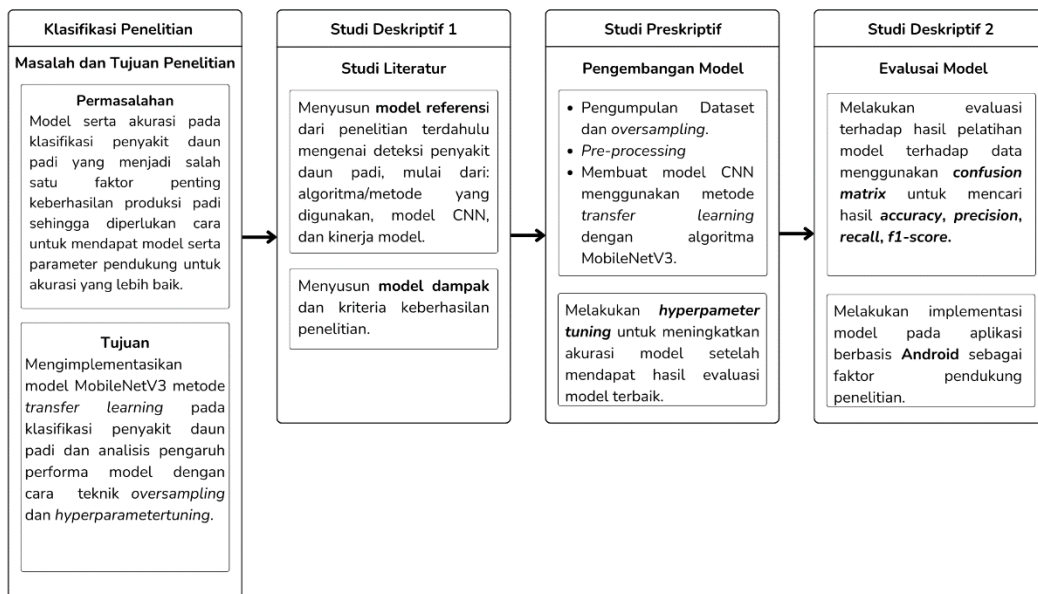


BAB III METODOLOGI PENELITIAN

3.1 Desain Penelitian

Desain Penelitian merupakan tahap-tahap atau alur kerja yang telah direncanakan untuk penelitian yang dilakukan. Penelitian ini berfokus pada pengembangan solusi suatu model CNN untuk deteksi penyakit daun padi yang dapat diimplementasikan langsung di dunia nyata. Berdasarkan hal tersebut, desain metode yang digunakan dalam penelitian ini adalah *Design Research Methodology* (DRM) yang dirancang oleh Blessing dan Chakrabarti pada tahun 2009. Tahapan-tahapan yang ada pada metode penelitian tersebut terdiri dari klarifikasi penelitian, studi deskriptif 1, studi preskriptif serta studi deskriptif 2 (Wynn dan Clarkson, 2024) . Rancangan alur yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Desain Penelitian

3.1.1 Klasifikasi Penelitian

Pada tahapan pertama pada metode DRM adalah klarifikasi penelitian. Tahap ini bertujuan untuk menguraikan permasalahan dan tujuan penelitian berdasarkan latar belakang identifikasi masalah penelitian yang akan dilakukan. Hasil dari identifikasi masalah ini digunakan sebagai dasar untuk mengembangkan

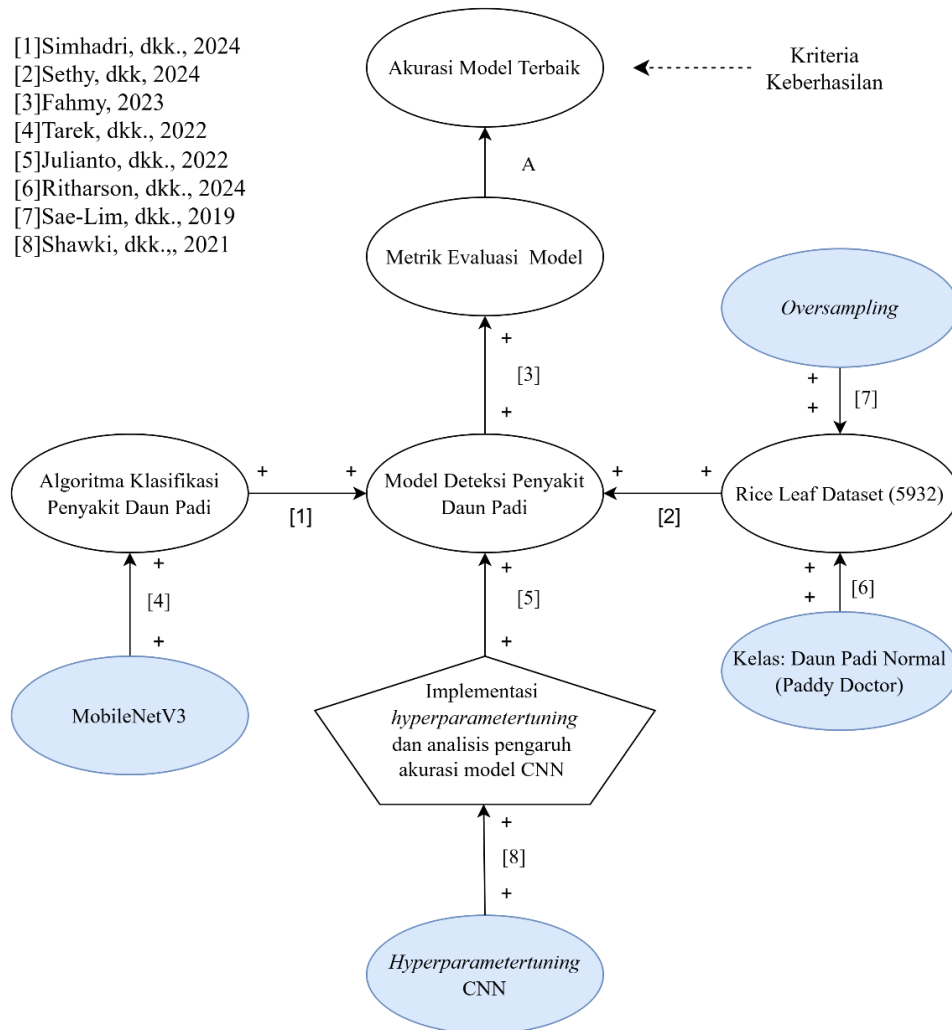
solusi terhadap masalah penelitian yang ingin dicapai sebagai target penelitian, juga menyusun sebuah model referensi, model dampak, dan kriteria keberhasilan pada tahap selanjutnya. Dari tahap pertama ini penulis mengambil topik penelitian mengenai klasifikasi penyakit daun padi menggunakan model CNN dengan fokus penelitian untuk membandingkan algoritma MobileNetV3Large dengan pengaruh dari teknik *oversampling* dan tanpa teknik *oversampling*, teknik *oversampling* yang digunakan adalah *random over-sampling*, kemudian menganalisis *hyperparameter tuning* dalam peningkatan performa model terbaik dari model tersebut.

3.1.2 Studi Deskriptif I

Pada tahap kedua, dilakukan studi literatur untuk menemukan masalah penelitian mengenai aspek-aspek terkait klasifikasi penyakit pada daun padi menggunakan model CNN dengan metode *transfer learning* dari penelitian terdahulu dan dirangkum sebagai model referensi. Kemudian, tahap ini juga akan menghasilkan model dampak dengan kriteria keberhasilan pada penelitian. Model dampak merupakan sebuah model sebagai gambaran acuan keberhasilan dan juga faktor-faktor yang menjadi dampak pada perubahan penelitian yang akan digunakan dengan didukung oleh penelitian sebelumnya.

Pada Gambar 3.2 merupakan diagram model dampak yang akan menjadi acuan pada penelitian dengan merujuk pada penelitian yang berhasil sebelumnya. Pada penelitian ini menerapkan model MobileNetV3 yang sebelumnya pada penelitian Tarek dkk. (2022) berhasil mencapai akurasi yang tinggi pada penyakit daun tanaman lain. Kemudian, menerapkan *oversampling*, merujuk pada penelitian Sae-Lim dkk. (2019) menyebutkan bahwa teknik *oversampling* dapat membantu mengatasi permasalahan dataset seperti kekurangan dataset dan dataset yang tidak seimbang. Dengan tujuan menambah jenis padi, pada penelitian ini mengikuti penelitian dari Ritharson dkk. (2024) yang menambah kelas jenis daun padi normal pada dataset, namun dengan sumber berbeda yakni dari Paddy Doctor (Petchiammal, dkk., 2023). Kemudian menerapkan *hyperparameter tuning*, merujuk pada penelitian Shawki dkk. (2021) menyebutkan bahwa *hyperparameter tuning* merupakan aspek krusial dalam memberi solusi yang optimal dalam membantu kinerja pada pelatihan model agar menghasilkan model dengan akurasi

yang terbaik. Aspek-aspek tersebut merupakan aspek pembaharuan yang akan digunakan dalam penelitian ini untuk mencapai kriteria keberhasilan.



Gambar 3.2 Model Dampak

3.1.3 Studi Preskriptif

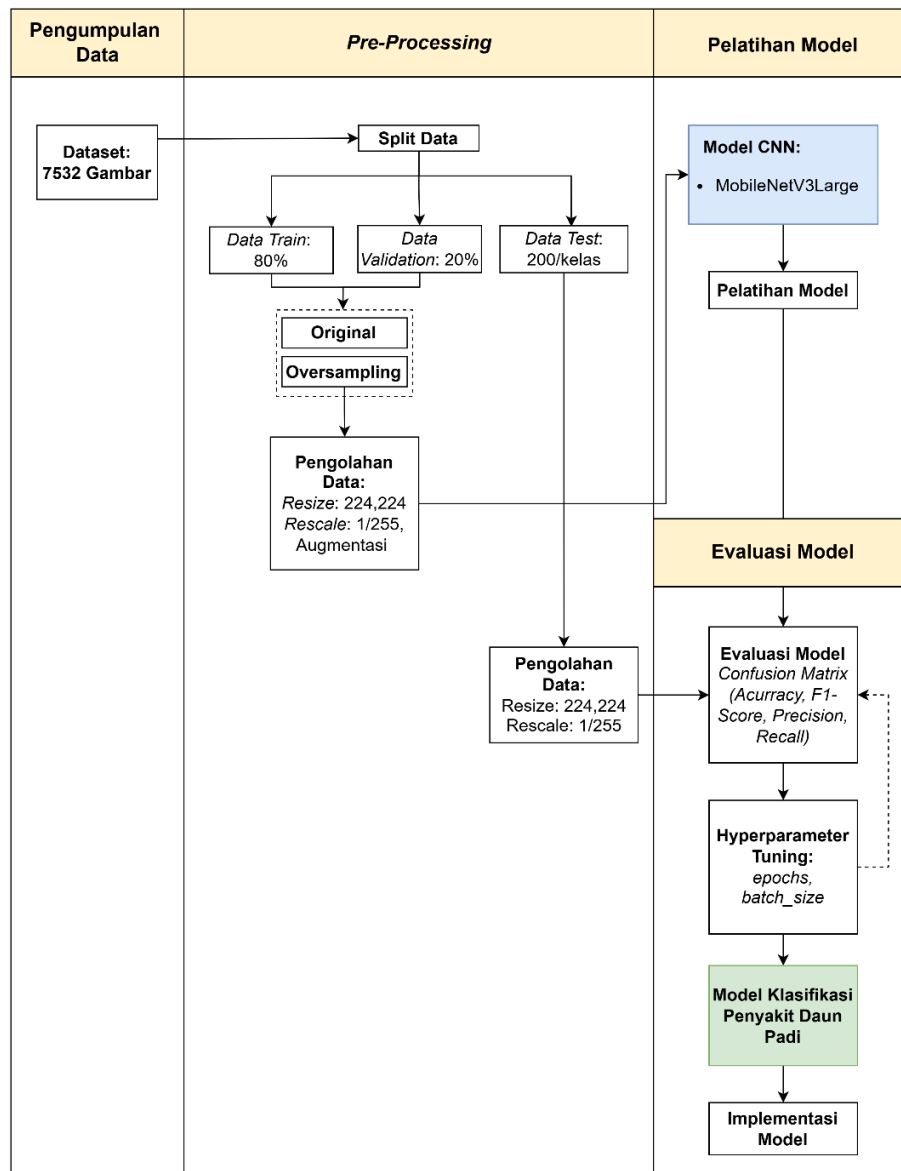
Pada tahapan ini bertujuan untuk menganalisis metode yang digunakan, dengan melakukan perencanaan pengembangan model untuk klasifikasi penyakit pada daun padi menggunakan model CNN dengan metode *transfer learning* MobileNetV3 yang dijabarkan melalui beberapa tahapan. Tahapan pengembangan model dibuat berdasarkan referensi pada penelitian terdahulu dengan beberapa perubahan untuk menyesuaikan data yang akan dianalisis, kemudian model yang sudah dikembangkan akan dievaluasi menggunakan metrik evaluasi *accuracy*, *f1-score*, *precision*, *recall* yang dihitung berdasarkan *confusion matrix*. Setelah

mendapat model dengan akurasi terbaik, model tersebut akan melalui proses *hyperparameter tuning* untuk menganalisis parameter yang berpengaruh dalam optimalisasi pelatihan model sehingga meningkatkan akurasi model.

3.1.3.1 Pengembangan Model Deteksi Penyakit Daun Padi

Dalam proses mengembangkan model deteksi penyakit daun padi akan dibagi melalui beberapa tahap. Model yang akan dibuat menggunakan model CNN metode *transfer learning*. Model yang akan dikembangkan sebelumnya telah dipilih dengan mencoba menggunakan model pembaharuan dari penelitian sebelumnya yakni MobileNetV3Large yang merupakan versi terbaru dari MobileNet dan mendapat akurasi tinggi pada kasus klasifikasi penyakit daun tanaman (Tarek, dkk., 2022). Model akan dilatih menggunakan dua jenis dataset yakni dataset yang tidak menggunakan teknik *oversampling* dan menggunakan teknik *oversampling*. Tujuannya adalah untuk melihat perbandingan akurasi model dan menganalisis pengaruh teknik *oversampling* terhadap akurasi model. Teknik *oversampling* yang digunakan adalah *random over-sampling*, yakni menambahkan jumlah gambar pada kelas minoritas yang menyesuaikan dengan jumlah kelas mayoritas dengan menduplikasi gambar secara acak (Sae-Lim, dkk., 2019).

Kemudian, kinerja dari model setelah pelatihan akan melalui proses pengujian akan dibandingkan dengan beberapa metrik evaluasi dan hasil tersebut akan menentukan model terbaik, kemudian model terbaik akan melalui proses *hyperparameter tuning* untuk mencari komposisi parameter dengan akurasi paling tinggi sehingga cocok untuk digunakan para pengembang model untuk mendapat parameter yang optimal untuk pelatihan (Shawki, dkk., 2021). Setelah model tersebut mendapat hasil yang optimal, kemudian model akan diimplementasikan menggunakan aplikasi berbasis Android yang dapat digunakan sebagai dasar bagi pengembang melakukan penelitian lebih lanjut. Alur pengembangan model klasifikasi penyakit padi tanpa *oversampling* ditunjukkan pada Gambar 3.3.



Gambar 3.3 Pengembangan Model Klasifikasi Penyakit Daun Padi

3.1.3.2 Pengumpulan Data

Pada pengembangan model klasifikasi penyakit daun padi menggunakan *dataset* Rice Leaf Disease Image Samples yang berasal dari website Mendeley Data yang dapat diakses melalui laman <https://data.mendeley.com/datasets/fwcj7stb8r/1>. *Dataset* Rice Leaf Disease Image Samples berisi 5.932 gambar dengan 4 jenis penyakit daun padi yakni *Bacterial blight* (Hawar Daun Bakteri Padi) 1.584 gambar, *Blast* (Blas) 1.440 gambar, *Brown Spot* (Bercak Coklat) 1.600 gambar dan Tungro 1.372 gambar. Penentuan *dataset* tersebut dikarenakan *dataset* tersebut memiliki foto jenis penyakit yang umumnya terjadi di Indonesia dengan beberapa pendukung dari artikel penelitian lain yang sudah menggunakan *dataset* tersebut

Reyhan Agus Priyatna, 2024

PENGEMBANGAN MODEL CONVOLUTIONAL NEURAL NETWORK METODE TRANSFER LEARNING MOBILENETV3 UNTUK KLASIFIKASI PENYAKIT DAUN PADI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

(Sethy, dkk., 2020; M. Aggarwal, dkk., 2023; Ritharson, dkk., 2024), kemudian menggunakan *dataset* Paddy Doctor yang berasal dari penelitian Petchiammal, dkk. (2023) yang berisi 1.600 gambar daun padi yang tergolong *healthy* (normal) sebagai kelas tambahan sehingga penulis dapat membuat model dengan hasil yang relevan dan dapat digunakan secara langsung oleh pengembang aplikasi yang nantinya dipakai oleh petani dalam mendeteksi penyakit daun padi.

3.1.3.3 Pra-Pemrosesan Data

Tahapan pra-pemrosesan data diawali dengan pembagian *dataset*, pertama *dataset* diambil sebanyak 15% yakni 200 gambar setiap kelasnya untuk dijadikan data *test*, bertujuan untuk menguji performa model secara independen menggunakan data *test* untuk mengukur generalisasi model pada data baru dan digunakan pada *confusion matrix* untuk mendapat nilai akurasi pada model sehingga data tersebut harus dipisahkan dari data pelatihan (Xiao, dkk., 2024). Kemudian data yang tidak diambil sebelumnya menjadi data pelatihan, penelitian Bichri, dkk. (2024) menunjukkan bahwa menggunakan data pelatihan lebih dari 70% dapat meningkatkan akurasi model. Oleh karena itu, data pelatihan dibagi menjadi 80% untuk data *train* dan 20% untuk data *validation*. Pembagian ini juga merujuk pada penelitian Sethy, dkk. (2020) yang mendukung pembagian tersebut untuk mengoptimalkan pelatihan model. Rincian jumlah data setelah pembagian data ditunjukkan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1
Jumlah Data Setiap Kelas Sebelum *Oversampling*

Jenis Daun Padi	Dataset Original	Dataset Pelatihan	Data Train (80%)	Data Val (20%)	Data Test
Hawar Daun	1.584	1.384	1107	277	200
Blas	1.440	1.240	992	248	200
Bercak Coklat	1.600	1.400	1120	280	200
Tungro	1.308	1.108	886	222	200
Normal	1.600	1.400	1120	280	200
Total	7.532	6.532	5225	1307	1000

Tabel 3.2
Jumlah Data Setiap Kelas Setelah *Oversampling*

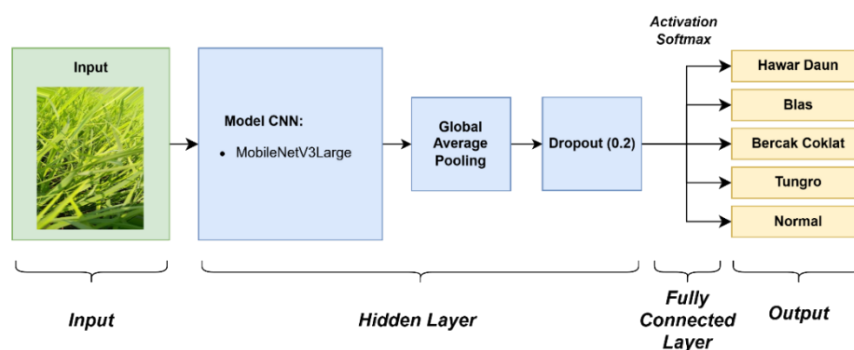
Jenis Daun Padi	Dataset Pelatihan	Dataset <i>Oversampling</i>	Data <i>Train</i> (80%)	Data <i>Val</i> (20%)	Data <i>Test</i>
Hawar Daun	1.384	1.400	1120	280	200
Blas	1.240	1.400	1120	280	200
Bercak Coklat	1.400	1.400	1120	280	200
Tungro	1.108	1.400	1120	280	200
Normal	1.400	1.400	1120	280	200
Total	6.532	7.000	5600	1400	1000

Pada Tabel 3.1 jumlah dataset sebanyak 7.532 setelah ditambahkan kelas normal, kemudian data tersebut diambil 200 setiap kelasnya dan menghasilkan jumlah 6.532 untuk dataset pelatihan, data tersebut yang digunakan untuk pembagian data yang terdiri dari data *train* dan data *val*. Sedangkan pada Tabel 3.2, dataset pelatihan akan melalui proses *oversampling* menggunakan *random oversampling* dengan menambahkan gambar pada kelas di bawah jumlah kelas mayoritas yakni bercak coklat sebesar 1.400 dengan menduplikasi gambar secara acak, sehingga setiap kelas memiliki jumlah yang sama dan seimbang. Kemudian data tersebut digunakan untuk pembagian data (Sae-Lim, dkk., 2019).

Setelah membagi *dataset*, kemudian data melalui proses pengolahan data agar dapat diproses oleh model. Pada setiap data termasuk data *train*, data *validation*, dan data *test*, setiap gambar akan diubah ukurannya menjadi 224x224 piksel agar ukuran pada setiap gambar sama mengikuti ketentuan dari model dan dilakukan proses *scaling* 1./255. Kemudian, pada data *train* dan data *validation* dilakukan augmentasi dengan parameter *rotation_range=20* untuk memutar gambar sebesar 20 derajat, *zoom_range=0.4* untuk memperbesar gambar menjadi 40%, dan *horizontal_flip=True* membalik gambar secara horizontal, proses augmentasi tersebut merujuk pada penelitian Oktaviana dkk. (2021) yang dapat mengurangi *overfitting* pada model untuk klasifikasi daun padi. Sedangkan data *test* tidak melalui proses tersebut agar tetap menjaga keutuhan gambar dari kondisi nyata di lapangan.

3.1.3.4 Arsitektur Model CNN

Tahap selanjutnya adalah membangun arsitektur pada model CNN, pada pembangunan model klasifikasi penyakit daun padi dengan metode *transfer learning* menggunakan pengaturan parameter yang merujuk pada penelitian (Putra, dkk., 2023) sebagai parameter sederhana untuk melakukan pengujian terhadap model *pre-trained* seperti metode *transfer learning*, perbedaan pada penelitian ini dengan sebelumnya yakni menggunakan model MobileNetV3Large sebagai *base model*. Pada pengembangan model tersebut perlu menambahkan beberapa parameter pendukung dengan menggabungkan pada pembangunan model tersebut perlu menambahkan beberapa parameter pendukung dengan menggabungkan *global_average_pool* dan *dropout* sebesar 0.2, kemudian *dense_layer* dengan dengan 5 kelas dan fungsi aktivasi *softmax* di akhir *layer* untuk jenis model klasifikasi gambar ke beberapa kelas. Tahapan pembangunan model untuk penelitian ini digambarkan pada Gambar 3.5.



Gambar 3.4 Pengembangan Arsitektur Model MobileNetV3 Metode *Transfer Learning*

3.1.3.5 Pelatihan Model Deteksi Penyakit Tanaman Daun Padi

Setelah pembangunan arsitektur model selesai, proses selanjutnya adalah pelatihan model atau *model training* yang merupakan proses untuk melatih setiap data melalui model yang telah dibangun. Pada tahap ini menentukan parameter yang akan digunakan untuk pelatihan model awal, parameter pelatihan pada penelitian ini merujuk pada penelitian sebelumnya, namun terdapat penyesuaian terhadap penggunaan *optimizer*. Penelitian ini menggunakan *batch size* sebesar 64, *learning rate* yang dipilih adalah 0.001, kemudian menggunakan jenis *loss categorical_crossentropy* karena cocok untuk tugas klasifikasi dengan lebih dari dua dan proses pelatihan akan berlangsung selama 50 *epoch* (Sethy, dkk., 2020).

Serta untuk *optimizer* yang digunakan adalah *Adaptive Moment Estimation Method* atau Adam, metode tersebut menggunakan estimasi adaptif dari momen-momen rendah untuk menghitung laju pembelajaran yang berbeda untuk setiap parameter. Adam menggabungkan keunggulan dari dua metode populer, yaitu AdaGrad dan RMSProp, sehingga cocok untuk masalah dengan data besar dan parameter yang banyak dan menunjukkan kinerja lebih baik daripada metode optimasi lain seperti SGD (Raiaan, dkk., 2024).

3.1.3.6 Hyperparameter tuning

Tahap ini dilakukan setelah menemukan model terbaik setelah pelatihan model dengan tujuan untuk mencari komposisi parameter terbaik untuk pengembangan model klasifikasi penyakit daun padi. Parameter yang akan menjadi pertimbangan meliputi jenis *hyperparameter general* seperti, *epoch*, dan *batch_size* yakni parameter yang mempengaruhi pelatihan model dan saling berkaitan dengan data, karena data yang telah dibagi dengan *batch size* akan dilatih dalam setiap iterasi/*epoch* (Julianto, dkk., 2022).

Tabel 3.3
Skenario *Hyperparameter tuning*

Parameter	Skenario
<i>Epoch</i>	[50, 100]
<i>Batch_Size</i>	[32, 64]

Berdasarkan Tabel 3.3, skenario *epoch* yakni meningkatkan nilai *epoch* menjadi 100 dan menurunkan *batch size* sebesar 32., hal ini dikarenakan semakin meningkat nilai *epoch* dapat meningkatkan kemampuan model untuk mengenal data, penelitian Alsaadi dan El Abbadi (2019) menunjukkan bahwa penggunaan 100 *epoch* dapat meningkatkan akurasi mencapai nilai optimal. Namun, menggunakan lebih dari 100 *epoch* tidak memberikan peningkatan akurasi yang signifikan dan memerlukan waktu pelatihan yang lebih lama. Oleh karena itu, 100 *epoch* sudah cukup untuk mencapai akurasi optimal. Sedangkan penentuan nilai *batch size* merujuk pada penelitian Kandel & Castelli (2020) yang merekomendasikan pelatihan model cukup dengan nilai *batch size* kecil yakni antara 32 dan 64.

3.1.4 Studi Deskriptif 2

Tahap akhir dari penelitian ini adalah melakukan evaluasi menggunakan hasil pelatihan model yang telah dilakukan untuk mendapat kesimpulan. Evaluasi pertama dilakukan pada data *train* dan data *validation* dengan menghitung akurasi dan *loss* serta menghitung waktu komputasi masing-masing model. Selanjutnya dilakukan evaluasi terhadap model dengan data *test* menggunakan *confusion matrix*, sehingga mendapat *accuracy*, *precision*, *recall* dan *f1-score* sebagai hasil kinerja model (Julianto dan Sunyoto, 2021). Setelah mendapat model yang optimal, akan diimplementasikan pada aplikasi berbasis Android sebagai faktor pendukung penelitian dan dasar untuk penelitian selanjutnya .

3.2 Instrumen Penelitian

Instrumen penelitian atau alat yang ditujukan untuk mengumpulkan data pada penelitian ini berupa metrik *confusion matrix* dan *library* yang digunakan untuk membantu proses pelatihan model sehingga didapatkan suatu data. *Confusion matrix* yang digunakan pada penelitian ini termasuk kategori *multiclass* (Fahmy, 2023), ukuran *confusion matrix* yang digunakan pada penelitian ini dibagi berdasarkan kelas pada dataset, pada penelitian ini terdapat 5 kelas yakni Hawar Daun, Blas, Bercak Coklat, Tungro, dan Normal. Skema *confusion matrix* ditunjukkan pada Tabel 3.4 berisi posisi pada setiap kelas dengan format *Cell* diikuti nomor baris dan kolom ($C_{\text{baris,kolom}}$).

Tabel 3.4
Confusion Matrix Model

		Prediksi				
		HawarDaun	Blas	BercakCoklat	Tungro	Normal
Aktual	HawarDaun	C11	C12	C13	C14	C15
	Blas	C21	C22	C23	C24	C25
	BercakCoklat	C31	C32	C33	C34	C35
	Tungro	C41	C42	C43	C44	C45
	Normal	C51	C52	C53	C54	C55

Tabel 3.5
Bagan Perhitungan Nilai *Confusion Matrix* Model

Class	TP	FP	FN	TN
HawarDaun	C11	C21 + C31 + C41 + C51	C12 + C13 + C14 + C15	C22 + C23 + C24 + C25 + C32 + C33 + C34 + C35 + C42 + C43 + C44 + C45 + C52 + C53 + C54 + C55
Blas	C22	C12 + C32 + C42 + C52	C21 + C23 + C24 + C25	C11 + C13 + C14 + C15 + C31 + C33 + C34 + C35 + C41 + C43 + C44 + C45 + C51 + C53 + C54 + C55
BercakCoklat	C33	C13 + C23 + C43 + C53	C31 + C32 + C34 + C35	C11 + C12 + C14 + C15 + C21 + C22 + C24 + C25 + C41 + C42 + C44 + C45 + C51 + C52 + C54 + C55
Tungro	C44	C14 + C24 + C34 + C54	C41 + C42 + C43 + C45	C11 + C12 + C13 + C15 + C21 + C22 + C23 + C25 + C31 + C32 + C34 + C35 + C51 + C52 + C54 + C55
Normal	C55	C15 + C25 + C35 + C45	C51 + C52 + C53 + C54	C11 + C12 + C14 + C14 + C21 + C22 + C24 + C24 + C31 + C32 + C34 + C34 + C41 + C42 + C43 + C44

Dari Tabel 3.5 di atas, akan mendapat nilai *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Contohnya dari kelas Hawar Daun sebagai kelas 1, dapat didefinisikan:

- TP1: Kelas-1 yang diprediksi dengan benar sebagai kelas-1. Ini adalah nilai pada sel C11 saja, terletak pada baris Kelas-1 dan kolom Kelas-1.
- TN1: Bukan sampel Kelas-1 (yakni sampel kelas-2 atau kelas-n) diklasifikasikan dengan benar atau salah sebagai bukan kelas-1. Ini adalah

jumlah dari sel C_{22} , C_{n2} , C_{2n} , C_{nn} , bagian dari matriks yang tersisa selain baris kelas-1 dan kolom kelas-1.

- FP1: Bukan sampel kelas-1 yang diklasifikasikan secara tidak benar sebagai kelas-1. Ini adalah jumlah dari sel C_{21} dan C_{n1} bagian kolom kelas-1 selain TP1.
- FN1: Bukan sampel kelas-1 yang diklasifikasikan secara tidak benar sebagai bukan kelas-1. Ini adalah jumlah dari sel C_{12} dan C_{1n} bagian baris kelas-1 selain TP1.

Dari hasil TP, TN, FP, dan FN. Selanjutnya dapat digunakan untuk perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1-score* seperti ditunjukkan pada Tabel 3.6.

Tabel 3.6
Rumus-Rumus Evaluasi Model

Rumus-Rumus
$Accuracy(\text{kelas}) = \frac{TP+TN}{TP+TN+FP+FN}$
$Precision(\text{kelas}) = \frac{TP}{TP+FP}$
$Recall(\text{kelas}) = \frac{TP}{TP+FN}$
$F1(\text{kelas}) = \frac{2(Precision(\text{kelas}) \times Recall(\text{kelas}))}{(Precision(\text{kelas}) + Recall(\text{kelas}))}$
$Accuracy(\text{model}) = \frac{TP(\text{semua kelas})}{N(\text{jumlah data})}$
$Precision(\text{model}) = \frac{\text{Jumlah Precision}(\text{semua kelas})}{5}$
$Recall(\text{model}) = \frac{\text{Jumlah Recall}(\text{semua kelas})}{5}$
$F1(\text{model}) = \frac{2(Precision(\text{model}) \times Recall(\text{model}))}{(Precision(\text{model}) + Recall(\text{model}))}$

Berdasarkan Tabel 3.6 di atas, terdapat dua macam untuk hasil evaluasi yakni berdasarkan kelas dan model atau keseluruhan yang saling berkaitan, untuk mendapat hasil evaluasi model diperlukan nilai evaluasi dari masing-masing kelas yang dihitung berdasarkan rumus terhadap kelas kemudian dihitung rata-rata atau berdasarkan rumus terhadap model. Kemudian terdapat daftar *library* yang digunakan untuk menyediakan alat dan fungsi yang mendukung pengembangan model, termasuk manipulasi data, pembangunan model, visualisasi hasil, serta evaluasi performa model yang ditunjukkan pada Tabel 3.7.

Tabel 3.7
Daftar *Library*

Nama <i>Library</i>	Deskripsi
Tensorflow	Sebagai <i>framework</i> dalam membangun dan proses pelatihan model <i>Convolutional Neural Network</i> (CNN).
TensorFlow Lite	Sebagai <i>framework</i> dari TensorFlow untuk menjalankan model di perangkat seluler.
OS	Untuk mengatur dan memanipulasi direktori file.
Numpy	Memberikan struktur data array yang efisien untuk manipulasi dan operasi matematika pada data gambar.
Matplotlib	Digunakan untuk membuat visualisasi grafik seperti kurva <i>loss</i> dan akurasi selama pelatihan model.
ImageDataGenerator	Menggunakan augmentasi gambar seperti rotasi, <i>zoom</i> , dan <i>flip</i> untuk meningkatkan keragaman data pelatihan.
Keras	Menyediakan <i>high-level</i> API untuk membangun dan mengelola model <i>Convolutional Neural Network</i> (CNN).
Sklearn	Menyediakan algoritma dan tools untuk evaluasi model, pemilihan model, dan tuning parameter.
Seaborn	Digunakan untuk visualisasi data statistik yang lebih kompleks dan informatif, seperti distribusi kelas dalam dataset.
Pandas	Digunakan untuk analisis data secara struktural

3.3 Alat dan Bahan Penelitian

Pada penelitian ini menggunakan beberapa alat dan bahan seperti perangkat lunak dan perangkat keras yang digunakan dalam proses mengumpulkan data, menjalankan suatu program dan sebagai dokumentasi penelitian, dengan spesifikasi yang ditampilkan pada Tabel 3.8. Selanjutnya terdapat konfigurasi terhadap perangkat yang digunakan saat implementasi model menggunakan perangkat Android yang ditunjukkan pada Tabel 3.9.

Tabel 3.8
Konfigurasi Perangkat Pengembangan Model

Konfigurasi		Keterangan
<i>Hardware</i>	GPU	GeForce GTX 1650
	RAM	8 GB
<i>Software</i>	OS	Windows 11
	Python	3.10
	IDE	Visual Studio Code, Android Studio
<i>Environment</i>	TensorFlow	2.16
	CUDA	11.2
	CUDNN	8.1

Tabel 3.9
Konfigurasi Perangkat Implementasi Model

Konfigurasi	Keterangan
Nama Perangkat	Redmi Note 9
RAM	4 GB
OS	Android 12
Chipset	Mediatek MT6769Z Helio G85
Kamera	48 MP (Min. 8 MP)

3.4 Analisis Data

Setelah mendapatkan hasil dari setiap model yang sudah melewati proses pelatihan dan evaluasi, kemudian hasil tersebut dianalisis menggunakan analisis komparatif yakni perbandingan dari hasil performa setiap model untuk mendapat kesimpulan penelitian menggunakan *dataframe* pada *library* Pandas. Pada penelitian ini faktor perbandingan antara model ditentukan dari dua tahap, pertama yakni berupa *accuracy* dan *loss* yang diperoleh dari hasil pelatihan model terhadap data *train* dan *validation* untuk mendapat hasil kinerja model saat pelatihan, kemudian perbandingan terhadap nilai *accuracy*, *precision*, *recall*, dan *f1-score* dari setiap hasil pengujian prediksi model terhadap data *testing* yang dihitung berdasarkan metrik *confusion matrix* untuk mendapat hasil performa kualitas model secara keseluruhan.