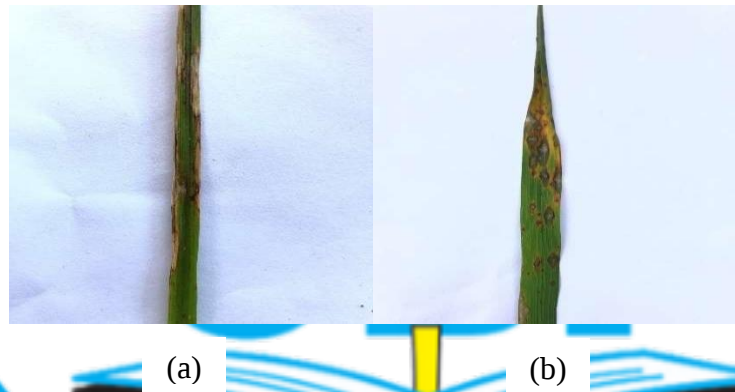


BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian yang digunakan dalam penelitian ini adalah penyakit daun padi Hawar Daun Bakteri dan *Blast* Daun, pengambilan sampel dari situs Kaggle.Com dengan pemilik sumber data Shayan Riyaz, Nizor Ogbezuode, Vbookshelf, yang berupa sebuah gambar Hawar Daun Bakteri dan *Blast* Daun dengan jumlah 1000 data gambar, diantaranya 500 data Hawar Daun Bakteri dan 500 data *Blast* Daun. Berikut contoh sampel yang telah diambil:



Gambar 3. 1 (a) Citra Hawar Daun Bakteri (b) Citra *Blast* Daun

3.2. Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di Dinas Pertanian Karawang, sejak bulan November 2022. Detail pelaksanaan penelitian dijelaskan pada tabel 3.1.

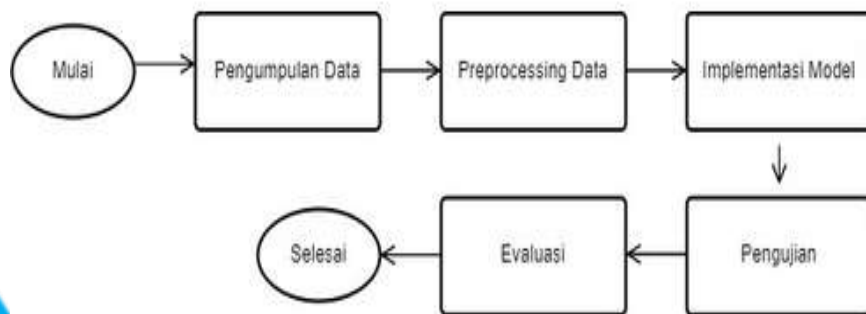
Tabel 3. 1 Lokasi & Waktu Penelitian

No	Proses	Nov	Des	Jan	Feb	Mar
		2022	2022	2023	2023	2023
1	Pengumpulan Data					
2	<i>Preprocessing</i> Data					
3	Implementasi Model					
4	Pengujian					

No	Proses	Nov 2022	Des 2022	Jan 2023	Feb 2023	Mar 2023
5	Evaluasi					

3.3. Prosedur Penelitian

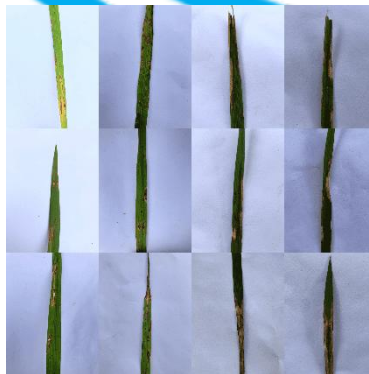
Prosedur penelitian dimulai dengan melakukan pengumpulan data, setelah pengumpulan data masuk tahap *preprocessing data*, kemudian implementasi model, lalu pengujian, dan yang terakhir dengan melakukan evaluasi.



Gambar 3. 2 Prosedur Penelitian

3.3.1. Pengumpulan Data

Pengumpulan data dalam penelitian ini diperoleh dari *website* Kaggle.com, dengan pemilik sumber data Shayan Riyaz, Nizor Ogbezuode, Vbookshelf. Data yang dikumpulkan berupa 500 gambar penyakit daun padi Hawar Daun Bakteri dan 500 gambar *Blast* Daun, dengan jumlah total keseluruhan 1000 data citra.



Gambar 3. 3 Sampel Data Kaggle.Com

3.3.2. *Preprocessing Data*

Preprocessing data bertujuan untuk mengubah data yang masih mentah agar menjadi suatu informasi yang lebih bersih atau mudah dipahami sebelum masuk ke tahap proses selanjutnya. *Preprocessing* data dilakukan *splitting dataset*, *dataset* dibagi menjadi 3 kelas yang terdiri dari data *training*, data *validation*, dan data *test*. Kemudian augmentasi data juga diterapkan untuk mencegah *overfitting* serta menambah jumlah data dengan cara memodifikasi data citra, yang menggunakan *Image Data Generator*. *Resize* dilakukan terlebih dahulu sebelum masuk augmentasi, dengan mengubah dimensi semua citra menjadi 256x256 piksel, supaya nantinya saat implementasi model tidak terlalu berat prosesnya. Tahapan augmentasi data yang diusulkan mengacu pada penelitian yang dilakukan (Pradika, Nugroho & Puspaningrum, 2020), berikut tahapan-tahapan augmentasi:

1. *Rescale*, sebagai mengubah skala citra.
2. *Rotation_range*, untuk merotasi/memutar citra.
3. *Width_shift_range*, sebagai fungsi menentukan seberapa besar gambar digeser secara acak *horizontal*.
4. *Height_shift_range*, digunakan untuk menentukan seberapa besar gambar digeser secara acak *vertical*.
5. *Shear_range*, sebagai fungsi memiringkan gambar dalam satu sumbu.
6. *Zoom_range*, digunakan untuk memperbesar gambar.
7. *Horizontal_flip*, membalikkan gambar secara *horizontal*.
8. *Fill_mode*, berfungsi mengisi wilayah/gambar yang tidak bernilai.

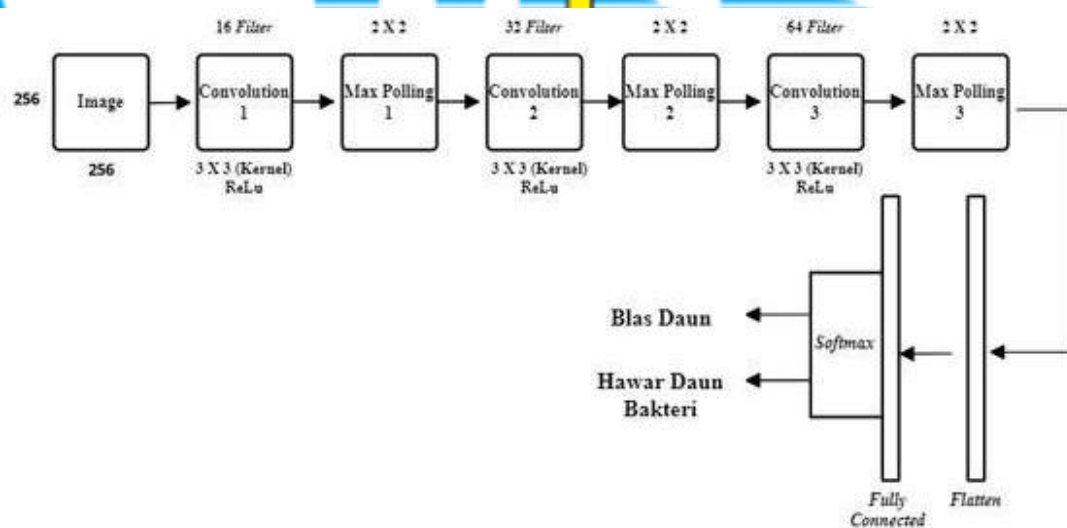


Gambar 3. 4 Contoh Augmentasi Data

3.3.3. Implementasi Model

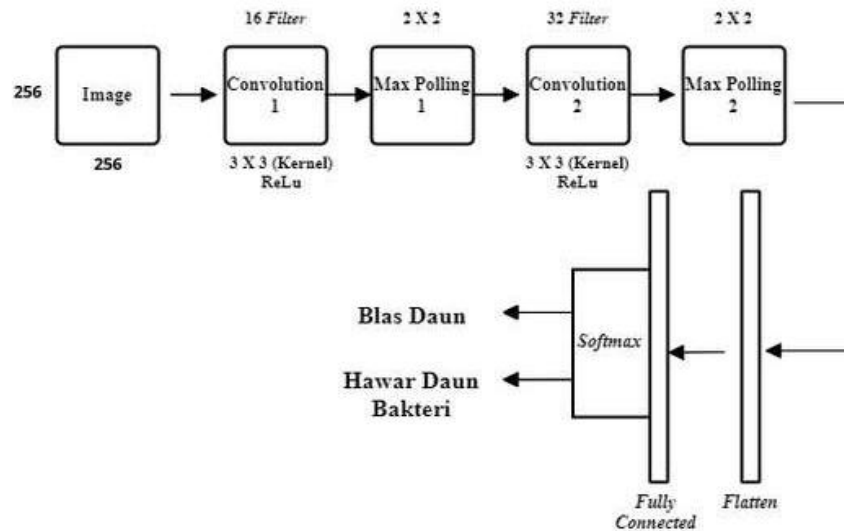
Implementasi model yaitu mengidentifikasi jenis penyakit daun padi Hawar Daun Bakteri dan *Blast* Daun. Perangkat keras yang digunakan dalam implementasi model berupa laptop dengan sistem operasi Windows 10 64-bit, RAM 4 GB, Processor Intel Core i3 5005U. Kemudian perangkat lunak yang digunakan dalam penelitian ini menggunakan *Jupyter Notebook*. Lalu pembuatan model di awali membagi data dengan berbagai macam perbandingan. Pembuatan model selanjutnya dilakukan dengan metode CNN (*Convolutional Neural Network*). Arsitektur CNN yang akan dibuat, menggunakan dengan beberapa arsitektur model:

1. Arsitektur Model 1



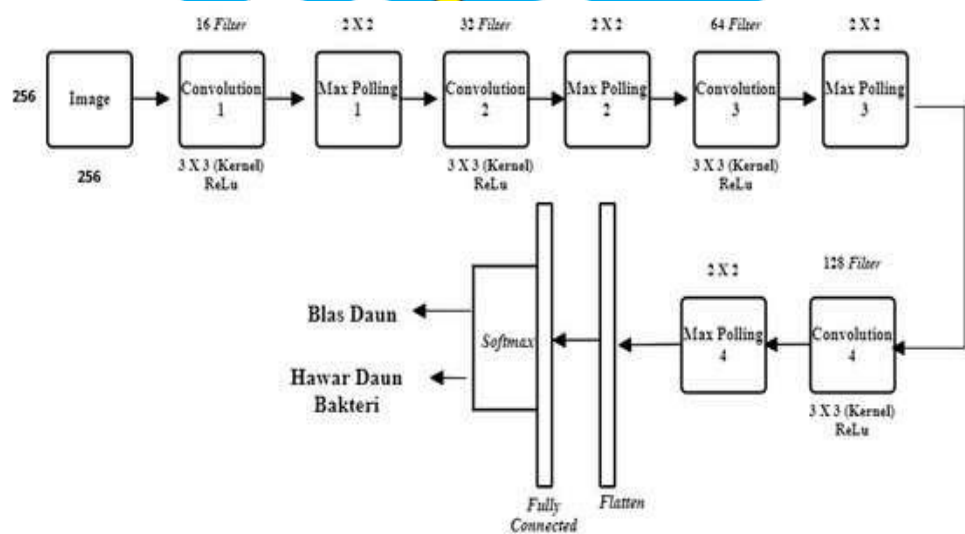
Gambar 3. 5 Pembuatan Arsitektur Model 1

2. Arsitektur Model 2



Gambar 3. 6 Pembuatan Arsitektur Model 2

3. Arsitektur Model 3



Gambar 3. 7 Pembuatan Arsitektur Model 3

Berdasarkan arsitektur model CNN yang dibuat seperti pada gambar 3.5, gambar 3.6, serta gambar 3.7, tahapan-tahapannya dirincikan sebagai berikut:

1. Tahap Konvolusi:

- a) Arsitektur Model 1 (3 Konvolusi Layer): Layer pertama menggunakan jumlah filter 16, kemudian konvolusi layer ke dua jumlah filter 32, dan konvolusi layer yang ke tiga menggunakan 64

filter. Jumlah kernel 3x3 dipakai pada setiap tahap konvolusi, serta aktivasi ReLu untuk menghilangkan nilai negatif.

- b) *Arsitektur Model 2 (2 Konvolusi Layer):* *Layer* pertama menggunakan jumlah filter 16, kemudian konvolusi *layer* ke dua dengan jumlah filter 32. Jumlah kernel 3x3 dipakai pada setiap tahap konvolusi, serta aktivasi ReLu untuk menghilangkan nilai negatif.
- c) *Arsitektur Model 3 (4 Konvolusi Layer):* *Layer* pertama menggunakan jumlah filter 16, kemudian konvolusi *layer* ke dua jumlah filter 32, dan konvolusi *layer* yang ke tiga menggunakan 64 filter, Lalu pada konvolusi *layer* ke empat menggunakan 128 filter. Jumlah kernel 3x3 dipakai pada setiap tahap konvolusi, serta aktivasi ReLu untuk menghilangkan nilai negatif.

2. Tahap *Polling*:

- a) *Arsitektur Model 1 (3 Polling Layer):* *Max pooling* digunakan pada setiap *layer* untuk mengambil nilai piksel paling tinggi, dengan nilai *matrix* 2x2.
- b) *Arsitektur Model 2 (2 Polling Layer):* *Max pooling* digunakan pada setiap *layer* untuk mengambil nilai piksel paling tinggi, dengan nilai *matrix* 2x2.
- c) *Arsitektur Model 3 (4 Polling Layer):* *Max pooling* digunakan pada setiap *layer* untuk mengambil nilai piksel paling tinggi, dengan nilai *matrix* 2x2.

3. Tahap *Fully Connected Layer*:

Pada tahap *fully connected layer* semua arsitektur model melakukan tahapan-tahapan yang sama, *flattening* dilakukan terlebih dahulu, dengan tujuan mengubah *feature map* yang masih berbentuk *multidimensional array* menjadi data satu dimensi *single vector*. Selanjutnya dilakukan *dropout* dengan nilai 0,5, bertujuan memilih acak beberapa neuron yang tidak akan digunakan selama proses pelatihan, agar mengurangi *overfitting* saat proses *training*. Kemudian masuk ke proses *dense*, berfungsi untuk menambahkan *layer* yang *fully connected* dengan nilai 512, sebagai jumlah node

yang harus ada di *hidden layer*, lalu aktivasi yang digunakan ReLu. Aktivasi pada *layer* terakhir menggunakan aktivasi *softmax*.

3.3.4. Pengujian

Pengujian dilakukan pada data *training* dan data *validation*, kemudian supaya nantinya mendapatkan *training* akurasi serta *validation* akurasi. Pengujian model menggunakan fungsi *early stopping* untuk mencegah terjadinya *overfitting* model. *Early stopping* dibutuhkan supaya nantinya akan menghentikan proses *training* ketika terjadi perbedaan antara performa *training* dengan validasi melebihi batasan yang ditentukan, sehingga ketika model akan *overfitting* proses *training* akan langsung dihentikan. Berikut beberapa *hyperparameter* yang akan digunakan dalam proses pengujian:

- 1) *Batch_size*, merupakan jumlah sampel *training* untuk sekali literasi yang digunakan dalam satu *batch*.
- 2) *Epoch*, merupakan banyaknya literasi pada proses *training*.
- 3) *Class_Mode*, yang merupakan metode untuk pemilihan klasifikasi.
- 4) *Target_size*, adalah dimensi dari sebuah citra yang akan dipakai dalam proses *training*.
- 5) *Monitor*, yaitu proses yang akan dipantau.
- 6) *Patience*, jumlah batas yang dipilih untuk menghentikan pelatihan saat model tidak *improvement*.

Tabel 3. 2 Pengujian Model

Pengujian Model	Nilai
<i>Epoch</i>	100
<i>Batch_size</i>	32
<i>Class Mode</i>	<i>Categorical</i>
<i>Monitor</i>	<i>Val_loss</i>
<i>Patience</i>	5

Pengujian Model	Nilai
Target Size	256, 256

3.3.5. Evaluasi

Tahap evaluasi dilakukan untuk setiap proses pendeteksian jenis penyakit daun padi yang menggunakan metode CNN. Proses evaluasi dengan menerapkan fungsi *confusion matrix* untuk mengetahui akurasi dari pembuatan model tersebut. Berikut persamaan untuk mencari akurasi:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

Pada tahap pendeteksian jenis penyakit padi Hawar Daun Bakteri dan *Blast* Daun, *True Positive* (TP) merupakan data jenis penyakit daun padi Hawar Daun Bakteri, dan model mengklasifikasi Hawar Daun Bakteri. *True Negative* (TN) data jenis penyakit daun padi *Blast* Daun, lalu model mengklasifikasi bukan Hawar Daun Bakteri. *False Positive* (FP) data jenis penyakit daun padi *Blast* Daun, tapi model mengklasifikasi Hawar Daun Bakteri. Kemudian *False Negative* (FN) yang merupakan data penyakit daun padi Hawar Daun Bakteri, namun model mengklasifikasi *Blast* Daun.