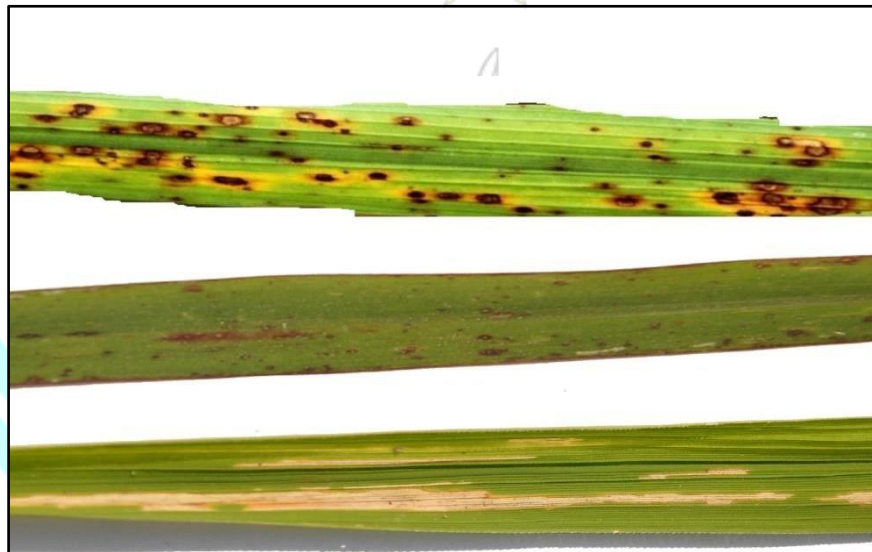


BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Dalam penelitian ini, obyek yang digunakan adalah citra dua dimensi dari tiga daun padi. Data tersebut kemudian diklasifikasikan menjadi tiga kategori, yaitu daun yang terinfeksi penyakit bintik coklat, daun busuk dan hawar daun. Perincian gambar ada pada gambar 3.1. Data citra tersebut diperoleh pribadi oleh penulis sebanyak 500 citra dengan menggunakan kamera handphone dengan kualitas kamera 12mp dan dari internet sebanyak 300 citra, total 800 citra.



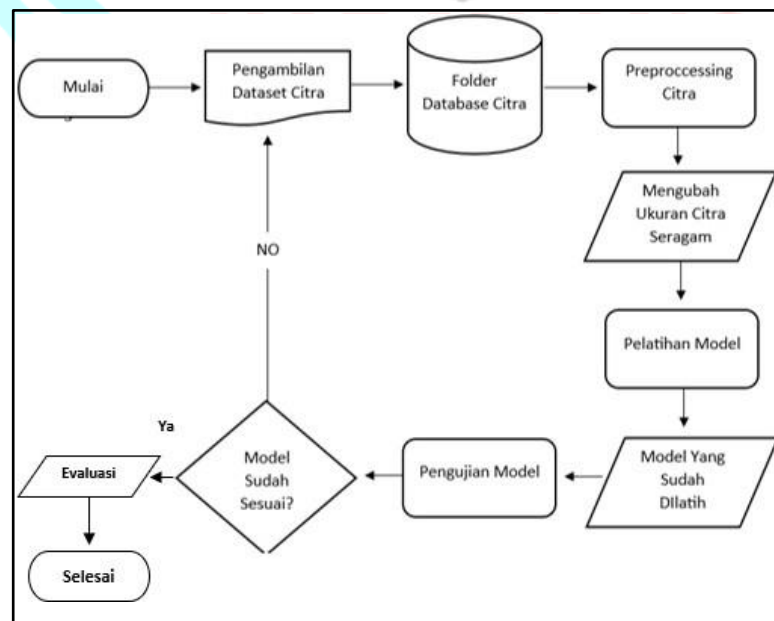
Gambar 3.1 Objek Penelitian

Lokasi penelitian ini berada di pesawahan milik Bapak Irwan di Desa Lemahduhur, Kecamatan Tempuran.

3.2. Prosedur Penelitian

Tiga tahapan pada prosedur penelitian ini dilakukan, tahapan tersebut meliputi Tiga tahapan pada prosedur penelitian ini dilakukan, yaitu persiapan data, pelatihan model, dan pengujian model yang telah dilatih. Tahapan pertama, persiapan data, mencakup pengumpulan citra atau gambar daun padi yang difokuskan pada proses persiapan dataset. Dataset ini terdiri dari citra daun padi yang telah dikelompokkan sesuai dengan kategori penyakit, seperti Hawar Daun, Blast, dan Bintik Coklat. Selanjutnya, dataset yang telah dikumpulkan dilakukan preprocessing, yang mencakup normalisasi ukuran gambar, augmentasi data untuk meningkatkan variasi dataset, dan konversi ke format yang kompatibel dengan model.

Tahapan kedua adalah pelatihan model, di mana citra yang telah melalui preprocessing digunakan untuk melatih model menggunakan arsitektur MobileNet v2 pada algoritma Convolutional Neural Network (CNN). Proses pelatihan ini melibatkan pembagian data menjadi data pelatihan dan data validasi dengan rasio tertentu, di mana parameter model dioptimalkan untuk mendapatkan kinerja terbaik. Tahapan terakhir adalah pengujian model, yaitu proses evaluasi performa model yang telah dilatih menggunakan data uji yang terpisah. Pada tahap ini, akurasi model dalam mendeteksi jenis penyakit pada daun padi diukur dan dianalisis.



Gambar 3.2 Diagram Alir Prosedur Penelitian

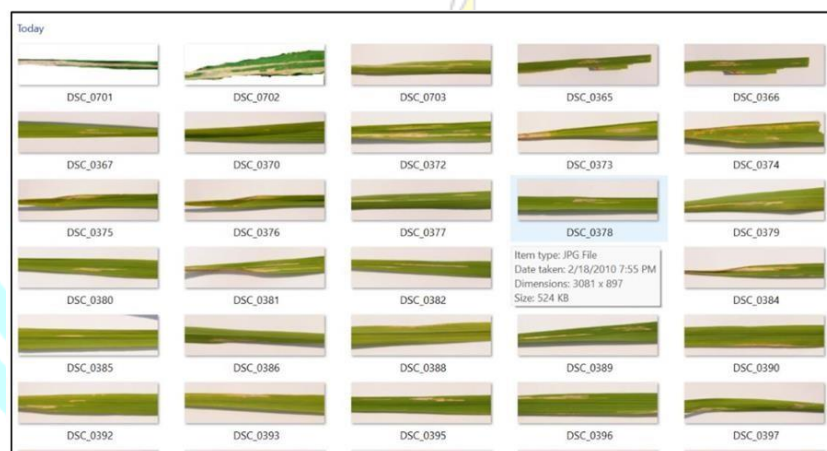
Dapat dilihat diagram alir pada Gambar 3.2, setelah dilakukan preprocessing data lalu dilatih dan mendapatkan sebuah model deteksi yang diperlukan pada penelitian ini. Fungsi dari model deteksi tersebut menandakan klasifikasi terhadap Jenis Penyakit pada daun padi. Untuk mengetahui jenis penyakit pada daun padi. Pengumpulan data dan *Preprocessing* Gambar.

3.2.1 Pengambilan Dataset Citra

Pada tahap ini, dilakukan pengumpulan gambar penyakit pada daun padi, yaitu penyakit hawar daun, daun busuk, dan bintik coklat.

1. Penyakit Hawar Daun

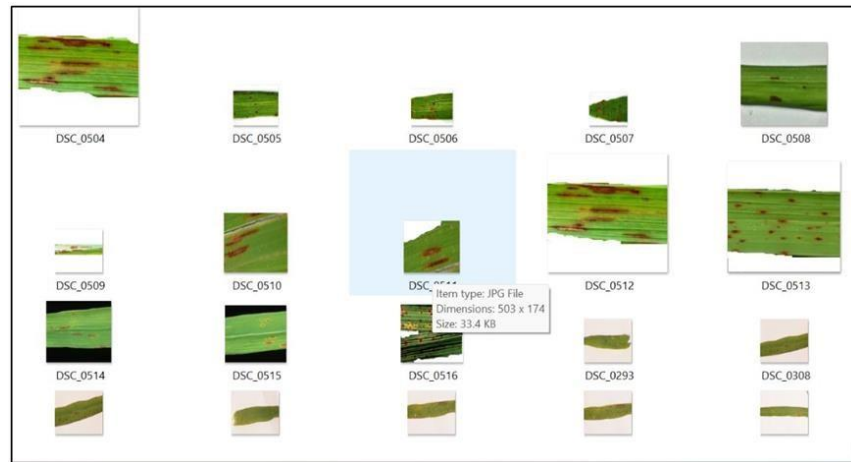
Gambar penyakit hawar daun dikumpulkan sebanyak 266 gambar. Penyakit ini menyebabkan kerusakan signifikan pada daun padi, yang dapat mengakibatkan penurunan hasil panen. Contoh gambar penyakit hawar daun dapat dilihat pada Gambar 3.3.



Gambar 3.3 Dataset Penyakit Hawar Daun

2. Penyakit Daun Busuk

Sebanyak 266 gambar penyakit daun busuk juga dikumpulkan. Penyakit ini dikenal sangat merusak, bahkan berpotensi menyebabkan kehancuran panen. Contoh gambar penyakit daun busuk dapat dilihat pada Gambar 3.4.



Gambar 3.4 Dataset Penyakit Belast

3. Penyakit Bintik Coklat

Penyakit bintik coklat dikumpulkan sebanyak 267 gambar. Penyakit ini menyebabkan bercak coklat pada daun padi, yang dapat memengaruhi kualitas dan hasil panen. Contoh gambar penyakit bintik coklat dapat dilihat pada Gambar 3.5.



Gambar 3.5 Dataset penyakit bintik coklat

3.2.2 Preprocessing Citra

Tahap preprocessing dilakukan untuk mempersiapkan gambar yang telah dikumpulkan menjadi dataset yang siap digunakan untuk pelatihan model. Pada langkah ini, gambar disimpan dalam folder yang terstruktur sesuai dengan kategori penyakit. Selanjutnya, setiap gambar diproses menggunakan library TensorFlow untuk memastikan konsistensi dalam input model.

3.2.3 Mengubah Ukuran Citra Seragam

Setelah gambar dikumpulkan, dilakukan proses resizing untuk mengubah ukuran gambar menjadi 224x224 piksel. Langkah ini bertujuan untuk menjaga keseragaman ukuran gambar dan meningkatkan konsistensi saat digunakan sebagai input pada model Convolutional Neural Network (CNN). Dengan ukuran seragam, proses pelatihan dapat berjalan lebih efisien, serta meningkatkan akurasi model.

3.2.4 Pelatihan Model

Gambar yang telah melalui preprocessing digunakan untuk melatih model CNN menggunakan arsitektur MobileNet v2. Data dibagi menjadi dua bagian, yaitu data pelatihan dan data validasi, untuk memastikan model mampu belajar dengan baik dan tidak mengalami overfitting. Pelatihan dilakukan dengan beberapa parameter, seperti jumlah epoch, batch size, dan learning rate yang telah disesuaikan untuk mencapai hasil optimal.

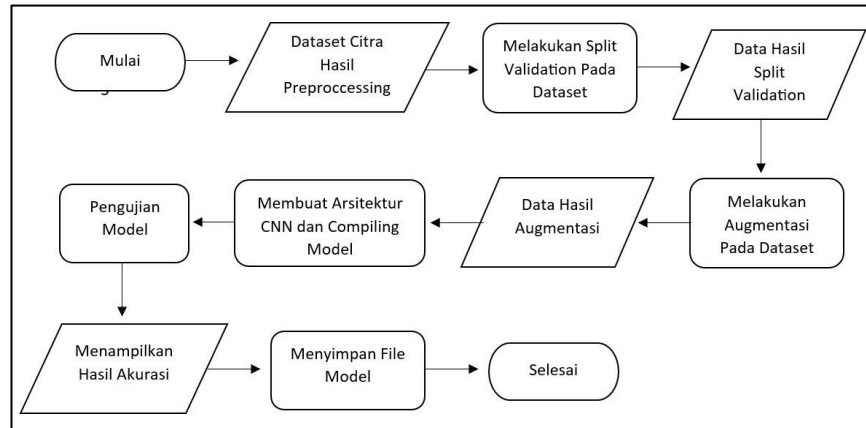
3.2.5 Pengujian Model

Tahap pengujian dilakukan untuk mengevaluasi kinerja model menggunakan data uji yang terpisah dari data pelatihan dan validasi. Pengujian ini bertujuan untuk mengetahui kemampuan model dalam mengenali penyakit pada daun padi berdasarkan gambar yang tidak termasuk dalam data pelatihan.

3.2.6 Evaluasi

Hasil pelatihan dan pengujian model dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score untuk mengetahui performa keseluruhan model dalam mengklasifikasikan penyakit pada daun padi. Evaluasi ini memberikan gambaran sejauh mana model mampu mendeteksi jenis penyakit secara akurat, yang dapat membantu petani dalam meminimalkan kerugian akibat penyakit padi.

3.3. Pelatihan Model

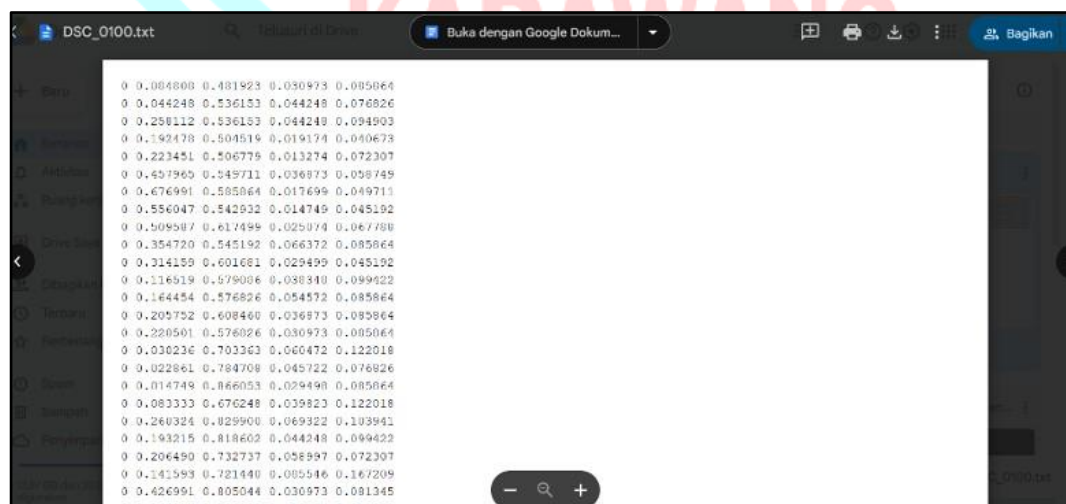


Gambar 3.6 Diagram Alir Pelatihan Model

Dikarenakan model deteksi penyakit daun padi ini merupakan bagian dari *supervised learning*, membuat deteksi memerlukan kumpulan data atau dataset dari suatu citra yang berlabel sebagai data pelatihan untuk mendeteksi objek pada citra masukan. Ada tiga tahapan pada proses pengembangan model deteksi objek, tahapan tersebut meliputi dataset citra, pelatihan model deteksi objek dan menampilkan hasil akurasi. *Convolutional Neural Network* digunakan, fungsinya sebagai algoritma pendeteksian objek. Berdasarkan pada Gambar 3.6 memiliki penjelasan sebagai berikut:

1. Dataset Preprocessing

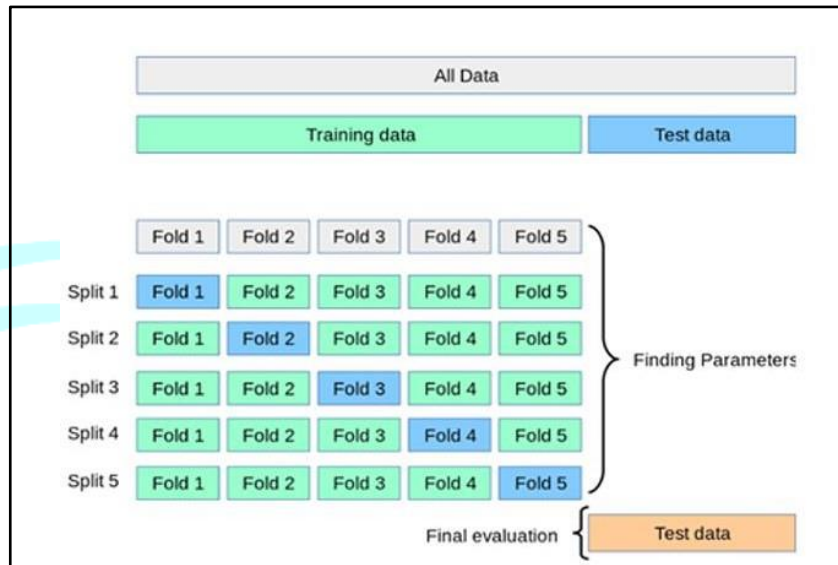
Pada proses ini yaitu memuat data set yang sebelumnya telah dilakukan tahap *preprocessing* (mengubah ukuran citra dan mengkonversi ke bentuk *array*).



Gambar 3.7 Permodelan Dataset Preprocessing

2. Split Validasi

Dataset pada gambar dilakukan *split* validasi menggunakan *library* dari *skicit-learn*. Tujuannya agar model yang dihasilkan memberikan hasil maksimal dan akurat.

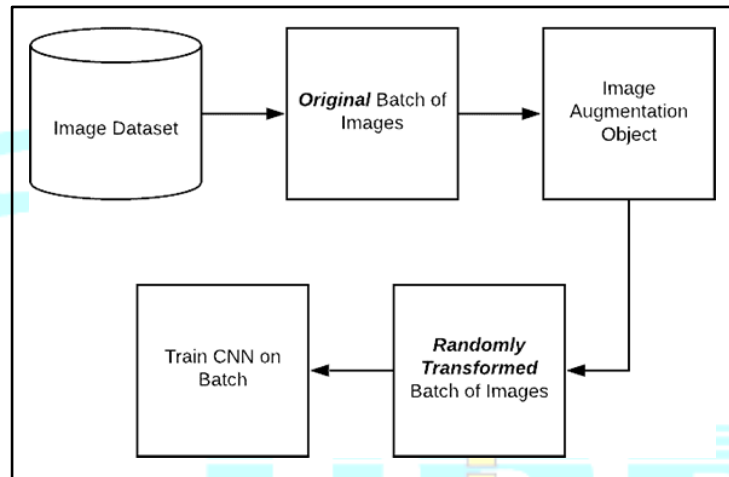


Gambar 3.8 Permodelan Split Validasi menggunakan skicit-learn



3. Data Augmentasi

Augmentasi dilakukan agar citra dapat dimanipulasi sehingga memperbanyak citra pada data set. Proses augmentasi ini dilakukan menggunakan *library* ImageDataGenerator dari *Tensorflow* Keras. Augmentasi yang dilakukan yaitu *zoom*, rotasi, menggeser, memotong dan membalikan citra secara acak dari citra yang ada pada data set.



Gambar 3.9 Permodelan Data Augmentasi menggunakan library ImageDataGenerator dari Tensorflow

4. Membuat arsitektur CNN dan Compiling Model

Dalam membuat arsitektur CNN terdapat dua tahapan yaitu *Feature learning* dan *Classification*. Tahapan tersebut tentunya memiliki langkah-langkah yang harus dilakukan, pada *Feature learning* terdapat *input* yaitu data set hasil *preprocessing*. Kemudian, dilanjutkan dengan *convolution* untuk melakukan proses *stride* yang didalamnya memiliki operasi matematika dengan hasil dimasukan pada peta fitur atau *feature map*. Selanjutnya fungsi aktivasi dilakukan agar menghapus nilai negatif yang terdapat pada data, lapisan penyatuan atau *pooling layer* digunakan untuk mengurangi komputasi pada proses pelatihan.

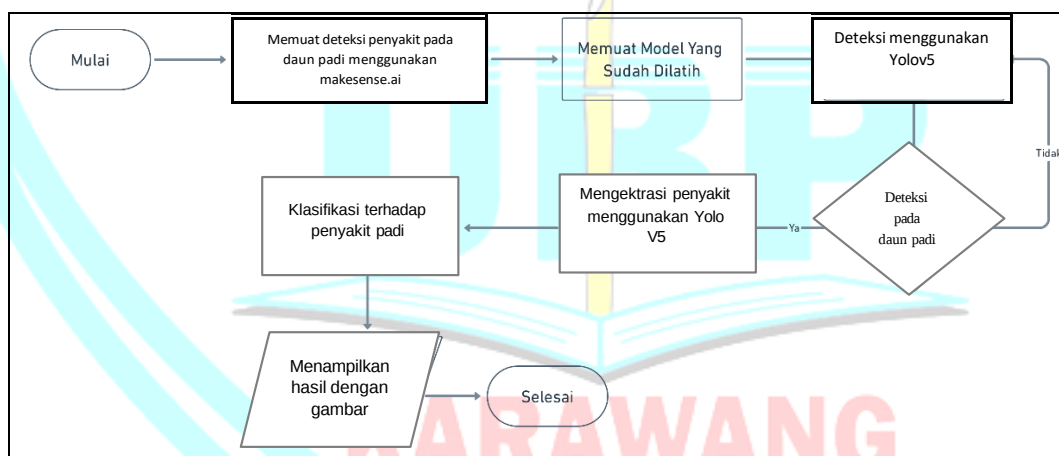
Tahap *Classification* berfungsi untuk hasil dari proses *feature learning* akan digunakan untuk proses klasifikasi berdasarkan subclass yang sudah ditentukan. Langkah pada tahap ini yaitu *flatten* yang dilakukan untuk mengubah data multi dimensi *array* pada *feature map*, menjadi *vector* agar dapat digunakan sebagai *input* pada dari *fully connected layer*. Kemudian, lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron selanjutnya. Fungsi

aktivasi yang digunakan tidak RELU tetapi menjadi *softmax* (mencari nilai maksimum dalam suatu klasifikasi).

5. Training Model

Tahap ini data pada hasil *processing* akan diinputkan ke dalam arsitektur tersebut untuk memulai proses pelatihan. Tujuan dari proses ini agar arsitektur CNN dapat mengenali *dataset* untuk membentuk model berdasarkan pelatihan. Beberapa parameter ditentukan seperti *epoch* dan *learning rate*. *Epoch* tersebut berguna untuk melakukan banyaknya perulangan yang pada proses pelatihan. Adapun *learning rate* berfungsi agar parameter yang akan digunakan untuk mengupdate bobot pada proses pelatihan di setiap kali iterasi (*epoch*). Pada tahap ini juga metode data augmentasi dikombinasikan. Bertujuan untuk memperbanyak data set dan meningkatkan hasil akurasi.

3.3.1 Pengujian Model



Gambar 3.10 Diagram Alir Pengujian Model

Pada Gambar 3.10, langkah pertama adalah menginstal semua komponen yang diperlukan seperti *mengunggah caffemodel* yang berfungsi sebagai pendeteksi daun dan langkah terakhir adalah mengunggah *fail* model yang kita latih sebelumnya. Selama fase pengujian ini, dapat memilih untuk mendeteksi melalui gambar atau video. Kemudian *caffemodel* akan mencari daun di sumbernya, jika ada daun, kurangi *Region of Interest* (ROI) untuk setiap daun yang terdeteksi.

3.3.2 Evaluasi

Tahap evaluasi dilakukan untuk menilai performa model yang telah dilatih menggunakan metrik tertentu, antara lain:

- a) **Akurasi:** Persentase prediksi model yang benar terhadap total data uji.
- b) **Presisi:** Proporsi prediksi positif yang benar dibandingkan seluruh prediksi positif.
- c) **Recall (Sensitivitas):** Kemampuan model mendeteksi seluruh kasus positif.
- d) **F1-Score:** Harmoni antara presisi dan recall untuk memberikan gambaran menyeluruh terhadap performa model.

Evaluasi juga dapat mencakup analisis visual, seperti **Confusion Matrix**, yang menampilkan jumlah prediksi benar dan salah untuk setiap kelas. Berdasarkan hasil evaluasi, model dapat ditingkatkan lebih lanjut dengan mengatur ulang parameter, meningkatkan dataset, atau memperbaiki preprocessing data.

