

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penulis menemukan beberapa penelitian terdahulu terkait klasifikasi penyakit tanaman padi dengan metode yang berbeda, seperti *deep learning*, *ensemble learning*, serta mengimplementasikan model yang ada pada sistem progressive web app.

2.1.1. *Paddy Doctor: A Visual Image Dataset for Paddy Disease Classification* [6]

Penelitian dengan judul “*Paddy Doctor: A Visual Image Dataset for Paddy Disease Classification*” yang dilakukan oleh Petchiammal A, Briskline Kiruba S, D. Murugan, dan Pandarasamy A. Penelitian ini dilakukan dengan latar belakang penelitian sebelumnya untuk identifikasi memiliki dataset yang kurang detail. Sehingga mengurangi akurasi deteksi dari klasifikasi penyakit pada tanaman padi. Penelitian terdahulu menggunakan dataset sendiri dengan total gambar 13,876 tanaman padi yang sudah terlabel ke dalam sepuluh kelas. Penelitian terdahulu juga menggunakan metode CNN buatan sendiri dan dua pendekatan *transfer learning* yaitu VGG16 dan MobileNet. Hasil akurasi yang didapat adalah 93.83% untuk MobileNet, 93,66% untuk VGG16, dan 87.97% untuk DCNN.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Dataset yang digunakan bukan hasil potongan ataupun hanya gambar sehelai daun padi, melainkan gambar yang memiliki latar belakang tanaman padi lain.
- Salah satu model dan pendekatan yang digunakan pada penelitian terdahulu yaitu MobileNet menjadi acuan bagi penulis untuk

menggunakan versi terbaru dari MobileNet yaitu MobileNetV3-Large.

- Penggunaan ukuran gambar 256×256 *pixel* pada penelitian terdahulu memberikan akurasi yang sangat baik dan menjadi acuan penulis untuk mencoba menggunakan model yang berbeda.
- Hasil penelitian terdahulu kurang memberikan gambaran secara terperinci perbandingan antar model.

2.1.2. Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (Convolutional Neural Network) [8]

Penelitian dengan judul “Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (Convolutional Neural Network)” yang dilakukan oleh Fakhri Habib Hawari, et al. Penelitian dengan latar belakang untuk meningkatkan produksi padi, minimal menjaga kestabilan ketahanan pangan nasional, dan membantu petani dalam mengklasifikasi penyakit tanaman padi serta mengurangi resiko gagal panen. Dataset yang digunakan menggunakan 1 daun padi yang terdiri dari 4 kelas yaitu *Brown Spot*, *Hawar*, *Leaf Brown*, dan daun sehat. Setiap gambar diubah menjadi 256×256 piksel dengan *batch size* 32. Model yang digunakan juga menggunakan model yang diajukan sendiri. Beberapa layer digunakan seperti *sequential layer*, *conv2D*, *maxPooling2D*, *flatten*, dan *dense layer* untuk membuat model CNN.

Hasil yang didapat dari grafik pembelajaran memiliki nilai akurasi pembelajaran sebesar 85% dan akurasi validasi sebesar 95%. *Training* dilakukan sebanyak 10 epoch. Walaupun memiliki akurasi validasi yang besar, tetapi akurasi uji yang didapat adalah 86% dengan loss yang cukup tinggi yaitu 49%.

Beberapa poin penting yang dapat diambil penulis adalah sebagai berikut:

- Penggunaan teknologi Deep Learning yaitu CNN mampu mengklasifikasikan penyakit tanaman padi dengan akurasi yang cukup baik.
- Berdasarkan saran pada penelitian terdahulu, dapat menambahkan kategori penyakit padi lain. Hal ini dikarenakan kelas penyakit pada penelitian terdahulu ini, hanya memiliki 4 kelas saja.
- Juga menambahkan arsitektur dari CNN lain seperti penggunaan *pre-trained* model untuk meningkatkan akurasi.

2.1.3. *A Deep Learning-Based Mobile App System for Visual Identification of Tomato Plant Disease* [12]

Penelitian dengan judul “*A Deep Learning-Based Mobile App System for Visual Identification of Tomato Plant Disease*” yang dilakukan oleh Aurelius Ryo Wang dan Nabila Husna Shabrina. Penelitian dengan latar belakang tanaman tomat sebagai salah satu peran penting dalam pemenuhan kebutuhan makanan yang rentan terhadap penyakit. Sehingga penelitian ini membuat sistem berbasis aplikasi *mobile* yang dapat mengidentifikasi penyakit pada tanaman tomat dengan metode EfficientNetB0. Hal penting yang dapat penulis dapatkan pada penelitian ini adalah penggunaan EfficientNetB0 pada aplikasi *mobile* dengan waktu prediksi rata-rata 40.3744 ms dan waktu pemrosesan rata-rata 103.183 ms. Selain itu, penelitian terdahulu ini memberikan saran untuk meningkatkan performa hasil klasifikasi dengan mencoba menggunakan model *deep learning* lainnya. Hal ini dikarenakan akurasi validasi yang didapat adalah 92.22% yang dapat dimungkinkan untuk ditingkatkan kembali.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Penggunaan model EfficientNetB0 mampu memberikan performa akurasi validasi yaitu 92.22% pada klasifikasi penyakit

tanaman tomat dan saran yang diberikan penelitian terdahulu, untuk mencoba menggunakan model *deep learning* lainnya untuk meningkatkan akurasi. Sehingga penulis ingin mencoba menggunakan model terbaru dari EfficientNetB0 yaitu EfficientNetV2B0 pada klasifikasi penyakit pada tanaman padi dengan harapan memberikan performa yang sama atau lebih baik.

- Penelitian terdahulu mencoba menganalisa prediksi waktu rata-rata dengan jenis ekstensi gambar yang berbeda-beda (JPEG, *string base 64*, dan *string base 64 x 5*). Hal ini menjadi sumber inspirasi penulis untuk menganalisa prediksi waktu rata-rata, tetapi dengan parameter model, ukuran gambar, dan teknik *balancing dataset* yang berbeda.

2.1.4. *Automatic Identification of Disease in Grains Crops through Computational Approches: A Review* [7]

Penelitian dengan judul “*Automatic Identification of Disease in Grains Crops throught Computational Approaches: A Review*” yang dilakukan oleh R. Manavalan. Penelitian ini berfokus pada membandingkan berbagai metode baik dari sisi *preprocessing*, segmentasi, ekstraksi fitur, seleksi fitur, dan klasifikasi pada berbagai variasi tanaman biji-bijian. Misalnya pada proses *preprocessing* terdapat berbagai metode seperti teknik *region filling*, *filter median* dan lain sebagainya dibandingkan tingkat akurasi masing-masing dari metode tersebut. Selain itu, penelitian terdahulu ini juga mengabungkan berbagai tahap dengan metode terbaik dan mencari kembali akurasi yang paling tinggi. Penulis mendapatkan banyak pembelajaran baru pada penelitian terdahulu ini karena memaparkan secara detail akurasi metode pada setiap tahap deteksi penyakit pada tanaman biji-bijian.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Penelitian terdahulu ini memperkuat latar belakang penelitian penulis mengenai petani yang memiliki pengetahuan yang kurang mengenai penyakit pada tanaman biji-bijian.
- Padi yang sudah matang dan terkena penyakit sulit atau bahkan tidak dapat disembuhkan. Sehingga menjadi poin penting, sistem klasifikasi penyakit tanaman padi yang dibuat oleh penulis, dapat dijadikan acuan bagi petani untuk kedepannya dalam menanam kembali padi yang baru dengan cara penanaman dan penggunaan pestisida yang tepat berdasarkan tanaman padi yang diklasifikasi melalui sistem.
- Penggunaan berbagai metode pada tahap *preprocessing* dan segmentasi menjadi acuan bagi penulis untuk menggunakan tahap *preprocessing* untuk augmentasi gambar agar dapat memperbaiki hasil latih.

2.1.5. Klasifikasi Penyakit Tanaman Padi Menggunakan Model *Deep Learning Efficientnet B3 Dengan Transfer Learning* [13]

Penelitian dengan judul “Klasifikasi Penyakit Tanaman Padi Menggunakan Model *Deep Learning Efficientnet B3 Dengan Transfer Learning*” yang dilakukan oleh Endang Anggiratih, Sri Siswanti, Saly Kurnia Octaviani, dan Arumsari. Penelitian ini berfokus pada membandingkan akurasi dua model yang digunakan yaitu EfficientNetB3 dan MobileNetV3. EfficientNetB3 memiliki akurasi tinggi daripada MobileNetV3 dengan hasil 79.53%. Namun MobileNetV3 memiliki *loss* yang lebih rendah dibandingkan EfficientNetB3 yaitu 0.007. Kekurangan dari penelitian ini adalah hanya menggunakan dua kelas klasifikasi yaitu *brown spot* dan *bacterial leaf*. Sehingga akurasi yang didapat bisa sangat tinggi.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Segmentasi atau pembagian dataset pada penelitian terdahulu ini yaitu 60:20:20 untuk data latih, data validasi, dan data uji memberikan hasil yang sangat baik.
- Penggunaan *epoch* yang berlebihan malah menurunkan tingkat akurasi pada model tersebut. Pada penelitian terdahulu ini, akurasi menurun ketika *epoch* lebih dari 300.
- Akurasi dari EfficientNetB3 dapat ditingkatkan berdasarkan tabel aplikasi keras untuk akurasi tertinggi adalah 79.53%.

2.1.6. *HiT-LIDIA: A Framework for Rice Leaf Disease Classification using Ensemble and Hierarchical Transfer Learning* [14]

Penelitian dengan judul “*HiT-LIDIA: A Framework for Rice Leaf Disease Classification using Ensemble and Hierarchical Transfer Learning*” yang dilakukan oleh Oddy Virgantara Putra, Niken Trisnaningrum, Niken Sylvia Puspitasari, Agung Toto Wibowo, dan Ema Rachmawaty. Penelitian ini berfokus membandingkan beberapa *pre-trained* model yang ada dengan model yang dibuat oleh penulis menggunakan teknik *ensemble* pada penyakit tanaman padi. Penelitian ini menggunakan dua *ensemble* model berbeda dengan tujuan yang berbeda. Model pertama gabungan antara MobileNet dan DenseNet (MK-I) digunakan untuk meneliti tanaman padi yang normal dan berpenyakit. Model kedua gabungan antara DenseNet dan XceptionNet (MK-II) digunakan untuk meneliti tiga penyakit tanaman padi.

Hasil yang didapat, kedua model *ensemble* tersebut mendapatkan akurasi yang lebih tinggi daripada model *pre-trained*. Akurasi yang didapat untuk MK-I adalah 0.89% berada diatas akurasi *pre-trained* model lainnya seperti DenseNet201 yang mendapatkan akurasi 0.88%. Akurasi yang didapat untuk MK-II adalah 0.91% berada diatas akurasi *pre-trained* model lainnya seperti Xception 0.8829%.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Penggunaan metode *ensemble* memiliki akurasi yang lebih tinggi dibandingkan penggunaan *pre-trained* model
- Penyajian evaluasi grafik akurasi dan loss yang rapi dan unik menjadi acuan penulis dalam menyusun hasil penelitian.

2.1.7. *Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images* [15]

Penelitian dengan judul “*Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images*” yang dilakukan oleh Vajira Thambawita, Inga Strumke, Steven A. Hicks, Pal Halvorsen, Sravanthi Parasa, dan Michael A. Reigler. Penelitian ini berfokus pada pengaruh penggunaan berbagai ukuran gambar dalam performa *training* dan juga rata-rata waktu hasil prediksi.

Penelitian ini menggunakan lima ukuran gambar yang berbeda serta dua model yang berbeda sebagai bahan pembandingan. Ukuran gambar yang digunakan adalah *32x32 pixel*, *64x64 pixel*, *128x128 pixel*, *256x256 pixel*, dan *512x512 pixel*. Sedangkan untuk model yang digunakan yaitu DenseNet-161 dan Resnet-152. Dari hasil penelitian yang dilakukan, dapat diambil kesimpulan bahwa semakin tinggi ukuran gambar juga akan meningkatkan performa akurasi dengan peningkatan yang tidak terlalu signifikan baik untuk kedua model. Selain itu hasil kecepatan waktu prediksi rata-rata menunjukkan hasil yang fluktuatif. Sehingga dapat diambil kesimpulan bahwa meningkatkan ukuran gambar tidak memiliki efek yang besar untuk kecepatan prediksi.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

- Peningkatan ukuran gambar juga meningkatkan performa akurasi hasil model yang dilatih.

- Peningkatan ukuran gambar tidak memiliki efek yang besar untuk kecepatan prediksi. Hal ini terbukti dari hasil yang didapat bersifat fluktuatif.

Berdasarkan berbagai penelitian terdahulu yang diambil penulis, berikut beberapa poin yang menjadi acuan penelitian yang dilakukan oleh penulis.

Table 1. Rangkuman Poin Acuan Penulis pada Penelitian Terdahulu

Judul Penelitian (Tahun)	Poin yang menjadi acuan penulis
<i>Paddy Doctor: A Visual Image Dataset for Paddy Disease Classification</i> (2022)	<ul style="list-style-type: none"> • Penelitian terdahulu ini memberikan <i>dataset</i> dengan 13,876 gambar terdiri dari 10 kelas (9 penyakit dan 1 normal) • Memastikan bahwa <i>dataset</i> dapat dilatih menggunakan <i>deep learning</i> • Penggunaan model MobileNet sebagai acuan penulis untuk menggunakan model MobileNet terbaru yaitu MobileNetV3-Large • Penggunaan gambar dengan ukuran 256x256 <i>pixel</i> memberikan hasil yang sangat baik dan menjadi acuan penulis untuk mencoba menggunakan ukuran gambar tersebut dengan model yang berbeda
Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (<i>Convolutional Neural Network</i>)	<ul style="list-style-type: none"> • Saran yang diberikan pada penelitian terdahulu dalam menggunakan <i>pre-trained CNN</i>, menjadi acuan penulis untuk menggunakan <i>pre-trained model</i> • Saran yang diberikan pada penelitian terdahulu dalam menggunakan kelas <i>dataset</i> yang lebih banyak, membuat

	<p>penulis mencoba mencari dataset yang memiliki kelas lebih banyak.</p>
<p><i>A Deep Learning-Based Mobile App System For Visual Identification of Tomato Plant Disease</i></p>	<ul style="list-style-type: none"> • Penggunaan model EfficientNetB0 sebagai acuan penulis untuk menggunakan model EfficientNetV2B0 • Sebagai acuan penulis mengimplementasikan model latih ke sistem PWA • Analisa rata-rata waktu prediksi yang dilakukan menjadi acuan penulis untuk melakukan analisa rata-rata waktu prediksi diimplementasi pada PWA dengan membandingkan model, ukuran gambar, dan teknik <i>balancing dataset</i> yang berbeda
<p><i>Automatic Identification of Diseases in Grains Crops Through Computational Approaches: A Review</i></p>	<ul style="list-style-type: none"> • Percobaan dalam berbagai metode <i>preprocess</i>, menjadi sumber acuan pada penelitian ini.
<p>Klasifikasi Penyakit Tanaman Padi Menggunakan Model <i>Deep Learning</i> EfficientNetB3 Dengan <i>Transfer Learning</i></p>	<ul style="list-style-type: none"> • Mendukung pembagian data latih dan data validasi. • Menunjukkan peningkatan <i>epoch</i> tidak akan selalu linear dengan peningkatan akurasi.
<p><i>HiT-LIDIA: A Framework for Rice Leaf Disease Classification using Ensemble and Hierarchical Transfer Learning</i></p>	<ul style="list-style-type: none"> • Penggunaan metode <i>ensemble</i> yang memiliki akurasi lebih baik daripada <i>pre-trained</i> model, menjadi acuan penulis untuk mencobanya. • Penyajian evaluasi akurasi dan <i>loss</i> yang unik menjadi acuan penulis untuk menyusun penelitian.

<p><i>Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images</i></p>	<ul style="list-style-type: none"> • Penggunaan ukuran gambar yang berbeda ternyata berpengaruh terhadap performa model. Sehingga menjadi acuan penulis untuk mencoba menggunakan dua ukuran gambar yang berbeda yaitu 224x224 pixel dan 256x256 pixel • Hasil waktu rata-rata prediksi ternyata bersifat fluktuatif. Penulis ingin mencoba dengan menggunakan model yang lain serta ukuran gambar 224x224 pixel untuk melihat apakah memang waktu rata-rata prediksi bersifat fluktuatif
---	---

2.2 Tinjauan Teori

2.2.1. Deep Learning

Deep learning merupakan salah satu bagian dari *machine learning*, yang simpelnya memiliki tiga atau lebih layer jaringan neural. Jaringan neural pada *deep learning* meniru cara kerja otak melalui kombinasi antara data input, *weight*, dan *bias*. *Deep learning* terdiri dari banyak lapisan *node* yang saling terhubung dan memiliki dua proses yaitu *forward propagation* dan *back propagation* yang digunakan untuk memprediksi *error* dan menyesuaikan *weight* dan *bias* dari setiap *node* yang ada. Proses ini akan terus berulang sampai mendapatkan akurasi yang sesuai.

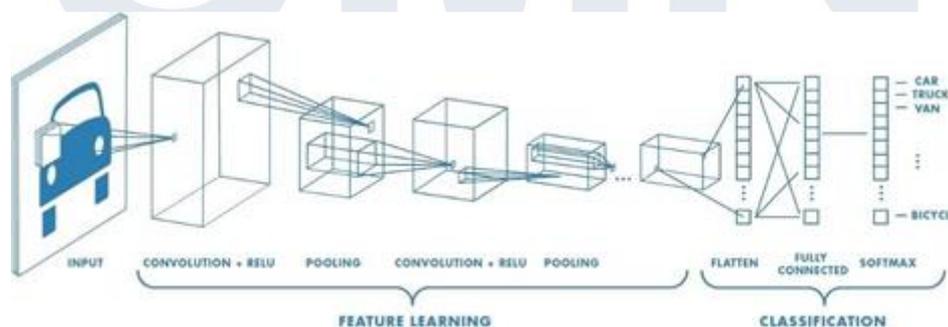
Deep learning sendiri dapat diklasifikasikan ke dalam tiga kategori utama yaitu *unsupervised learning*, *semi-supervised learning*, dan *supervised learning*. *Unsupervised learning* adalah proses pembelajaran yang datanya tidak diberikan label, contohnya adalah *clustering*. *Supervised learning* adalah proses pembelajaran yang datanya diberikan label, contohnya adalah CNN yang akan digunakan pada penelitian ini. *Semi-supervised learning* adalah proses pembelajaran yang datanya beberapa diberi label dan beberapa tidak diberi label, contohnya *Generative Adversarial Networks* (GANs).

2.2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu jenis dari *deep learning* yang dapat digunakan untuk mengklasifikasi, mendeteksi, ataupun mengenali obyek pada suatu gambar. CNN terdiri dari neuron yang memiliki *weight*, *bias*, dan fungsi aktivasi. Strukturnya sendiri meniru struktur neuron pada otak.

CNN bekerja dengan menggerakkan *kernel* (yang dapat diatur ukurannya) pada gambar dan memecah gambar menjadi gambar yang lebih kecil dan saling tindih. Gambar-gambar kecil tersebut akan dimasukkan dalam jaringan *neural* untuk diekstrak fiturnya. Jika ada bagian dari gambar dirasa menarik, maka gambar tersebut akan diambil sebagai *object of interest*. Hasil yang diambil tersebut tersimpan dalam bentuk *array* yang nantinya akan masuk ke proses *downsampling*. *Downsampling* digunakan untuk mengecilkan ukuran dari *array*, tetapi tidak mengurangi informasi penting pada *array* tersebut. *Array* yang berukuran kecil akan masuk ke jaringan saraf terakhir yang bisa disebut dengan *fully connected network* yang bertujuan untuk memprediksi gambar.

CNN sendiri terbagi menjadi 2 bagian, yaitu lapisan ekstraksi fitur, dan lapisan terhubung penuh. Lapisan ekstraksi fitur bertujuan untuk mengubah gambar dalam bentuk angka yang akan menjadi fitur daripada gambar tersebut. Sedangkan untuk lapisan terhubung penuh bertujuan untuk mengubah bentuk *array* fitur gambar menjadi sebuah *vector* yang digunakan untuk mengolah data sehingga dapat diklasifikasikan. Berikut Gambar 2.1 struktur umum dari metode CNN.



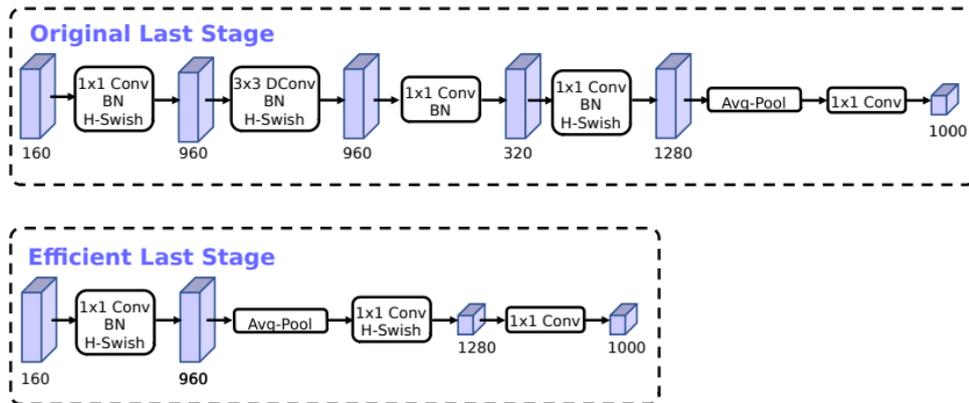
Gambar 2.1 Ilustrasi Arsitektur CNN [18]

2.2.3. MobileNetV3

MobileNetV3 diawali dengan versi pertamanya yaitu MobileNet. MobileNet merupakan model yang dibuat dengan arsitektur jaringan yang efisien dan menggunakan dua *hyperparameter* untuk membangun model yang sangat kecil dan latensi rendah. Sehingga cocok diimplementasi pada perangkat *mobile* maupun aplikasi kamera tertanam. MobileNet dibuat dengan metode *depthwise separable convolution* yang merupakan gabungan dari dua layer yaitu *depthwise convolutions* (menggunakan satu filter pada setiap *input channel*) dan *pointwise convolution* (konvolusi simpel dengan ukuran 1x1). Dengan tambahan *width multiplier* dan *resolution multiplier*. Sehingga dapat memperkecil dan mempercepat model.

Setelah itu muncullah MobileNetV2, dengan urgensi berfokus meningkatkan efisiensi model ke dalam perangkat *mobile* lebih tinggi lagi. Model dasarnya menggunakan pendekatan *bottleneck depth-separable convolution with residuals* dan menggunakan *trade-off hyperparameters*. *Inverted residual bottleneck layer* mengizinkan penggunaan memori yang hemat sangat penting bagi penggunaan aplikasi *mobile*.

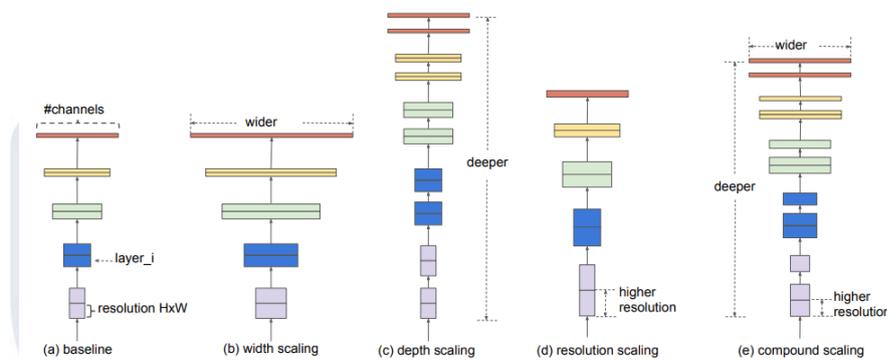
MobileNetV3 versi terbaru dari model MobileNet yang menggunakan dua teknik *network search* yaitu *platform aware NAS* untuk pencarian *block-wise* dan NetAdapt untuk pencarian *layer-wise*. Namun, setelah diteliti ternyata layer awal dan akhir sangat mahal dalam hal komputasi. Sehingga ditawarkan beberapa modifikasi untuk mengurangi latensi pada layer tersebut dengan tetap menjaga akurasi. Modifikasi pertama, pada model MobileNetV2, konvolusi 1x1 digunakan sebagai *final layer* akan dipindahkan ke layer setelah *final average pooling*. Modifikasi kedua, membuang *layer bottleneck* yang ada di belakang layer konvolusi 1x1. Modifikasi ketiga, penggunaan 32 filter pada konvolusi 3x3 tidak terlalu penting dan dapat dikurangi dengan menggunakan *hard swish*. Berikut Gambar 2.2 perubahan layer pada MobileNetV3, baik sebelum dan sesudah dimodifikasi.



Gambar 2.2 Hasil Modifikasi Model dari MobileNetV2 ke MobileNetV3 [10]

2.2.4. EfficientNetV2B0

EfficientNet untuk saat ini memiliki dua versi, EfficientNet dan EfficientNetV2. EfficientNet terbentuk menggunakan pencarian arsitektur neural (NAS) yang menjadi dasar jaringan. Selain itu EfficientNet menggunakan metode *compound scaling* yang merupakan gabungan dari tiga skala yaitu *width*, *depth*, dan *resolution scaling*. Metode *compound scaling* ini yang akan menciptakan EfficientNetB1 sampai EfficientNetB7.



Gambar 2.3 Compound Scaling pada EfficientNet [21]

Selanjutnya EfficientNetV2 datang yang merupakan versi terbaru, lebih cepat, dan memiliki performa lebih baik daripada EfficientNet. Hal yang membuat EfficientNetV2 memiliki performa lebih baik sekaligus perbedaan dari versi sebelumnya yaitu:

- EfficientNetV2 menggunakan MBConv dan Fused-MBConv (pada awal layer). Sedangkan EfficientNet hanya menggunakan

MBCConv. Fused-MBCConv memberikan performa yang lebih baik pada *mobile* atau akselerator server. Namun tidak semua layer pada EfficientNetV2 diubah dari MBCConv menjadi Fused-MBCConv. Karena pada percobaan yang dilakukan, agar mendapatkan performa yang lebih baik, harus terdapat kombinasi antara keduanya.

- EfficientNetV2 menggunakan *training aware NAS framework* yang lebih besar daripada *NAS Framework*.
- EfficientNetV2 memiliki layer MBCConv yang lebih sedikit untuk mengurangi kelebihan akses memori
- EfficientNetV2 menghapus *stride* terakhir pada *original* EfficientNet, untuk mengurangi besarnya ukuran parameter

2.2.5. Ensemble Learning

Ensemble learning merupakan salah satu metode untuk menggabungkan beberapa model *deep learning*. Hasil penggabungan beberapa model tersebut menghasilkan model baru yang memberikan hasil prediksi lebih baik. Selain itu penggunaan *ensemble learning* juga dapat membantu mengurangi penggunaan dataset yang beragam, mengatasi *overfitting*, dan tidak membutuhkan waktu yang lama dalam membuat model yang baru.

Terdapat beberapa teknik untuk menggabungkan beberapa algoritma deep learning [22], sebagai berikut:

- *Average Method*

Metode *Average* membuat model baru dengan cara mengambil rata-rata sederhana dari dua atau lebih model yang memiliki akurasi yang lebih baik dibandingkan dengan gabungan dua model. Metode *Average* mengambil rata-rata dari nilai weight pada masing-masing model yang digabungkan.

- *Bagging Method*

Metode *Bagging* adalah metode yang menggabungkan versi atau iterasi yang berbeda pada masing-masing model *deep learning* untuk

menghasilkan model yang lebih akurat dan mempunyai performa yang tinggi. Caranya mirip sama dengan metode *Average*, tetapi perbedaannya adalah model baru yang terbuat, tidak memiliki *bias*.

- *Boosting Method*

Metode *Boosting* mirip seperti penggunaan *feedback loop* pada suatu model. Performa model yang sekarang digunakan untuk penyesuaian pada model berikutnya. Ini akan menciptakan *feedback loop* yang bersifat positif dengan mengumpulkan semua faktor yang membantu kesuksesan suatu model.

- *Concatenation Method*

Metode ini menggunakan *input* yang berbeda pada masing-masing model dan menggabungkannya ke satu model yang sama. Sehingga dataset yang akan dilatih akan memiliki dimensi yang lebih banyak dibandingkan *dataset* awal. Namun, jika kita menggabungkan beberapa model secara berulang, akan menyebabkan *overfitting* dan hilangnya informasi penting pada data yang dilatih.

- *Stacking Method*

Metode *Stacking* mengintegrasikan beberapa model *deep learning* layaknya disusun untuk mendapatkan performa yang lebih baik. Sempelnya adalah hasil dari model pertama akan digunakan untuk model kedua.

2.2.6. Balanced dan Imbalanced Dataset

Dalam mengumpulkan data, wajar jika data yang terkumpul tidaklah seimbang, ada yang lebih sedikit dan ada yang lebih banyak. Contohnya ketika mengidentifikasi penyakit pada tanaman padi, penyakit Dead Heart lebih sering ditemukan daripada penyakit Bacterial Panicle Blight. Hal ini tercermin pada *database* yang akan digunakan oleh penulis. Dataset inilah yang disebut *dataset* yang tidak *balanced* atau *imbalanced dataset*. *Imbalanced dataset* adalah ketika terdapat perbedaan yang cukup tinggi antara kelas positif dan kelas negatif. Sedangkan *balanced dataset* adalah ketika jumlah data kelas positif dan kelas negatif memiliki jumlah data yang hampir sama.

Imbalanced data tidak selalu buruk bagi performa model, jika perbedaan antar kelas tidak terlalu signifikan. Namun alangkah baiknya untuk melakukan *balancing data* untuk mendapatkan performa model lebih baik. Terdapat beberapa cara untuk *membalancing* data yang penulis gunakan untuk penelitian ini, yaitu:

- *Over-sampling*

Over-sampling adalah teknik yang memperbanyak jumlah data pada kelas yang lebih sedikit daripada kelas mayoritas. Untuk memperbanyak jumlah data, dapat menggunakan metode augmentasi data yang akan dijelaskan pada subbab berikutnya. Keuntungan menggunakan *Over-sampling* adalah tidak ada informasi yang hilang saat proses latih. Namun kekurangan yang didapat adalah dapat menyebabkan *Overfitting*.

- *Under-sampling*

Under-sampling adalah teknik mengurangi jumlah data pada kelas yang lebih banyak daripada distribusi kelas yang ada. Keuntungan menggunakan *Under-sampling* adalah mempercepat waktu proses *training* dan menyelesaikan permasalahan memori saat proses *training*. Namun kekurangan yang didapat adalah dapat menghilangkan beberapa informasi penting yang mungkin ada pada gambar yang kita kurangi.

- Teknik *Ensemble Learning* pada Model

Ensemble learning pada model adalah teknik lain yang dapat digunakan untuk menangani kumpulan data yang tidak seimbang. Teknik ini menggabungkan hasil atau kinerja beberapa model untuk mendapatkan peningkatan performa model yang baru. Terdapat berbagai metode yang dapat digunakan untuk mengimplementasi *ensemble learning*, yaitu *Bagging*, *Boosting*, dan lainnya yang telah dibahas pada sub-bab sebelumnya.

2.2.7. Augmentasi Data

Augmentasi data adalah teknik yang dapat dilakukan untuk mengatasi dataset yang jumlah per kelasnya tidak sama. Tujuan penggunaan augmentasi data adalah untuk memperbanyak variasi pada dataset, balancing data, menghindari *overfitting*, serta meningkatkan performa. Banyak teknik yang digunakan untuk augmentasi data seperti meningkatkan kecerahan, memotong gambar, membalikkan gambar secara vertikal maupun *horizontal*, *blur*, *zoom*, *translasi*, dan lain sebagainya.

2.2.8. FastAPI

FastAPI adalah *framework* untuk pengembangan *website* yang dapat digunakan pada Python 3.6 maupun versi terbaru. FastAPI dirilis pada tahun 2018 dan tujuan utamanya adalah membangun aplikasi web dengan cepat dan REST API. FastAPI sekarang digunakan di Uber, Microsoft, Explosion AI dan lainnya [25]. Alasan penulis memilih FastAPI adalah sebagai berikut:

- Performa FastAPI sangat baik bahkan menjadi salah satu *framework* python tercepat.
- Dukungan asinkronus yang membuat penulis dapat menggunakan satu *framework* untuk semua *endpoint*.
- Dokumentasi API yang sangat mudah dibuat karena otomatis dari FastAPI tanpa harus membuat secara manual

2.2.9. NuxtJs

NuxtJs adalah *framework* yang digunakan untuk membuat *Universal VueJS Application* [26]. Selain itu NuxtJs juga merupakan salah satu *framework open source* untuk membangun *website Front-End* yang berbasis VueJs. NuxtJs memudahkan penulis dalam membuat tampilan serta koneksi ke sistem *backend*. NuxtJs memiliki beberapa kelebihan yang menjadi dasar oleh penulis untuk menggunakan *framework* ini untuk penelitian, antara lain:

- Dipercaya memiliki performa lebih baik daripada *framework* SSR lainnya.

- Penggunaan yang mudah, memudahkan penulis untuk mempelajari dan membangun PWA serta dokumentasi yang tertata rapi.
- NuxtJs telah menyediakan *framework* CSS yang membantu dalam pembuatan tampilan.
- Terdapat Nuxt-PWA membantu dalam pengembangan sistem PWA.

2.2.10. Progressive Web App (PWA)

Progressive Web App atau PWA adalah sebuah *website* yang dapat berperilaku layaknya aplikasi *mobile*. PWA mengambil keuntungan dari aplikasi *mobile* tanpa harus mengunduh PWA melalui *app store*. Google menggunakan istilah FIRE (*Fast, Integrated, Reliable, dan Engaging*) pada PWA yang artinya cepat, terintegrasi, dapat diandalkan dan menarik [27]. Google mengatakan seperti tersebut dikarenakan PWA memiliki beberapa kelebihan, yaitu:

- Penghematan biaya produksi karena dengan hanya membuat sistem PWA, dapat berjalan di pada *website* maupun aplikasi *mobile*.
- Dapat digunakan pada berbagai perangkat, Android, iOS, dan Windows.
- Tidak perlu berurusan dengan *App Store* yang membutuhkan biaya registrasi (25 dollar) dan langganan (99 dollar per tahun).
- Performa yang lebih cepat, bahkan Uber mengatakan PWA adalah alasan sistem mereka dapat berjalan di jaringan 2G.
- *Update* otomatis akan sangat membantu pengguna yang jarang membuka *app store*.
- Ramah storage dan juga bandwidth.

2.2.11. Karakteristik Penyakit Tanaman Padi

Padi (*Oryza Sativa*) merupakan tanaman budidaya yang sangat penting bagi manusia di seluruh dunia terutama di negara Asia. Bagi Indonesia sendiri, padi yang diolah menjadi nasi, merupakan makanan pokok sebagai sumber energi. Tidak hanya itu padi juga menjadi komoditas pangan utama di Indonesia [28] dan juga beberapa negara Asia lainnya. Padi sendiri ditanam di area sawah dan memiliki karakteristik tersendiri dalam cara menanamnya.

Padi memiliki karakteristik anatomi yaitu akar, batang, daun, bunga, dan buah. Akar tanaman padi berfungsi dalam penyerapan nutrisi yang akan disalurkan ke bagian padi lainnya melalui batang padi. Batang padi sendiri dapat tumbuh sekitar 107-115 cm. Untuk daunnya, padi memiliki daun bersisik yang terbagi menjadi beberapa bagian yaitu helaian daun, pelepah daun, dan lidah daun. Normalnya daun padi berwarna hijau dengan posisi tegak. Padi juga mempunyai bunga dan buah, yang nantinya buah tersebut akan tertutup lemma dan pelea saat mengalami penyerbukan. Saat dewasa, pelea dan lemma akan membuka serta membentuk sudut yang nantinya akan dipanen dan dikonsumsi sebagai makanan pokok [28].

Namun seperti pada tanaman lainnya, padi juga memiliki berbagai jenis penyakit. Berikut penjelasan mengenai penyakit pada tanaman padi berdasarkan *dataset* yang penulis gunakan untuk penelitian. Informasi mengenai penyakit tanaman padi diambil pada beberapa *website* yang tersedia secara publik [5][29][30][31][32].

Table 2. Karakteristik Penyakit Tanaman Padi

Penyakit	Gejala	Penyebab
Bacterial Leaf Blight [5] 	Pada bibit, daun yang terinfeksi berubah menjadi hijau keabu-abuan dan menggulung. Saat penyakit mulai berkembang, daun menguning dan layu	Adanya gulma yang terinfeksi. Lingkungan tropis dan subtropis terutama di sawah yang kurang diairi air. Penyakit ini berkembang di suhu 25-34 C

<p>Bacterial Leaf Streak [5]</p> 	<p>Daun memiliki garis-garis coklat tetapi tipis. Jika sudah parah, seluruh daun bisa berwarna coklat</p>	<p>Muncul di area dengan temperatur dan kelembapan tinggi. Biasanya muncul di daerah tropis dan subtropis</p>
<p>Bacterial Panicle Blight [32]</p> 	<p>Malai mengalami pembusukan, perubahan warna coklat muda pada sepertiga bagian bawah hingga setengah malai.</p>	<p>Karena panas, cuaca kering yang membantu bakteri berkembang</p>
<p>Blast [5]</p> 	<p>Daun memiliki bercak elips dan bagian tengah berwarna keputihan hingga abu-abu dengan batas merah hingga kecoklatan</p>	<p>Terjadi di daerah dengan tanah kurang lembap, curah hujan tinggi, suhu dingin di siang hari. Embun juga mendukung perkembangan penyakit</p>

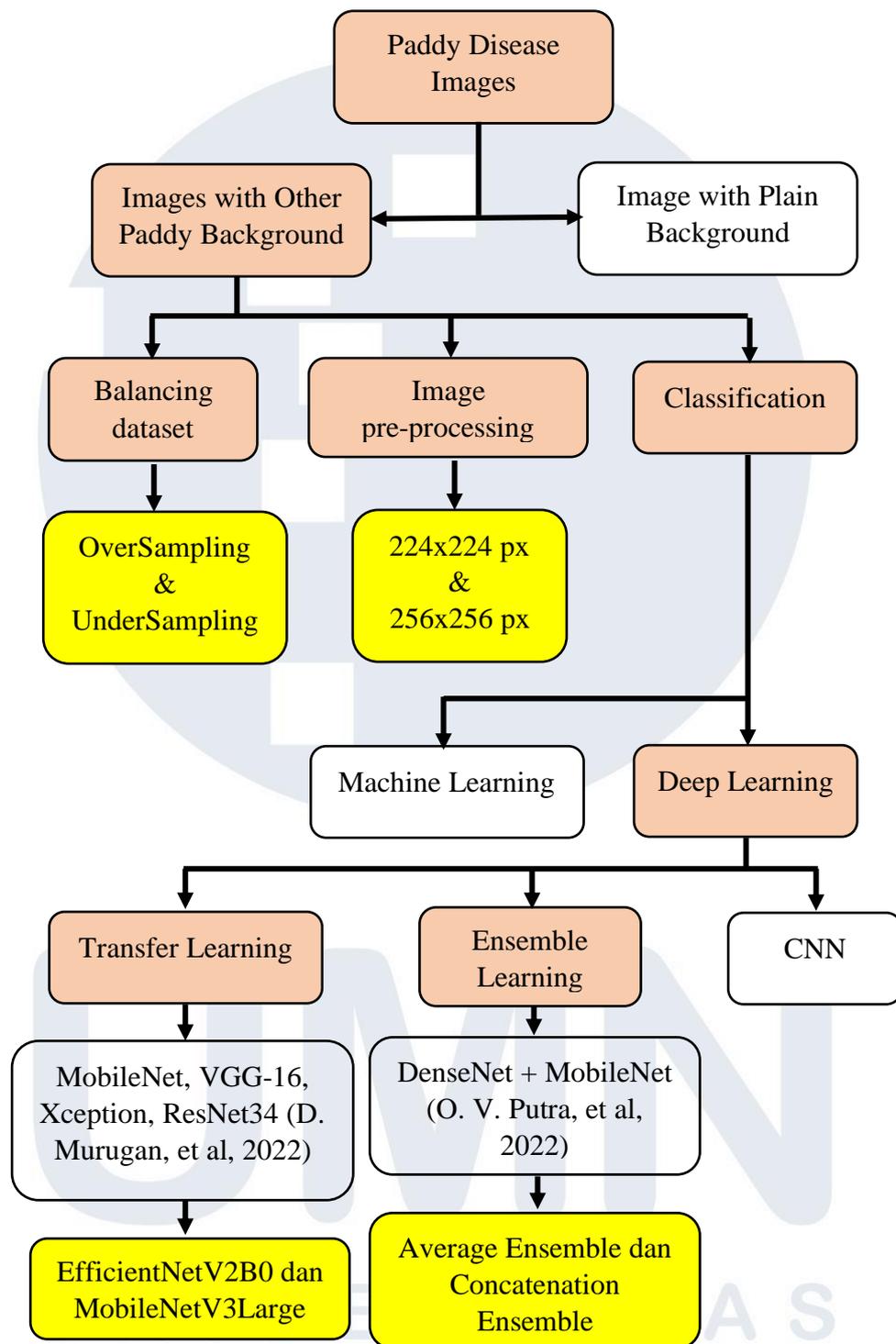
<p>Brown Spot [5]</p> 	<p>Daun memiliki bercak lingkaran hingga lonjong dengan bagian tengah berwarna coklat muda hingga abu-abu, dikelilingi tepi berwarna coklat kemerahan</p>	<p>Muncul di area dengan kelembapan (86% - 100%) dan suhu (16 – 36 C). Tanah yang tidak tergenang air dan kekurangan nutrisi</p>
<p>Dead Heart [31]</p> 	<p>Malai berwarna keputihan dan tidak berisi. Lubang kecil pada batang</p>	<p>Larva hama pelubang batang padi</p>
<p>Downy Mildew [29]</p> 	<p>Bintik kuning pada lapisan daun atas yang menyebar ke beberapa titik pada daun</p>	<p>-</p>
<p>Hispa [30]</p> 	<p>Pengikisan permukaan atas helai daun menghasilkan guratan putih. Daun keputihan dan berselaput</p>	<p>Adanya hama hispa, hujan lebat, curah hujan rendah, perbedaan suhu malam siang kecil, menguntungkan hama</p>

<p>Tungro [5]</p> 	<p>Daun berubah warna kuning / oranye dimulai dari ujung daun sampai bawah daun. Juga dapat menunjukkan bintik-bintik berwarna karat. Tanaman dapat menunjukkan gejala kerdil dan pembungaan tertunda.</p>	<p>Menular dari tanaman lain saat proses pemindahan bibit, persemaian, pembajakan dan proses lainnya yang kurang tepat</p>
---	--	--

2.3 Summary

Berdasarkan penelitian terdahulu dan tinjauan teori yang penulis analisa, maka rancangan penelitian yang akan penulis lakukan adalah sebagai berikut:

- Dataset akan menggunakan dataset publik dari Paddy Doctor [9]. Dataset tersebut bersifat *imbalanced*, sehingga penulis akan menggunakan teknik undersampling dan oversampling sebagai tahap *data balancing* dataset.
- Sistem akan menggunakan teknik *deep learning* yang akan menggunakan *pre-trained model* dan *ensemble learning*. Penggunaan *pre-trained model* terbukti mampu memberikan performa yang baik dan tidak perlu merancang sistem model dari awal. Penggunaan *ensemble learning* juga terbukti mampu memberikan performa lebih baik dari *pre-trained model* dengan cara menggabungkan dua atau lebih model.
- Analisa performa setiap model akan menggunakan bantuan grafik akurasi dan *loss*, *classification report*, dan *confusion matrix*.
- Model yang telah dilatih akan diimplementasi pada FastAPI yang merupakan satu kesatuan sistem PWA yang akan penulis buat dengan pemilihan NuxtJS sebagai *user interface* dan konfigurasi PWA.
- Analisa rata-rata waktu prediksi model pada sistem PWA akan dibantu oleh *network waterfall* serta menggunakan analisa korelasi antara rata-rata waktu prediksi model dengan ukuran *file h5* model.



Gambar 2.4 Diagram state of the art