasio pembagian data yang dipakai dalam penelitian yaitu 3 macam 90:10, 80:20, 70:30. Data *train* dipakai dalam melatih model, sedangkan data *test* dipakai dalam menguji model yang sudah dilatih. Dapat dilihat pada gambar 4.2.

```
# memuat data training dan validation
train_data = train_datagen.flow_from_directory(
    data_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=True,
    subset='training')

valid_data = train_datagen.flow_from_directory(
    data_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False,
    subset='validation')
```

Gambar 4. 2 Implementasi Pseudocode dalam proses pemecahan dataset

Pada gambar 4.2, digunakan untuk memecah dataset menjadi data *Training* dan data *Validasi*. langkah selanjutnya adalah mengatur pada parameter *train\_data* dan *valid\_data*, ada beberapa parameter didalamnya yaitu *target\_size* digunakan untuk mengatur ukuran dataset citra daun penyakit daun kentang, *class\_mode* digunakan untuk mengatur mode kelas yang digunakan. Pada dataset citra penyakit daun tanaman kentang ini memiliki 3 kelas, sehingga *class\_mode* yang digunakan adalah *categorical*, dan *shuffle* menunjukkan bahwa pengambilan sampel untuk dijadikan data training diambil secara acak dengan tujuan agar pengenalan dan pembelajaran dataset dapat dilakukan secara acak dan menyeluruh, sehingga sistem dapat mengenali berbagai jenis citra pada dataset penyakit daun tanaman kentang, sedangkan sampel untuk dijadikan data validasi tidak diambil secara acak dengan tujuan agar pada saat pembuktian menggunakan *confusion matrix*, data validasi tidak diacak kembali.

### 4.3 Augmentasi Citra

*Augmentasi* adalah proses memodifikasi suatu citra. Tujuan *augmentasi* ialah untuk menambah data citra pada proses pelatihan, misalnya citra di perkecil, atau di rotasi. Penelitian ini menggunakan *augmentasi rescale*, *horizontal flip* dan *rotation*. *Rescale* yaitu skala ulang meliputi nilai yang mempunyai banyak data sebelum lanjut ke proses selanjutnya. Citra pada penelitian ini terdiri dari koefisien *RGB* pada 0-255, tapi nilai tersebut terlalu besar untuk diproses, jadi dalam penelitian ini akan mengganti nilai tersebut antara 0 dan 1, dengan menskalakan nilai  $1/255$ . *Horizontal flip* adalah pergantian citra secara *horizontal* sebanyak 180 derajat. *Horizontal flip* dapat memudahkan penelitian pada proses augmentasi, seperti kita mengambil gambar dari kanan dan dengan fitur ini tidak usah mengambil gambar dari kiri. *Rotation* ialah suatu gerak berputar suatu benda pada porosnya seperti halnya roda yang berputar, sudut rotasi berlawanan dengan arah jarum jam. Berikut bisa dilihat pada gambar 4.3

```
# melakukan augmentasi pada data training
train_datagen = keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    zoom_range=0.1,
    horizontal_flip=True,
    validation_split=0.1)
```

Gambar 4. 3 Implementasi *Pseudocode* dalam proses augmentasi citra

### 4.4 Ekstraksi Fitur

Ekstraksi fitur adalah proses mengidentifikasi dan memilih informasi yang relevan atau mencirikan dari data yang kompleks. Ini berlaku untuk berbagai bidang, termasuk pengolahan citra, pengenalan pola, pengolahan bahasa alami. Fungsi ekstraksi fitur melibatkan konversi data yang kompleks menjadi representasi yang lebih sederhana dan bermakna. Ekstraksi fitur yang dipakai pada penelitian ini menggunakan transfer learning, yaitu *Inception ResnetV2*.

Pada tahap ini data latih dan data validasi yang sudah dipecah dalam tahap sebelumnya akan dilangsungkan ekstraksi fitur. Fitur data *train* dan valid dalam tahap sebelumnya ialah citra digital yang masuk dalam tahap ekstraksi fitur, dimana citra digital diproses dengan arsitektur *Inception Resnet V2*.

**a. Model Inception Resnet-V2**

Dalam pendekatan dengan metode transfer learning, layer untuk feature extraction menggunakan layer dari Inception ResnetV2 dan untuk output layer menggunakan *Fully connected layer* ini bertanggung jawab untuk melakukan klasifikasi pada gambar berdasarkan fitur-fitur yang telah di ekstrak sebelumnya.

Proses pertama adalah membuat bobot (*weight*) dari model yang telah dilatih sebelumnya dan hanya mengambil bagian feature extraction layer, tanpa mengambil *fully connected layer*. Untuk arsitektur Inception ResnetV2 menggunakan operasi pooling maksimum (*max pooling*) setelah beberapa blok konvolusi. *Max pooling* adalah jenis pooling yang paling umum digunakan dalam arsitektur *Convolutional Neural Network (CNN)*. Pada setiap blok konvolusi di *Inception ResnetV2*, setelah serangkaian lapisan konvolusi dilakukan, dilakukan operasi *max pooling* untuk mengurangi dimensi spasial dari fitur yang dihasilkan. Berikut ini merupakan kode untuk pembuatan model *Inception Resnet-V2* :

```

num_classes = 3

# create the base model
base_model = InceptionResNetV2(
    weights='imagenet',
    include_top=False,
    input_shape=(224, 224, 3))

base_model.trainable = False

# add custom head
inputs = keras.Input(shape=(224, 224, 3))
x = base_model(inputs, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.BatchNormalization()(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(num_classes, activation='softmax')(x)
model = keras.Model(inputs=inputs, outputs=outputs)

```

Gambar 4. 4 Implementasi *Pseudocode* pada *Inception Resnet-V2* Transfer Learning Model

Pada gambar 4.4 merupakan proses pengaturan *transfer learning* yang digunakan, dalam penelitian ini menggunakan transfer learning Inception ResNetV2 dengan input shape 224x224. Pada pengaturan input terdapat input shape dan parameter warna, pada penelitian ini menggunakan input shape 224x224 dan parameter warna sebanyak 3, yaitu RGB (Red, Green dan Blue), setelah memasuki proses input, maka dilanjutkan dengan proses pemanggilan *transfer learning Inception ResNetV2*, selanjutnya adalah menggunakan *Global Average Pooling 2D* berfungsi mengambil nilai rata-rata pada matriks yang terdapat dalam citra, setelah itu akan memasuki layer *Dense* yaitu proses klasifikasi citra berdasarkan kelas yang telah ditentukan. Penelitian ini menggunakan 4 layer, yaitu *batch normalization*, *Dropout*, *Flatten* dan *Dense*. *Normalization* ialah teknik pra-pemrosesan data yang dipakai dalam membakukan data. Jika tanpa memakai layer *batch normalization* dalam proses pelatihan data, maka data akan sulit dilatih dan *overfitting* terhadap grafik data *train* dan *valid*.

Citra akan dilakukan penyesuaian ukuran gambar sebesar 224 x 224 piksel. selanjutnya masuk ke tahap *transfer learning*. Setelah memasuki proses *transfer learning*, tahap selanjutnya adalah proses normalisasi menerapkan transformasi rata-rata mendekati 0 dan standar *deviasi* keluaran mendekati 1. Normalisasi ini bekerja secara berbeda selama proses pelatihan dan pengujian data. Setelah itu dilanjutkan melewati layer *dropout* dengan nilai 0,5. Kemudian lanjut melewati layer *flatten* untuk mengubah *array* multidimensi menjadi satu dimensi, selanjutnya melewati layer *dense* sebanyak 2 (kali) dengan nilai filter *dense* 32.

#### 4.5 Implementasi kode model

Dalam tahapan implementasi kode model, akurasi dari data training akan dihitung dengan data *test*, maka dari itu bisa diketahui performa dari model CNN yang dibuat sebelumnya. Pada penelitian ini digunakan suatu parameter dalam pengujian model, yaitu tabel *confusion matrix* untuk melihat sebaran data yang telah diuji pada tahapan ekstraksi fitur, parameter akurasi dalam melihat persentase ketepatan model untuk memprediksi citra penyakit daun kentang, parameter *precision*, *recall*, dan *F1-score* untuk melihat performa akurasi dalam setiap kelas

##### a. *Confusion Matrix*

*Confusion matrix* adalah pengukuran performa akurasi data test dalam setiap kelas yang dijabarkan pada bentuk tabel sesuai jumlah kelas yang dipakai. Pada penelitian ini terdapat 3 kelas dengan total data uji 364 citra (pada pembagian 90:10), 730 citra (dengan pembagian 80:20), dan 1095 citra (dengan pembagian 70:30)

```

# melakukan prediksi pada dataset validasi
y_pred = model.predict(valid_data)

# mengubah nilai prediksi menjadi label class dengan nilai tertinggi
y_pred = np.argmax(y_pred, axis=1)

# mengambil label sebenarnya dari dataset validasi
y_true = valid_data.classes

# mendapatkan nama class
class_names = list(valid_data.class_indices.keys())

# membuat confusion matrix
cm = confusion_matrix(y_true, y_pred)

```

Gambar 4. 5 Implementai *Pseudocode* untuk menampilkan Confusion Matrix

#### b. Akurasi (*Accuracy*)

Akurasi ialah keakuratan model atau arsitektur (*convolutional neural network*) untuk memprediksi citra. Pada evaluasi model, akurasi dinyatakan dengan persentase (%). Semakin tinggi akurasi yang didapat, maka semakin bagus model untuk memprediksi suatu citra.

```

score = model.evaluate(valid_data)
print('Validation loss:', score[0])
print('Validation accuracy:', score[1])

```

Gambar 4. 6 Implementasi *Pseudocode* akurasi

#### c. *Precision, Recall, F1-Score*

*Precision* ialah rasio prediksi benar positif dibandingkan dengan seluruh hasil yang diprediksi positif, *recall* adalah rasio prediksi benar positif dibandingkan dengan seluruh data yang benar positif, *F1-Score* ialah perbandingan rata-rata *precision* dan *recall*.

```

print("\nClassification Report:")
print(classification_report(y_true, y_pred, target_names=target_names))

```

Gambar 4. 7 Implementasi *Pseudocode* dalam menampilkan *Precision, Recall, dan F1-Score*

Pada gambar 4.7 digunakan untuk menampilkan *Classification Report* yaitu terdapat 3 parameter dalam *Classification Report* yaitu *Precision*, *Recall* dan *F1-Score*, terdapat 2 parameter dalam menentukan *Classification Report* yaitu *y\_true* dan *y\_pred*.

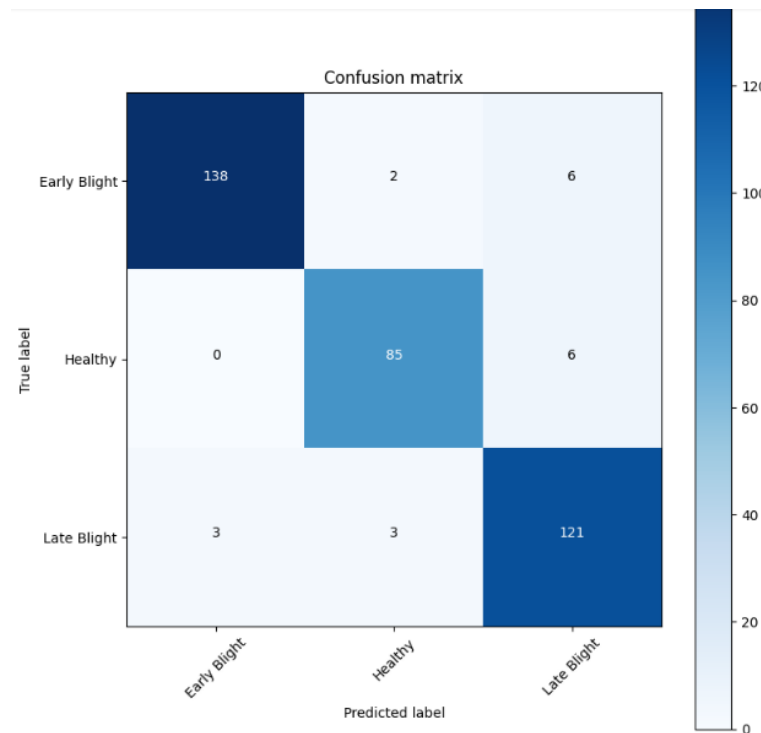
#### 4.6 Evaluasi Hasil Model

Evaluasi hasil model dipakai dalam menguji model arsitektur CNN, setelah model dilakukan pelatihan dan pengujian pada dataset penyakit daun kentang. Selanjutnya ialah menguji model. Pada penelitian ini *menggunakan confusion matrix* dalam mengetahui sebaran data uji yang bernilai benar dan salah. Dan juga terdapat parameter *precision*, *recall*, dan *F1-Score*.

Pada penelitian ini terdapat 3 (tiga) jenis rasio pembagian data, yaitu data latih dan uji sebesar 90:10, 80:20, 70:30. Dari 3 (tiga) jenis rasio pembagian data tersebut dilakukan pengujian terhadap arsitektur *Inception ResnetV2* dan akurasi tertinggi pada pembagian 90:10

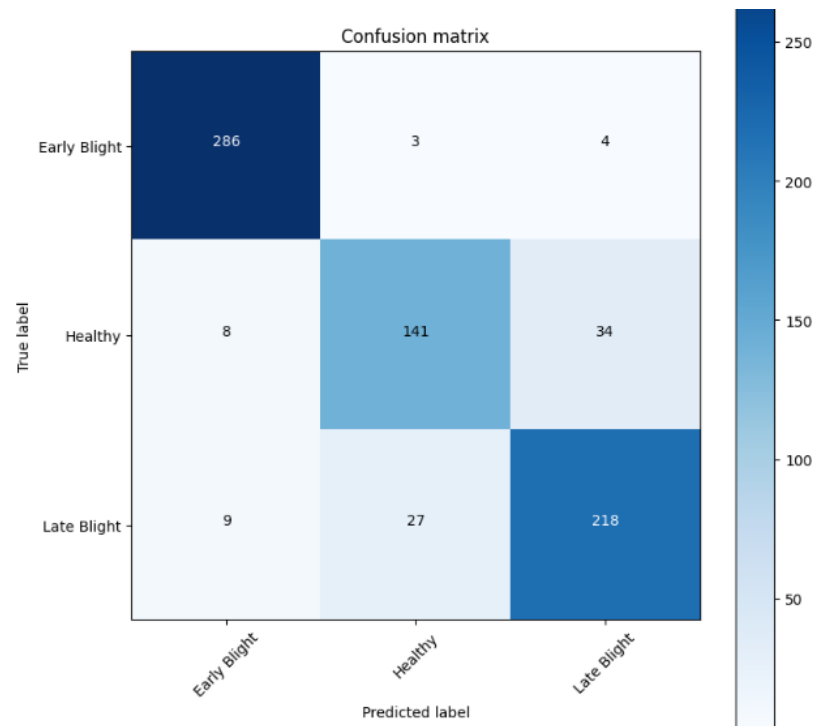
##### a. *Confusion Matrix*

*Confusion matrix* ialah sebaran data uji ke dalam bentuk tabel matriks. Pada penelitian ini *confusion matrix* dicontohkan 3 kelas pada sumbu X adalah label prediksi dan sumbu Y adalah label benar. Suatu citra dikatakan prediksi benar jika sumbu X dan sumbu Y bertemu pada huruf yang sama. Semakin tinggi nilai prediksi, maka warna akan bertambah gelap (biru tua), sedangkan semakin rendah nilai prediksi, maka warna akan bertambah terang (biru muda hingga putih).



Gambar 4. 8 Confusion Matrix Arsitektur Inception ResnetV2  
dengan pembagian data 90:10

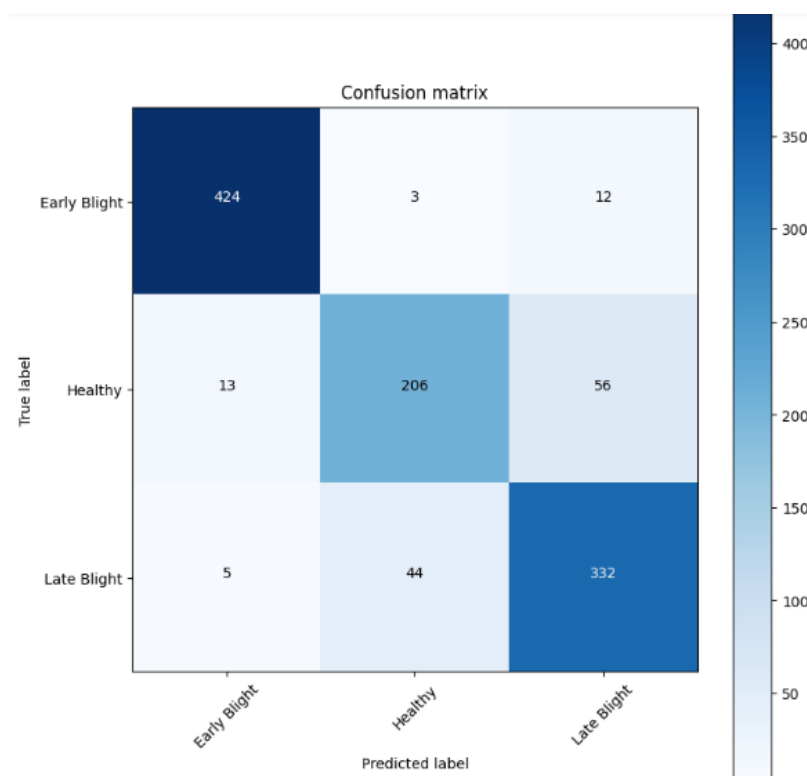
Pada gambar 4.8, disimpulkan bahwa model atau arsitektur Inception ResnetV2 dapat memprediksi dengan benar pada kelas *Early Blight*, tetapi pada model tersebut terdapat suatu kelas dengan prediksi salah, contohnya dataset yang seharusnya kelas *Early Blight* diprediksi sebagai *Healthy* sejumlah 2 citra dan kelas *Late Blight* sejumlah 6 citra, dataset yang seharusnya kelas *Healthy* diprediksi sebagai *Early Blight* sejumlah 0 citra dan kelas *Late Blight* sejumlah 6 citra, dataset seharusnya diprediksi kelas *Late Blight*, ada yang diprediksi sebagai *Early Blight* sejumlah 3 citra dan kelas *Healthy* 3 citra.



Gambar 4.9 Confusion Matrix Arsitektur Inception ResnetV2

dengan pembagian data 80:20

Pada gambar 4.9, disimpulkan bahwa model atau arsitektur Inception ResnetV2 dapat memprediksi dengan benar pada kelas *Early Blight*, tetapi pada model tersebut terdapat suatu kelas dengan prediksi salah, contohnya dataset yang seharusnya kelas *Early Blight* diprediksi sebagai *Healthy* sejumlah 3 citra dan kelas *Late Blight* sejumlah 4 citra, dataset yang seharusnya kelas *Healthy* diprediksi sebagai *Early Blight* sejumlah 8 citra dan kelas *Late Blight* sejumlah 34 citra, dataset seharusnya diprediksi kelas *Late Blight*, ada yang diprediksi sebagai *Early Blight* sejumlah 9 citra dan kelas *Healthy* 27 citra.



Gambar 4. 10 Confusion Matrix Arsitektur Inception ResnetV2  
dengan pembagian data 70:30

Pada gambar 4.10, disimpulkan bahwa model atau arsitektur Inception ResnetV2 dapat memprediksi dengan benar pada kelas *Early Blight*, tetapi pada model tersebut terdapat suatu kelas dengan prediksi salah, contohnya dataset yang seharusnya kelas *Early Blight* diprediksi sebagai *Healthy* sejumlah 3 citra dan kelas *Late Blight* sejumlah 12 citra, dataset yang seharusnya kelas *Healthy* diprediksi sebagai *Early Blight* sejumlah 13 citra dan kelas *Late Blight* sejumlah 56 citra, dataset seharusnya diprediksi kelas *Late Blight*, ada yang diprediksi sebagai *Early Blight* sejumlah 5 citra dan kelas *Healthy* 44 citra.

b. *Precision, Recall* dan *F1-Score*

Setelah data uji dideskripsikan dalam bentuk tabel *confusion matrix*, maka pada poin b ini, dari tabel *confusion matrix* dapat dihitung parameter *precision*, *recall* dan *F1-score* sesuai dengan kategori klasifikasinya. Semakin tinggi nilai *precision* dan *recall* maka semakin tinggi pula hasil pada parameter *F1-score*. Namun, semakin rendah pada salah satu nilai

precision dan recall atau keduanya, maka akan berpengaruh terhadap nilai F1-score menjadi rendah.

Classification Report:				
	precision	recall	f1-score	support
Early Blight	0.98	0.95	0.96	146
Healthy	0.94	0.93	0.94	91
Late Blight	0.91	0.95	0.93	127
accuracy			0.95	364
macro avg	0.94	0.94	0.94	364
weighted avg	0.95	0.95	0.95	364

Gambar 4. 11 Classification Report Arsitektur Inception ResnetV2

pada Pembagian Data (90:10)

Pada gambar 4.11, bisa disimpulkan bahwa nilai *precision* paling tinggi bernilai 98% yaitu kelas *Early Blight* sedangkan nilai *precision* paling rendah yaitu kelas *Late Blight* yaitu bernilai 91%. Untuk parameter *recall* paling tinggi bernilai 95% yaitu kelas *Early Blight* dan *Late Blight*, Dan nilai *recall* paling rendah yaitu kelas *Healthy* bernilai 93%, sedangkan nilai F1-score paling tinggi yaitu kelas *Early Blight* sebesar 96% dan nilai F1-score paling rendah yaitu kelas *Late Blight* sebesar 93%.

Classification Report:				
	precision	recall	f1-score	support
Early Blight	0.94	0.98	0.96	293
Healthy	0.82	0.77	0.80	183
Late Blight	0.85	0.86	0.85	254
accuracy			0.88	730
macro avg	0.87	0.87	0.87	730
weighted avg	0.88	0.88	0.88	730

Gambar 4. 12 Classification Report Arsitektur Inception ResnetV2

Pada Pembagian Data (80:20)

Pada gambar 4.12, dapat disimpulkan bahwa nilai *precision* paling tinggi 94% yaitu kelas *Early Blight*, sedangkan nilai *precision* paling

rendah yaitu kelas *Healthy* dengan nilai 0,82. Untuk parameter *recall* paling tinggi dengan nilai 98% yaitu kelas *Early Blight*, dan nilai *recall* paling rendah yaitu kelas *Healthy*, dengan nilai 0,77, sedangkan nilai F1-score paling tinggi yaitu kelas *Early Blight* dengan nilai 0,96, nilai F1-score paling rendah yaitu kelas *Healthy* dengan nilai 0,80.

Classification Report:				
	precision	recall	f1-score	support
Early Blight	0.96	0.97	0.96	439
Healthy	0.81	0.75	0.78	275
Late Blight	0.83	0.87	0.85	381
accuracy			0.88	1095
macro avg	0.87	0.86	0.86	1095
weighted avg	0.88	0.88	0.88	1095

Gambar 4. 13 Classification Report Arsitektur Inception ResnetV2




Pada Pembagian Data (70:30)

Pada gambar 4.13, dapat disimpulkan bahwa nilai *precision* paling tinggi 96% yaitu kelas *Early Blight*, sedangkan nilai *precision* paling rendah yaitu kelas *Healthy* dengan nilai 0,81. Untuk parameter *recall* paling tinggi dengan nilai 97% yaitu kelas *Early Blight*, dan nilai *recall* paling rendah yaitu kelas *Healthy* dengan nilai 0,75, sedangkan nilai F1-score paling tinggi yaitu kelas *Early Blight* dengan nilai 0,96, nilai F1-score paling rendah yaitu kelas *Healthy* dengan nilai 0,78.

#### 4.7 Sistem Inferensi

Citra yang digunakan yaitu berasal dari Kaggle. Pada percobaan penelitian ini citra yang digunakan adalah 3 kelas yaitu *Early Blight*, *Late Blight* dan *Healthy*. Terdapat masing-masing percobaan pada sistem ini, pada tabel 4.1 akan diuraikan citra yang digunakan dalam menguji sistem ini.

Tabel 4. 1 Penjelasan Citra yg digunakan dalam Menguji Sistem Inferensi

Nama kelas	Gambar	Penjelasan	Hasil Uji	Kesimpulan
Early Blight		Citra mempunyai sedikit bercak kering di daun	Healthy	Salah memprediksi karena pada citra Early Blight tingkat kemiripan hamper sama dengan healthy.
Late Blight		Citra daun memiliki busuk daun	Late Blight	Model dapat memprediksi citra Late Blight dengan benar.
Healthy		Citra daun bersih hijau dan terlihat sehat	Healthy	Model dapat memprediksi citra healthy dengan benar.

Setelah dataset untuk menguji sistem disiapkan, kemudian menuju proses *preprocessing* untuk *resize* ukuran citra menjadi 224 x 224. Kemudian citra

dimasukkan satu persatu kedalam sistem, sistem akan memprediksi citra tersebut dengan menampilkan *output* berupa kelas.

## BAB V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil pembuatan model dan telah dilakukan pelatihan pada dataset penyakit daun tanaman kentang pada penelitian ini dengan menggunakan arsitektur Inception Resnet-V2, maka akan dapat ditarik kesimpulan sebagai berikut :

1. Penelitian ini mengimplementasikan arsitektur Inception Resnet-V2 dengan transfer learning untuk mengklasifikasikan daun tanaman kentang, termasuk *Early Blight*, *Healthy*, dan *Late Blight*. Penelitian ini membandingkan tiga skenario pembagian data yaitu 70%:30%, 80%:20%, dan 90%:10%. Hasil penelitian menunjukkan bahwa skenario terbaik ditemukan pada pembagian data 90%:10%.
2. Berdasarkan pengujian menggunakan Precision, Recall, F1-score, dan Akurasi model klasifikasi penyakit daun tanaman kentang dengan menggunakan arsitektur Inception Resnet-V2, yaitu terbaik dari keseluruhan skenario untuk precision 94%, recall 94%, f1-score 94% dan akurasi 95%.

### 5.2 Saran

Adapun saran yang diberikan berdasarkan dari penelitian ini sebagai upaya perbaikan dan pengembangan pada penelitian lanjutan adalah sebagai berikut:

1. Peneliti selanjutnya diharapkan dapat membuat sebuah perangkat lunak aplikasi berbasis website maupun mobile yang user-friendly dan mudah digunakan untuk klasifikasi penyakit daun tanaman kentang. Hal ini akan memungkinkan para petani dan ahli pertanian untuk dengan mudah mendeteksi dan mengidentifikasi penyakit daun tanaman kentang dengan cepat dan akurat.
2. Penelitian mendatang perlu melengkapi analisis citra untuk kategori jenis penyakit daun tanaman kentang yang belum tercakup sebelumnya, sehingga memberikan pemahaman yang lebih komprehensif tentang berbagai penyakit daun tanaman kentang.

## DAFTAR PUSTAKA

- Alamsyah, Derry, dan Dicky Pratama. 2020. “Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada FER-2013 Dataset.” *Jurnal Teknologi Informasi* 4 (2): 350–55. <https://doi.org/10.36294/jurti.v4i2.1714>.
- Inception-resnetv, Perbandingan Kinerja, D A N Resnet, Pada Gambar, dan Bentuk Wajah. 2023. “Performance Comparison of Inception-Resnetv2 , Xception , Inception-V3 , and Resnet50 on Face Shape Images” 10 (1): 2. <https://doi.org/10.25126/jtiik.2023104941>.
- Kamble, Ravi M., Manesh Kokare, Genevieve C.Y. Chan, Oscar Perdomo, Fabio A. González, Henning Müller, dan Fabrice Mériaudeau. 2019. “Automated Diabetic Macular Edema (DME) analysis using fine tuning with inception-resnet-v2 on oct images.” *2018 IEEE EMBS Conference on Biomedical Engineering and Sciences, IECBES 2018 - Proceedings* cl: 442–46. <https://doi.org/10.1109/IECBES.2018.8626616>.
- Nugroho, Pulung Adi, Indah Fenriana, dan Rudy Arijanto. 2020. “Implementasi Deep Learning Menggunakan Convolutional Neural Network ( Cnn ) Pada Ekspresi Manusia.” *Algor* 2 (1): 12–21.
- Nurona Cahya, Fani, Nila Hardi, Dwiza Riana, Sri Hadiani, Nusa Mandiri Jakarta Cipinang Melayu, Kec Makasar, Kota Jakarta Timur, dan Daerah Khusus Ibukota Jakarta. 2021. “SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network ( CNN)” 10: 618–26. <http://sistemasi.ftik.unisi.ac.id>.
- Pradhitya, Rio. 2015. “Pembangunan Aplikasi Deteksi dan Tracking Warna Virtual Drawing Menggunakan Algoritma Color Filtering Jurnal Ilmiah Komputer dan Informatika ( KOMPUTA ).” *Jurnal Ilmiah Komputer dan Informatika*.
- Prakosa, Andhika Bagas, Fakultas Teknologi Informasi, Universitas Kristen, dan Satya Wacana. 2023. “IMPLEMENTASI MODEL DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK ( CNN ) PADA CITRA

PENYAKIT DAUN JAGUNG,” no. April: 107–16.

- Putri, Andi Besse Sri, Hajrah Hajrah, Devi Armita, dan Ika Roostika Tambunan. 2021. “Teknik kultur jaringan untuk perbanyak dan konservasi tanaman kentang (*Solanum tuberosum* L.) secara in vitro.” *Filogeni: Jurnal Mahasiswa Biologi* 1 (2): 69–76. <https://doi.org/10.24252/filogeni.v1i2.23801>.
- Rozaqi, Abdul Jalil, Andi Sunyoto, dan Rudyanto Arief. n.d. “Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network Detection of Potato Leaves Disease Using Image Processing with Convolutional Neural Network Methods.”
- Thomas, Anju, P. M. Harikrishnan, P. Palanisamy, dan Varun P. Gopi. 2020. “Moving Vehicle Candidate Recognition and Classification Using Inception-ResNet-v2.” *Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020*, 467–72. <https://doi.org/10.1109/COMPSAC48688.2020.0-207>.
- Wahyutama Fitri Hidayat, dan Taufik Asra. 2022. “Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression.” *Indonesian Journal on Software Engineering* 8 (2): 173–79.
- Winarto, Eveline Gabriela, Rahmayati, dan Armin Lawi. 2021. “Implementasi Arsitektur Inception Resnet-V2 untuk Klasifikasi Kualitas Biji Kakao.” *Konferensi Nasional Ilmu Komputer (KONIK) 2021*, 132–37.
- Wonohadidjojo, Daniel Martomanggolo. 2021. “Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih.” *Ultimatics: Jurnal Teknik Informatika* 13 (1): 51–57. <https://doi.org/10.31937/ti.v13i1.2040>.

## Lampiran

```

Model: "model"

```

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
inception_resnet_v2 (Functional)	(None, 5, 5, 1536)	54336736
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
batch_normalization_203 (BatchNormalization)	(None, 1536)	6144
dropout (Dropout)	(None, 1536)	0
dense (Dense)	(None, 3)	4611

```

=====
Total params: 54347491 (207.32 MB)
Trainable params: 7683 (30.01 KB)
Non-trainable params: 54339808 (207.29 MB)

```

Lampiran 1 Arsitektur Inception ResnetV2 Transfer Learning