

BAB III

METODOLOGI PENELITIAN

3.1 Subjek dan Objek Penelitian

Pada Bab I diterangkan adanya latar belakang masalah yang akan diteliti, yaitu hama ulat pada daun sawi hijau di Indonesia. Subjek dari penelitian ini adalah hama ulat pada daun sawi hijau. Sementara objek penelitian ini adalah deteksi hama ulat dengan menggunakan *Convolutional Neural Network* (CNN) dengan metode *transfer learning*, sesuai dengan batasan masalah pada Bab I.

3.2 Alat dan Bahan Penelitian

Dalam penelitian ini diperlukan spesifikasi minimum dari peralatan dan bahan yang digunakan seperti data, perangkat lunak, perangkat keras, yaitu sebagai berikut.

3.2.1 Data

Pada penelitian ini menggunakan *dataset caisim* yang diperoleh dari Kaggle <https://www.kaggle.com/datasets/cendekialuthfietanz/caisim> [50]. Data yang dianalisis dan digunakan pada penelitian ini berupa gambar daun sawi hijau dengan hama ulat dan tanpa hama ulat. Adapun jumlah *dataset* yang digunakan dapat dilihat pada Tabel 3.1 berikut ini.

Tabel 3. 1 *Dataset Caisim*

No	Label	Jumlah
1.	Data Sawi Ada Hama	500
2.	Data Sawi Tanpa Hama	500
Total Jumlah		1000

3.2.2 Spesifikasi Kebutuhan Perangkat Keras

Berikut ini merupakan spesifikasi perangkat keras yang dibutuhkan pada penelitian ini yang terdapat pada Tabel 3.2.

Tabel 3. 2 Spesifikasi Kebutuhan Perangkat Keras

No	Komponen	Spesifikasi
1.	<i>Processor</i>	Intel Core i5-9300H
2.	<i>Memory</i>	8 GB RAM DDR3
3.	<i>SSD</i>	256 GB
4.	<i>Graphic Card</i>	NVIDIA GeForce GTX 1650 Ti
5.	<i>Mouse</i>	Optical
6.	<i>Keyboard</i>	QWERTY

3.2.3 Spesifikasi Kebutuhan Perangkat Lunak

Berikut ini merupakan spesifikasi perangkat lunak yang dibutuhkan pada penelitian ini yang terdapat pada Tabel 3.3.

Tabel 3. 3 Spesifikasi Kebutuhan Perangkat Lunak

No	Kebutuhan	Keterangan	Fungsi
1.	Sistem Operasi	<i>Windows 11 64 bit</i>	Sistem operasi yang digunakan untuk menjalankan perangkat lunak
2.	Bahasa Pemrograman	Python	Untuk membangun model CNN
		Kotlin	Untuk membangun aplikasi android
3.	Aplikasi	Google Colab	Untuk membangun dan <i>training</i> model CNN
		Android Studio	Membangun aplikasi
		Figma	Membuat desain aplikasi
		Draw.io	Untuk membuat desain UML
		Github	Mendokumentasikan program kedalam <i>repository</i> git

3.2.4 Spesifikasi Perangkat Android

Perangkat android yang digunakan dalam pengembangan aplikasi perangkat bergerak ini, memiliki spesifikasi yang dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Spesifikasi Perangkat Android

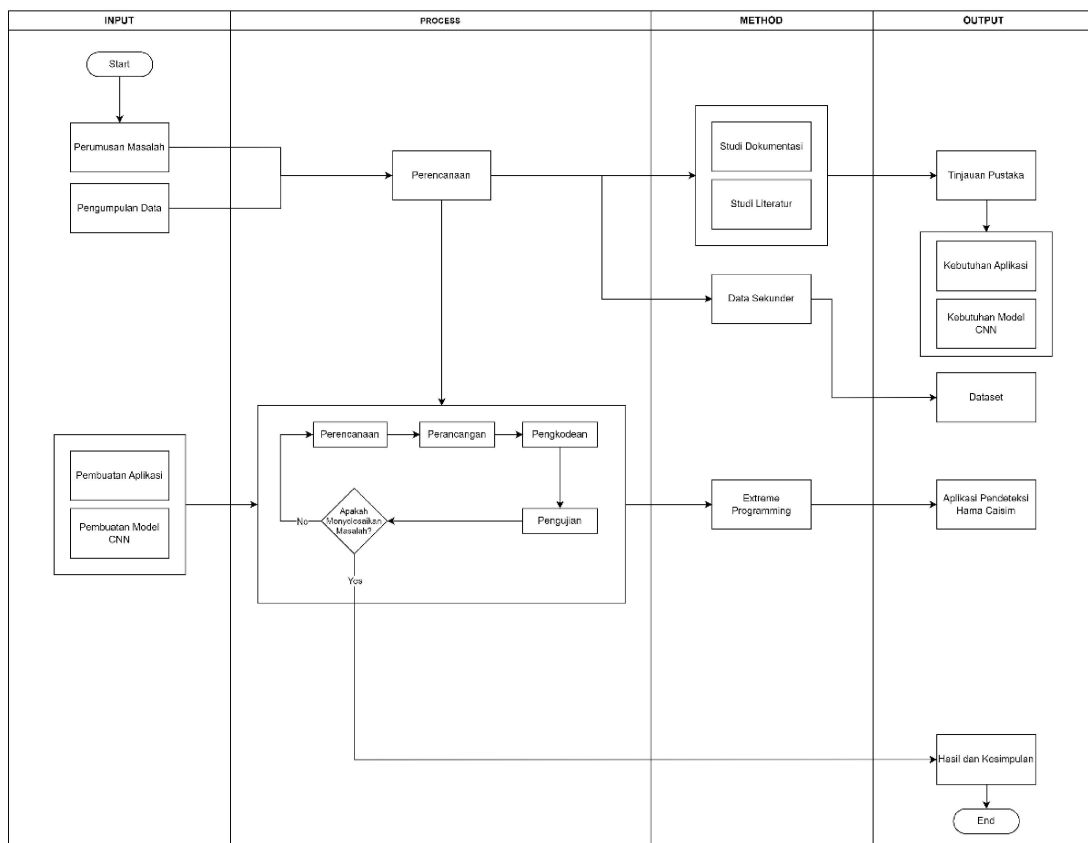
No	Komponen	Spesifikasi
1.	Sistem Operasi	Android 11 (<i>Red Velvet Cake</i>)
2.	<i>Memory</i>	4 GB RAM
		64 GB <i>Internal</i>

No	Komponen	Spesifikasi
3.	<i>Display</i>	IPS LCD, 720 x 1560 <i>pixels</i> , 6.5 <i>inches</i>
4.	<i>Processor</i>	<i>Octa-core</i> 1.8 GHz <i>Cortex-A53</i>

3.3 Diagram Alir Penelitian

Dalam penelitian ini, tahapan pengembangan dapat diilustrasikan melalui diagram alir yang mengaplikasikan metode *extreme programming*. Metode *extreme programming* adalah suatu pendekatan pengembangan perangkat lunak yang mengikuti urutan linier, di mana setiap tahapan dilakukan secara berurutan dan bergantung pada keberhasilan tahapan sebelumnya.

Berikut ini adalah ilustrasi diagram alir yang menggambarkan metode *extreme programming* :



Gambar 3. 1 Diagram Alir Penelitian

Gambar 3.1 adalah gambaran secara keseluruhan tahapan penelitian yang akan dilakukan.

3.3.1 Perencanaan

Pada tahap analisis kebutuhan akan membahas mengenai perumusan masalah dan studi literatur.

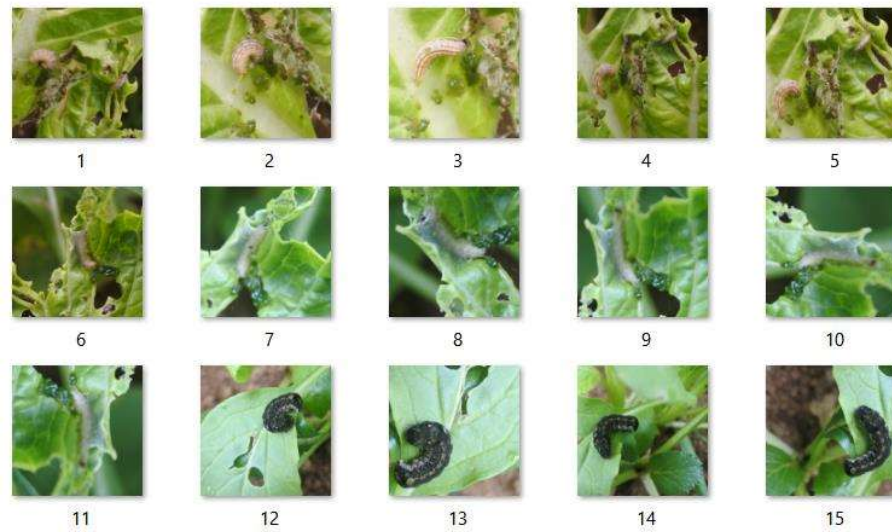
a. Studi Literatur

Pada tahapan ini dilakukan untuk melakukan peninjauan ulang mengenai penelitian terdahulu yang dijadikan sebagai acuan untuk melakukan pengembangan dari penelitian yang telah dilakukan sebelumnya dari hasil dan performa yang dihasilkan dan solusi yang dihasilkan nantinya tidak sama.

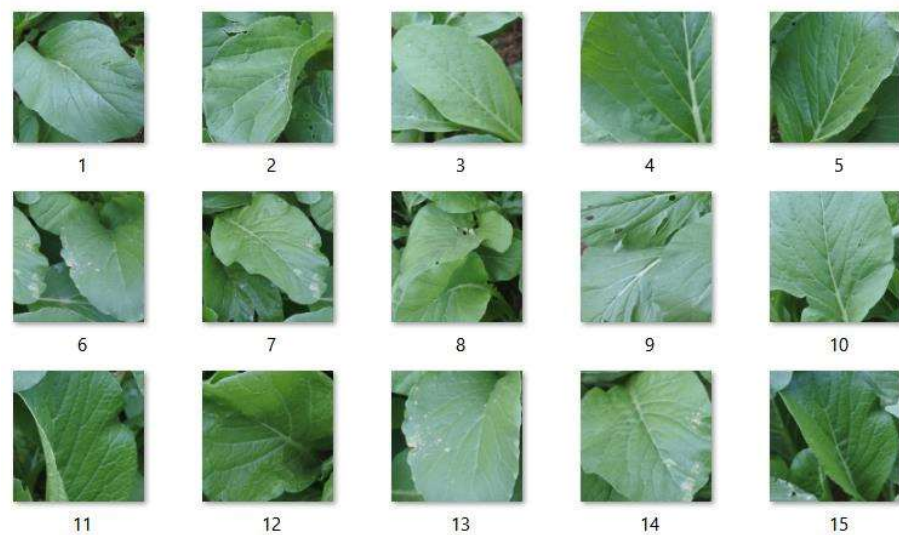
b. Studi Dokumentasi

Pada tahapan ini dilakukan untuk mengumpulkan *dataset* yang akan digunakan pada penelitian ini. *Dataset* sekunder yang telah dikumpulkan akan dibagi menjadi 3 bagian, yaitu data *train* yang akan digunakan untuk melatih model, data *validation* untuk mengevaluasi performa model selama proses *training*, dan data *testing* untuk mengevaluasi performa model menggunakan data yang belum pernah dilihat sebelumnya oleh model, sehingga dapat terlihat apakah model dapat menangani data baru dengan baik atau tidak.

Total jumlah *dataset* sekunder yang ada pada *dataset caisim* adalah 1000, yang dimana dari 1000 data tersebut memiliki 500 gambar sayur sawi hijau dengan hama dan 500 gambar sayur sawi hijau tanpa hama. Gambar 3.2 dan Gambar 3.3 merupakan masing-masing contoh gambar dari label sawi hijau dengan hama dan sawi hijau tanpa hama dari *dataset* yang telah dikumpulkan. Jenis hama pada *dataset* penelitian ini adalah jenis ulat.



Gambar 3. 2 Data Sawi Hijau Ada Hama



Gambar 3. 3 Data Sawi Hijau Tanpa Hama

3.3.2 Perancangan

Pada tahap desain akan membahas mengenai membuat desain diagram UML dan membuat desain aplikasi android.

a. Membuat Desain Diagram UML

Pada tahap ini akan dilakukan desain dari *use case* dan juga activity diagram berdasarkan *use case* yang telah dibuat sebelumnya.

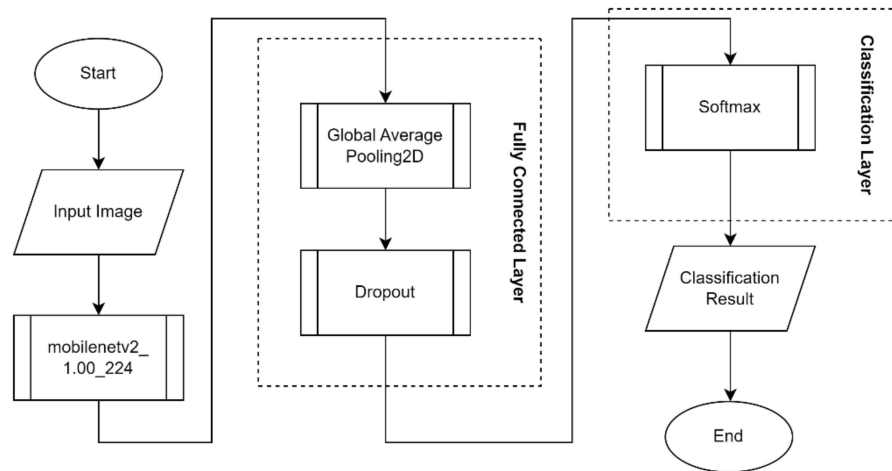
b. Membuat Desain Aplikasi Android

Pada tahap ini akan dilakukan desain dari antarmuka dan juga beberapa fungsional dari aplikasi android.

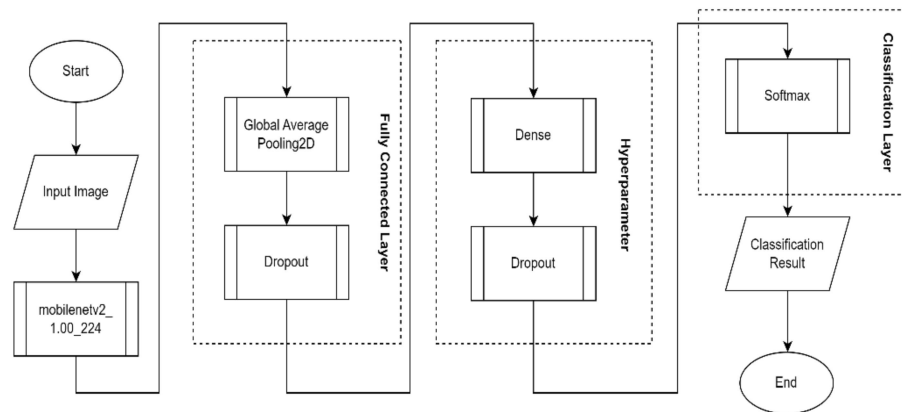
c. Membuat Perancangan Arsitektur Model

Pada tahap ini akan dilakukan perancangan arsitektur untuk model CNN arsitektur *MobileNetV2* dengan metode *transfer learning* yang akan dilatih menggunakan data yang telah di proses. Metode *transfer learning* adalah metode untuk melakukan *transfer* pengetahuan dari model yang telah dilatih sebelumnya menggunakan dataset besar dan juga banya kelas. Pada studi kasus ini terdapat dua label yang digunakan yaitu ada hama ulat dan tidak ada hama ulat, maka perlu dilakukan penyesuaian pada *output layer* dari arsitektur yang digunakan menjadi dua kelas dengan activation *softmax*. Kemudian dilakukan pula teknik pendekatan yaitu *freezing layer* pada model *transfer learning* yang akan digunakan agar bobot dari model tidak berubah saat proses *training* dilakukan, hal ini juga dapat membantu agar model tidak kehilangan pengetahuan yang telah diperoleh sebelumnya, serta model dapat digunakan sebagai *feature extraction global average pooling 2D*. Model yang telah dibuat kemudian akan di *compile* menggunakan optimasi *adam*, *learning rate* 0.001, dan *loss sepase_categorical_crossentropy*.

Pada proses rancangan akan diterapkan pula metode *hyperparameter*. *Hyperparameter* yang dapat dilakukan seperti *optimizer*, *learning rate*, *epoch*, *batch size*, *dropout*, *layer model* dan augmentasi untuk pengoptimalan model [40], [51], [52]. Dapat dilihat pada Gambar 3.4 dan 3.5 masing-masing merupakan gambaran alur rancangan dari model CNN arsitektur *MobileNetV2* dengan metode *transfer learning* dan teknik *freezing layer* tanpa *hyperparameter* dan dengan *hyperparameter* yang akan dilakukan untuk melakukan deteksi keberadaan hama ulat pada daun sawi hijau.



Gambar 3. 4 CNN arsitektur *MobileNetV2* tanpa *hyperparameter*



Gambar 3. 5 CNN arsitektur *MobileNetV2* dengan *hyperparameter*

Gambar diatas adalah gambaran dari alur rancangan CNN arsitektur *MobileNetV2* yang akan dibuat. Apabila model CNN arsitektur *MobileNetV2* yang telah dibuat memiliki performa yang kurang baik, maka dengan melakukan teknik *hyperparameter* dapat meningkatkan performa dari model sehingga menjadi lebih optimal [51].

3.3.3 Pengkodean

Pada tahap implementasi akan membahas mengenai pengkodean aplikasi android, membuat rancangan model CNN arsitektur *MobileNetV2* dengan metode *transfer learning* serta teknik *hyperparameter*, pelatihan dan pengujian model, *deployment* model ke dalam aplikasi android.

a. *Data Preprocessing*

Data *preprocessing* adalah tahap dimana data yang telah dikumpulkan kemudian akan diolah lebih lanjut sehingga dapat diterima dengan baik oleh komputer, kemudian siap digunakan untuk melakukan pengembangan model *machine learning*. Pada penelitian ini *preprocessing* yang dilakukan adalah sebagai berikut :

1. *Decode* data. Pada proses *decode*, seluruh gambar yang ada dalam *dataset* akan diubah dalam format tertentu menjadi bentuk yang dapat diolah oleh komputer.
2. *Resize* data. Pada proses *resize* seluruh ukuran data akan diubah menjadi 224x224 pixel dari ukuran aslinya 512x512 pixel. Normalisasi data ini dilakukan dengan menggunakan *tf.image.resize* dari *TensorFlow*.
3. *Rescale* data. Pada proses *rescale* data akan di normalisasi yang dimana dilakukan penskalaan citra pada gambar sawi hijau dari [0, 255] menjadi [0, 1]. Normalisasi data ini dilakukan dengan menggunakan *tf.image.convert_image_dtype* dari *TensorFlow*.
4. *Split* data. Pada proses *split* data yang telah dikumpulkan akan dibagi menjadi 3 bagian, yaitu data *training*, *validation*, dan *testing*.
5. *Augmentasi* data. Pada proses *augmentasi* data akan ditambah, yang pada awalnya hanya sebanyak 1000, pada proses ini akan ditambah kembali dengan total 1000 variasi data gambar baru, yaitu 500 data sawi hijau dengan hama ulat dan 500 data sawi hijau tanpa hama ulat, sehingga total data yang digunakan pada penelitian ini adalah 2000 data. *Augmentasi* data yang dilakukan pada penelitian ini akan menggunakan *tf.image* dari *TensorFlow*. Proses *augmentasi* yang digunakan pada penelitian ini adalah *random_flip_left_right*, *random_flip_up_down*, *rot90*, *random_brightness*, *random_contrast*, *random_hue*, dan *random_saturation*.

Proses tersebut dilakukan untuk meningkatkan akurasi serta mengurangi *loss* pada saat pelatihan model [40].

b. Modelling

Pada tahap ini model yang telah dirancang dengan baik menggunakan metode *transfer learning* dan teknik *freezing layer* serta *hyperparameter* akan dilakukan proses *training* menggunakan total 2000 data yang telah selesai di proses sebelumnya.

c. Pengujian Model

Pada tahap ini model CNN yang telah selesai dilatih kemudian diuji untuk bahan evaluasi menggunakan *confusion matrix* untuk melihat gambaran kinerja dari model dalam melakukan klasifikasi. Melalui teknik ini, akan dapat pula menghitung metrik seperti *accuracy*, *recall*, *precision*, dan *f1-score* untuk menganalisis kinerja model dengan lebih baik.

d. Pengkodean Aplikasi Android

Pada tahap ini *software* yang akan dirancang menggunakan bahasa pemrograman Kotlin serta menggunakan *tools* IDE Android Studio untuk pengembangan aplikasi sistem operasi Android.

e. Deployment

Pada tahap ini model yang telah melewati tahapan evaluasi akan diintegrasikan ke dalam aplikasi android yang telah dirancang dan dibangun sebelumnya menggunakan *TensorFlow Lite*.

3.3.4 Pengujian

Pada tahap ini akan dilakukan pengujian untuk memastikan bahwa setiap fungsionalitas dalam sistem berjalan dengan baik. Pengujian ini menggunakan metode *blackbox testing*, di mana pengguna dapat mencoba melihat hasil klasifikasi sistem dengan memberikan *input* gambar daun sawi hijau, dan memastikan bahwa *output* nya sesuai dengan yang diharapkan. Pada tahap ini juga akan dilakukan *review* atau analisis lebih lanjut mengenai aplikasi yang telah diuji, apakah aplikasi tersebut sudah menyelesaikan masalah secara keseluruhan atau belum. Jika belum,

maka akan iterasi kembali yang dimulai dari perencanaan. Jika sudah, maka proses iterasi akan dihentikan, kemudian aplikasi akan di rilis.