

## BAB III METODOLOGI PENELITIAN

### 3.1 Subyek dan Obyek Penelitian

Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi penyakit pada daun kentang. Data yang digunakan adalah *dataset PlantVillage* yang diunduh pada situs Kaggle yang didapatkan pada *link* berikut <https://www.kaggle.com/emmarex/plantdisease>.

### 3.2 Alat dan Bahan Penelitian

#### 3.2.1 Alat

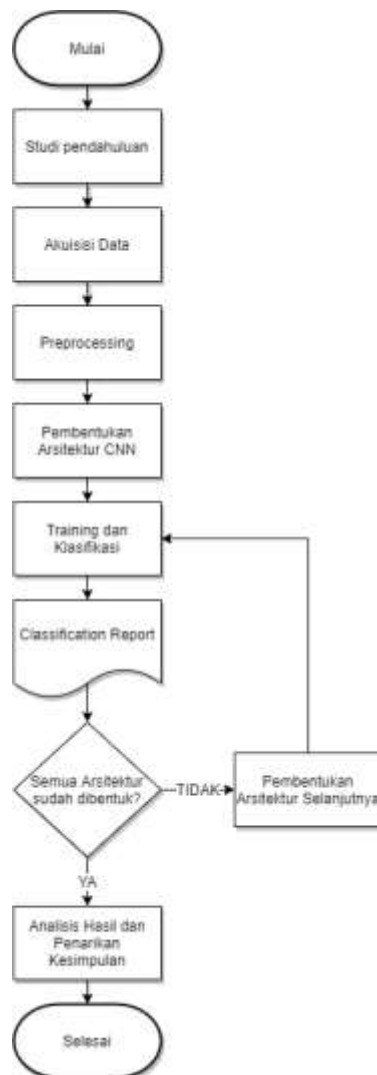
- a. Laptop dengan spesifikasi :
  1. Intel(R) Core(TM) i5-8300H CPU @ 2.30 GHz
  2. RAM 16GB
  3. Intel UHD Graphics 630 VRAM 128 MB
  4. NVIDIA GeForce GTX 1050 VRAM 4 GB
- b. Alat perangkat lunak
  1. Jupyter Notebook
  2. Python 3.6.5 (Anaconda)
  3. Windows 10 OS

#### 3.2.2 Bahan

Bahan penelitian berupa *dataset PlantVillage* yang diunduh pada situs Kaggle yang berisi beraneka ragam kelas penyakit dan jenis tanaman yang didapatkan pada *link* berikut didapatkan pada *link* berikut <https://www.kaggle.com/emmarex/plantdisease>. Kemudian diambil hanya 3 kelas saja yaitu : *potato early blight*, *potato healthy*, dan *potato late blight*, yang setelah dilakukan *preprocessing* data yang digunakan berjumlah 1000 tiap kelasnya.

### 3.3 Proses Penelitian

Terdapat beberapa tahap dalam pelaksanaan penelitian ini, diawali dengan melakukan studi pendahuluan yang bertujuan untuk dijadikan pondasi dalam penyusunan penelitian. Selanjutnya yaitu akuisisi data yang diperlukan pada penelitian ini. Tahap ketiga dilakukan *preprocessing* pada data sehingga data siap digunakan. Tahap keempat adalah training dan klasifikasi menggunakan *Convolutional Neural Network* pada data yang sudah dilakukan *preprocessing*. Tahap terakhir adalah analisis, evaluasi hasil, dan penarikan kesimpulan. Diagram proses penelitian dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Proses Penelitian**

### 3.3.1 Studi Pendahuluan

Studi pendahuluan dilakukan dengan melakukan studi literatur dari berbagai sumber yang ada seperti jurnal, buku, maupun penelitian-penelitian terdahulu yang masih berhubungan dengan penelitian ini baik secara objek yang diteliti maupun metode yang digunakan. Studi pendahuluan ini menjadi dasar pengetahuan untuk melakukan penelitian ini.

### 3.3.2 Akuisisi Data

Akuisisi data dilakukan dengan mengunduh *PlantVillage dataset* yang tersedia di situs Kaggle. *Dataset* dapat diunduh pada link berikut <https://www.kaggle.com/emmarex/plantdisease> [22]. *Dataset* ini berisi beraneka ragam kelas jenis dan penyakit tanaman, hanya 3 kelas yang diambil dari *dataset* ini yaitu: *potato early blight* yang berjumlah 1000 gambar, *potato healthy* yang berjumlah 152 gambar, *potato late blight* 1000 gambar. Gambar-gambar pada *dataset* ini memiliki rasio gambar 1:1 dengan ukuran 256x256 piksel.

### 3.3.3 Preprocessing

*Preprocessing* dilakukan dengan melakukan pembagian *dataset* dan augmentasi. *Dataset* akan dibagi menjadi 60% data *training*, 20% data *validation*, dan 20% data *testing*. Kemudian dilakukan augmentasi data, gambar akan diperbanyak dengan cara merotasi gambar sebanyak maksimal 20 derajat, karena rotasi bukan kelipatan 90 derajat maka hasil augmentasi akan otomatis dilakukan *cropping* oleh *library Augmentor*. Data yang diaugmentasi adalah data pada kelas *potato healthy* karena pada kelas tersebut hanya terdapat 152 data gambar. Sehingga pada akhir proses ini gambar mencapai jumlah 1000 gambar tiap kelas atau total 3000 gambar. Jumlah pasti pembagian data *training*, data *validation* dan data *testing* dapat dilihat pada Tabel 3.1. Selanjutnya data akan dinormalisasi dengan mengubah nilai skala piksel dari *range* 0-255 menjadi 0-1 dan kemudian dilakukan perubahan ukuran data dari 256x256 piksel ke 128x128 piksel sesuai dengan ukuran *input* pada arsitektur yang digunakan.

**Tabel 3.1 Pembagian Dataset**

No	Kelas	Train	Validation	Testing	Jumlah
1	Early Blight	600	200	200	1000
2	Healthy	600	200	200	1000
3	Late Blight	600	200	200	1000
Jumlah		1800	600	600	3000

### 3.3.4 Training dan Klasifikasi menggunakan *Convolutional Neural Network*

Pada tahap ini akan dilakukan *training* dan klasifikasi menggunakan *Convolutional Neural Network*. *Training* pada tiap arsitektur dilakukan sampai *training accuracy* tidak menunjukkan kenaikan lagi. Proses ini akan dilakukan pada 9 arsitektur yang masing-masing menggunakan ukuran *convolution layer* 3x3, 5x5, dan 7x7 dan juga menggunakan jumlah *convolution layer* 2, 3, dan 4.

Setelah dilakukan *training*, model pada 9 arsitektur akan diuji menggunakan data *testing* dan akan didapatkan *accuracy* dan *f1-score* yang pada tahap selanjutnya akan di bandingkan untuk mengetahui pengaruh ukuran *convolution layer* terhadap akurasi dan *f1-score* dan pengaruh jumlah *convolution layer* terhadap akurasi dan *f1-score*. Untuk memudahkan dalam penyebutan arsitektur, maka dilakukan penamaan arsitektur sesuai dengan ukuran dan jumlah *convolution layer* yang digunakan. Tabel 3.2 merupakan arsitektur-arsitektur yang akan digunakan pada penelitian ini.

**Tabel 3.2 Arsitektur-Arsitektur CNN yang digunakan**

No	Nama Arsitektur	Jumlah <i>Convolution Layer</i>	Ukuran <i>Convolution Layer</i>	Jumlah <i>filter</i> tiap <i>convolution layer</i>	Pooling tiap <i>convolution layer</i>	<i>Input Size</i>
1	3x3-2	2	3x3	8, dan 16	Max Pooling 2x2	128x128 x3

No	Nama Arsitektur	Jumlah Convolution Layer	Ukuran Convolution Layer	Jumlah filter tiap convolution layer	Pooling tiap convolution layer	Input Size
2	3x3-3	3	3x3	8, 16, dan 32	Max Pooling 2x2	128x128 x3
3	3x3-4	4	3x3	8, 16, 32 dan 64	Max Pooling 2x2	128x128 x3
4	5x5-2	2	5x5	8, dan 16	Max Pooling 2x2	128x128 x3
5	5x5-3	3	5x5	8, 16, dan 32	Max Pooling 2x2	128x128 x3
6	5x5-4	4	5x5	8, 16, 32, dan 64	Max Pooling 2x2	128x128 x3
7	7x7-2	2	7x7	8, dan 16	Max Pooling 2x2	128x128 x3
8	7x7-3	3	7x7	8, 16, dan 32	Max Pooling 2x2	128x128 x3
9	7x7-4	4	7x7	8, 16, 32, dan 64	Max Pooling 2x2	128x128 x3

### 3.3.5 Analisis hasil dan Penarikan Kesimpulan

Pada tahapan ini akan dilakukan analisis pada hasil *testing* pada arsitektur-arsitektur CNN. Analisis akan dilakukan dengan melihat nilai *f1-score* dan *accuracy* yang didapatkan dengan *confusion matrix* pada tiap-tiap arsitektur sehingga akan disimpulkan arsitektur mana yang memiliki akurasi dan *f1-score* terbaik. Penggunaan kedua metrik tersebut berdasarkan penelitian sebelumnya yang juga menggunakan akurasi dan *f1-score* [14].

Kemudian akan dilakukan juga analisis pengaruh ukuran *convolution layer* terhadap akurasi dan *f1-score* dengan membandingkan arsitektur yang

memiliki ukuran *convolution layer* berbeda namun memiliki jumlah *convolution layer* yang sama. Arsitektur 3x3-2 akan dibandingkan dengan arsitektur 5x5-2 dan 7x7-2, arsitektur 3x3-3 akan dibandingkan dengan arsitektur 5x5-3 dan 7x7-3, dan arsitektur 3x3-4 akan dibandingkan dengan arsitektur 5x5-4 dan 7x7-4.

Selain itu, akan dilakukan juga analisis pengaruh jumlah *convolution layer* terhadap akurasi dan *f1-score* dengan membandingkan arsitektur yang memiliki jumlah *convolution layer* berbeda namun memiliki ukuran *convolution layer* yang sama. Arsitektur 3x3-2 akan dibandingkan dengan arsitektur 3x3-3 dan arsitektur 3x3-4, Arsitektur 5x5-2 akan dibandingkan dengan arsitektur 5x5-3 dan 5x5-4, dan arsitektur 7x7-2 akan dibandingkan dengan arsitektur 7x7-3 dan 7x7-4.