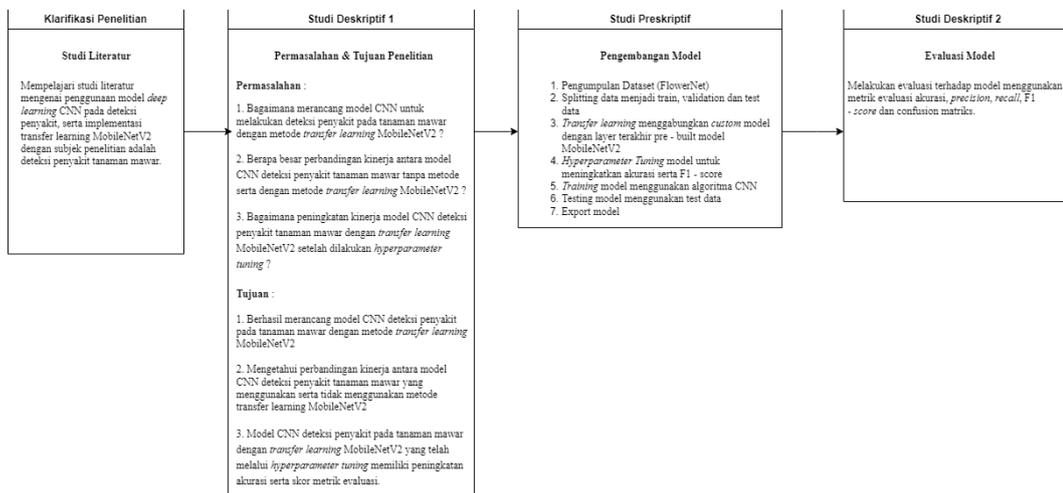


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian merupakan kerangka kerja yang berfungsi sebagai acuan untuk melaksanakan penelitian. Metode yang dipakai dalam penelitian ini adalah metode penelitian *Design Research Methodology* (DRM). Tahapan dalam DRM terdiri dari : klarifikasi penelitian, studi deskriptif 1, studi preskriptif serta studi deskriptif 2 (Blessing dan Chakrabarti, 2009). Penulis memilih DRM karena metode penelitian ini sangat cocok digunakan dalam pengembangan suatu perangkat lunak. Berdasarkan metode penelitian DRM, Penulis merancang skema penelitian seperti pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

3.1.1 Klarifikasi Penelitian

Pada tahapan ini dilakukan studi literatur untuk mengidentifikasi permasalahan penelitian terkait pendeteksian penyakit pada tanaman mawar menggunakan model CNN. Berbagai buku, artikel jurnal serta prosiding digunakan untuk memahami lebih dalam tentang penyakit pada tanaman mawar serta referensi model CNN untuk klasifikasi gambar dari penelitian terdahulu.

3.1.2 Studi Deskriptif 1

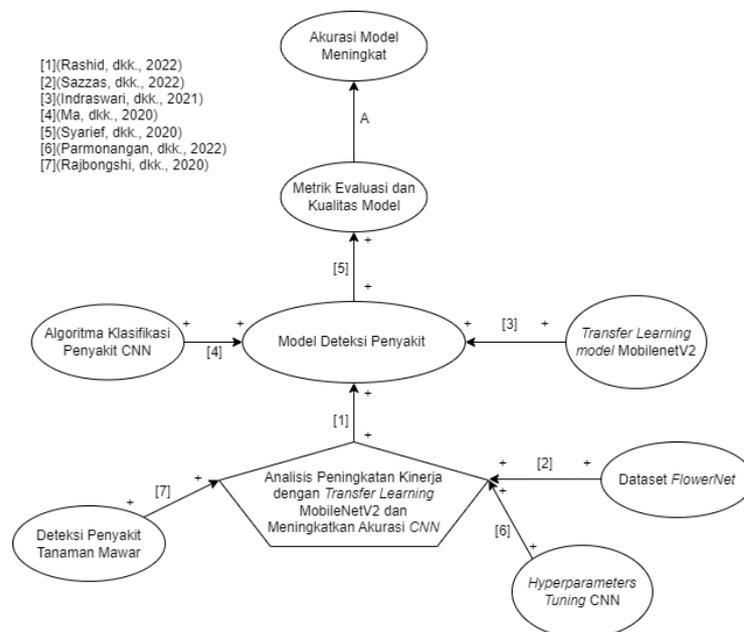
Tahapan berikutnya adalah menjabarkan permasalahan serta tujuan penelitian berdasarkan studi literatur yang telah dilakukan pada tahap sebelumnya. Hasil dari tahapan ini akan menjadi landasan dalam membangun solusi dari permasalahan penelitian ini.

3.1.3 Studi Preskriptif

Selanjutnya pada tahapan ini, langkah – langkah dalam pembuatan model untuk deteksi penyakit pada tanaman mawar diterapkan. Langkah – langkah tersebut merujuk pada penelitian terdahulu sebagai referensi dengan beberapa penyesuaian, kemudian model dievaluasi menggunakan beberapa metrik seperti F1 – score dan Confusion Matrix.

3.1.3.1 Model Dampak

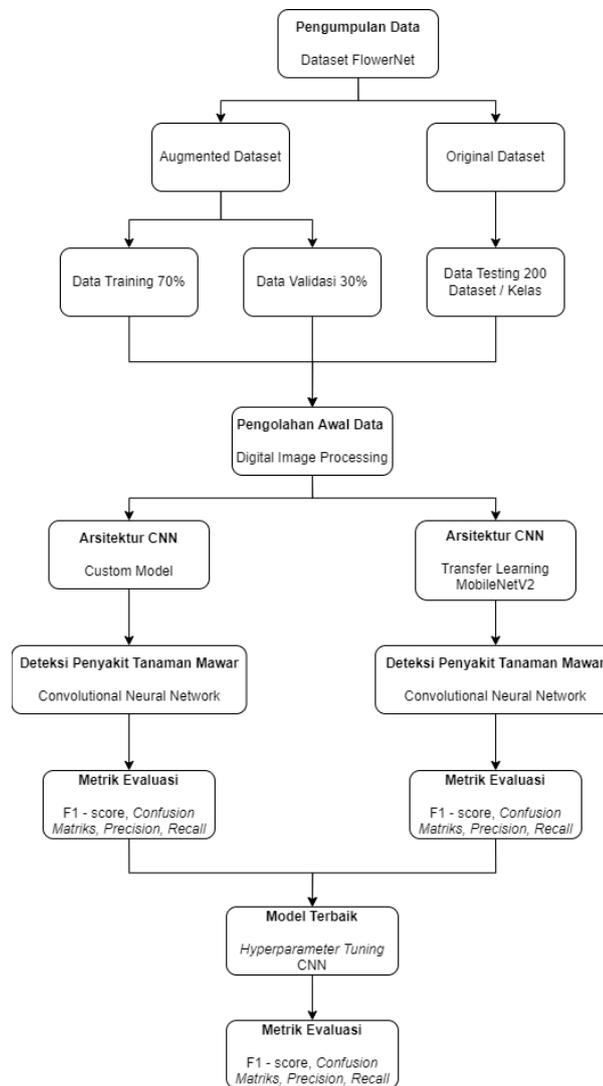
Gambar 3.2 merupakan model dampak pada penelitian ini. Model dampak adalah gambaran yang dibuat oleh penulis, yang dimaksudkan sebagai acuan untuk tujuan yang ingin dicapai dan perubahan dampak yang didukung oleh hasil penelitian sebelumnya.



Gambar 3.2 Model Dampak

3.1.3.2 Pengembangan Model Deteksi Penyakit Pada Tanaman Mawar

Gambar menggambarkan tahapan pengembangan model deteksi penyakit, dimana fitur utama dari model ini adalah menggunakan metode *transfer learning* menggunakan *pre – built* model MobileNetV2 pada jenis algoritma klasifikasi berupa CNN. Model yang menggunakan *transfer learning* dan model yang tidak menggunakan *transfer learning* kemudian dibandingkan dengan hipotesis bahwa model yang menggunakan transfer learning akan memperoleh kinerja lebih optimal dibandingkan model yang tidak memakai transfer learning.



Gambar 3.3 Tahapan Pengembangan Model Deteksi Penyakit Pada Tanaman Mawar

3.1.3.2.1 Pengumpulan Data

Data yang digunakan adalah *dataset* FlowerNet yang berasal dari website Mendeley Data yang dapat diakses pada halaman <https://data.mendeley.com/datasets/7z67nyc57w/2>. FlowerNet berisi 5259 gambar yang dibagi menjadi 2 jenis *dataset* : original dataset berisi 917 foto, serta augmented dataset berisi 4342 foto yang diambil dari Village of Roses (Golap Gram) yang berasal dari daerah Sadullapur Dhaka, Bangladesh. Masing – masing dataset dibagi menjadi 3 kelas data yang terdiri dari : *downy mildew* (bulai), *fresh leaf* (sehat), serta *black spot* (bintik hitam). Penulis memilih *dataset* FlowerNet karena *dataset* ini menggunakan foto yang diambil dari lapangan langsung, sehingga Penulis dapat membuat model yang lebih relevan (Sazzad, Rajbongshi, Shakil, Akter, dan Kaiser, 2022). Dataset ini juga telah memiliki data yang telah dilakukan augmentasi sehingga dapat menghasilkan akurasi yang lebih baik.

3.1.3.2.2 Pra Pemrosesan – Data

Proses pertama yang dilakukan dalam pra – pemrosesan data adalah dengan membagi dataset FlowerNet menjadi data latih dan data validasi. Presentase perbandingan kedua dataset tersebut adalah 70% untuk dataset latih, serta 30% untuk dataset validasi. Dataset yang telah dibagi kemudian ditambahkan *Digital Image Processing* (DIP) yang terdiri dari mengubah *range* piksel gambar dari 0 – 255 menjadi 0 - 1 agar lebih mudah diproses menggunakan *rescale*, mengubah kecerahan dalam gambar menggunakan *channel_shift_range* dan *brightness_range*, menambahkan rotasi menggunakan *rotation_range*, membuat gambar dapat dibolak – balik vertikal menggunakan *horizontal_flip*, mengubah distorsi gambar baik dari sumbu x maupun sumbu y menggunakan *shear_range*, serta mengisi tempat yang kosong karena gambar disebabkan oleh penyamaan dimensi gambar menggunakan *fill_mode*.

3.1.3.2.3 Arsitektur CNN

Selanjutnya adalah membuat 2 arsitektur model yang berbeda. Yaitu model *custom* yang dibuat tanpa menggunakan *transfer learning* menggunakan MobileNetV2, serta model yang dibuat dengan menambahkan *transfer learning*

menggunakan MobileNetV2. *Transfer learning* dilakukan dengan menambahkan *library tensorflow keras* untuk menggunakan *pra - processed* model MobileNetV2, kemudian dilanjutkan dengan mengambil *layer* terakhir dari MobileNetV2 sebelum *layer* konvolusi, terakhir *layer* yang telah diambil ditambahkan kedalam *custom* model CNN yang telah dibuat sebelumnya.

3.1.3.2.4 Deteksi Penyakit Tanaman Mawar

Proses selanjutnya, kedua jenis model yang telah dibuat sebelumnya kemudian dilatih menggunakan algoritma CNN untuk klasifikasi gambar pada *dataset* FlowerNet. Parameter pelatihan yang digunakan diantaranya implementasi fungsi optimisasi adam dengan besar parameter 0.001 serta 32 epoch. Pelatihan dijalankan pada google colab dengan tambahan TPU *accelerator*.

3.1.4 Studi Deskriptif 2

Terakhir, Pada tahap ini model yang telah dikembangkan kemudian dievaluasi untuk kemudian dibuat kesimpulan dari penelitian ini.

3.1.4.1 Evaluasi Model Deteksi Penyakit Pada Tanaman Mawar

Pengujian model dilakukan dengan menggunakan test dataset yang telah disiapkan sebelumnya, kemudian dievaluasi menggunakan beberapa metrik evaluasi seperti *F1- score*, *confusion matrix*, *recall* dan *precision* menggunakan *library* Scikit-learn.

3.1.4.1.1 F1 – Score

F1-Score adalah suatu metrik evaluasi yang umum digunakan dalam bidang pengenalan pola dan pembelajaran mesin, terutama dalam konteks klasifikasi biner. F1-Score menggabungkan presisi (*precision*) dan *recall* (ingatan) untuk memberikan ukuran tunggal yang menggambarkan performa keseluruhan dari sebuah model klasifikasi.

Rumus dari F1 – Score adalah :

$$F1 = \frac{1}{\frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right)}$$

3.1.4.1.2 Accuracy

Akurasi adalah metrik yang digunakan untuk mengevaluasi kinerja model pembelajaran mesin dalam masalah klasifikasi. akurasi didefinisikan sebagai rasio jumlah prediksi yang benar dengan jumlah total prediksi yang dibuat. Nilai akurasi terbaik adalah 1, dan nilai terburuk adalah 0.

Rumus dari Akurasi adalah :

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

3.1.4.1.3 Recall

Recall adalah salah satu metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi secara benar sejumlah kasus positif dalam dataset. Recall juga dikenal sebagai "sensitivitas" atau "true positive rate" (tingkat positif benar).

Rumus dari Recall adalah :

$$Recall = \frac{TP}{TP + FN}$$

3.1.4.1.4 Precision

Precision (presisi) adalah salah satu metrik evaluasi yang mengukur sejauh mana model dapat mengidentifikasi dengan benar kasus positif dari semua kasus yang diklasifikasikan sebagai positif oleh model. Presisi juga dikenal sebagai "positive predictive value" (nilai prediktif positif).

Rumus dari precision adalah :

$$Precision = \frac{TP}{TP + FP}$$

3.1.4.1.5 Confusion Matriks

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi. Ini adalah salah satu metrik evaluasi untuk masalah klasifikasi pembelajaran mesin yang keluarannya dapat menjangkau dua kelas atau lebih.

Confusion matrix memiliki empat elemen:

- True Positive (TP): Jumlah instance yang diklasifikasikan dengan benar sebagai positif.
- True Negative (TN): Jumlah instance yang diklasifikasikan dengan benar sebagai negatif.
- False Positive (FP): Jumlah instance yang diklasifikasikan salah sebagai positif.
- False Negative (FN): Jumlah instance yang diklasifikasikan salah sebagai negatif.

Dengan menganalisis confusion matrix, kita dapat menilai sejauh mana model dapat membedakan antara kelas, dan apakah model memiliki kecenderungan tertentu dalam membuat kesalahan untuk kelas tertentu dengan cara memasukkan nilai TP, TN, FP dan FN

Berikut adalah penggambaran umum dari confusion matriks :

Tabel 3.1
Confusion Matrix

Observasi	Prediksi Positif	Prediksi Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

3.1.4.1.6 FLOPS

FLOPS (Floating Point Operations Per Second) adalah metrik evaluasi yang digunakan untuk mengukur seberapa kompleks suatu arsitektur *deep neural network* dengan melihat banyaknya perhitungan matematika didalam lapisan arsitektur tersebut. semakin kecil ukuran FLOPs maka semakin cepat proses komputasi dilakukan sehingga waktu klasifikasi menjadi lebih cepat (Zhao, Wang, Jia, dan Cui, 2022)

Rumus dari FLOPS berdasarkan lapisan *neural network* adalah :

Sr. No	Layer	The Equation to Calculate FLOPs
1	Convolution Layer	$2 \times \text{No. of Kernel} \times \text{Kernel's Shape} \times \text{Output Shape} \times \text{Repeat Count (if available)}$
2	Pooling Layer (Without stride)	$\text{Height} \times \text{Width} \times \text{Depth of an input Image}$
3	Pooling Layer (With stride)	$(\text{Height}/\text{Stride}) \times \text{Depth} \times (\text{Width}/\text{Stride}) \text{ of an input Image}$
4	Fully Connected Layer (FC Layer)	$2 \times \text{Input Size} \times \text{Output Size}$

Output shape of Convolution Layer can be determined using following equation, Output Shape = (Input Shape - Kernel Shape) + 1.

Gambar 3.4 Rumus FLOPS

3.2 Alat dan Bahan Penelitian

Berikut ini alat-alat yang digunakan untuk menunjang proses penelitian. Alat terbagi menjadi dua kategori, yaitu perangkat keras (hardware) dan perangkat lunak (software).

1. Perangkat keras (*hardware*)

Penelitian ini dilakukan menggunakan seperangkat laptop dengan spesifikasi sebagai berikut.

- Processor: Intel® Core(TM) i5-7200U CPU @ 2.50GHz
- Kartu grafis: NVIDIA GeForce 940MX
- Memory: 12GB RAM
- Penyimpanan data: HDD 1TB

2. Perangkat lunak (*software*)

Penelitian ini didukung menggunakan software tools atau perangkat lunak sebagai berikut.

- Operating System Windows 10
- Microsoft Edge Web browser
- Python
- TensorFlow
- Keras
- Google Collaboratory dengan GPU
- Python *Libraries*

3.3 Hipotesis

Hipotesis dari penelitian ini adalah sebagai berikut :

Hipotesis 1 : Model yang menggunakan transfer learning MobileNetV2 mendapatkan peningkatan kinerja dibandingkan dengan model yang tidak menggunakan transfer learning MobileNetV2.

Hipotesis 2 : Tuning CNN dengan menambah hyperparameter akan meningkatkan akurasi model.