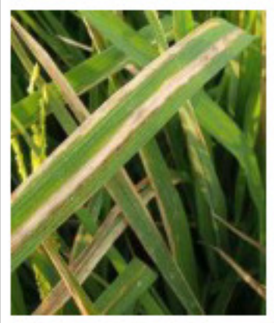
BAB II
KAJIAN LITERATUR

2.1 Penyakit Daun Padi

Berbagai jenis penyakit dapat menyerang daun padi, diantaranya *Brown Spot*, *Blast Disease*, *Bacterial Leaf Blight* yang disebabkan oleh jamur maupun bakteri. Penyakit-penyakit tersebut menyebabkan penurunan luas area fotosintesis aktif, perubahan warna daun, hingga kematian jaringan. Identifikasi yang tepat mengenai karakteristik penyebab dan gejala visual sangat penting karena setiap penyakit memerlukan penanganan yang berbeda. Ringkasan mengenai perbedaan karakteristik utama dari ketiga penyakit tersebut disajikan dalam Tabel 2.1.

Tabel 2.1 Karakteristik Penyakit Daun Padi

Jenis Penyakit	**Penyebab**	**Gejala Utama pada Daun**	**Ilustrasi Penyakit pada Daun Padi**	**Dampak terhadap Tanaman**
Bercak Daun Coklat (*Brown Spot*)	*Helminthosporium oryzae (sin. Bipolaris oryzae)*	Bercak berbentuk oval atau bulat berwarna cokelat tua, yang kemudian bagian tengahnya berubah menjadi abu-abu atau keputihan.		Menghambat perkecambahan, mematikan bibit, serta menurunkan luas area fotosintesis dan kualitas gabah
Blast (*Blast Disease*)	*Magnaporthe oryzae* (sin. *Pyricularia oryzae*)	Bercak penyakit berbentuk belah ketupat dengan ujung meruncing, pinggir cokelat, tengah abu-abu atau keputihan, serta sering berisi serbuk putih berupa spora jamur yang siap menyebar.		Kematian jaringan daun dan patah leher malai (*neck blast*).

Hawar Daun Bakteri (*Bacterial Leaf Blight*)	*Xanthomonas oryzae* pv. *oryzae*	Garis basah di tepian daun yang berubah menjadi kuning bergelombang, lalu memutih/abu-abu		Penurunan luas fotosintesis secara masif hingga kematian tanaman (kresek).

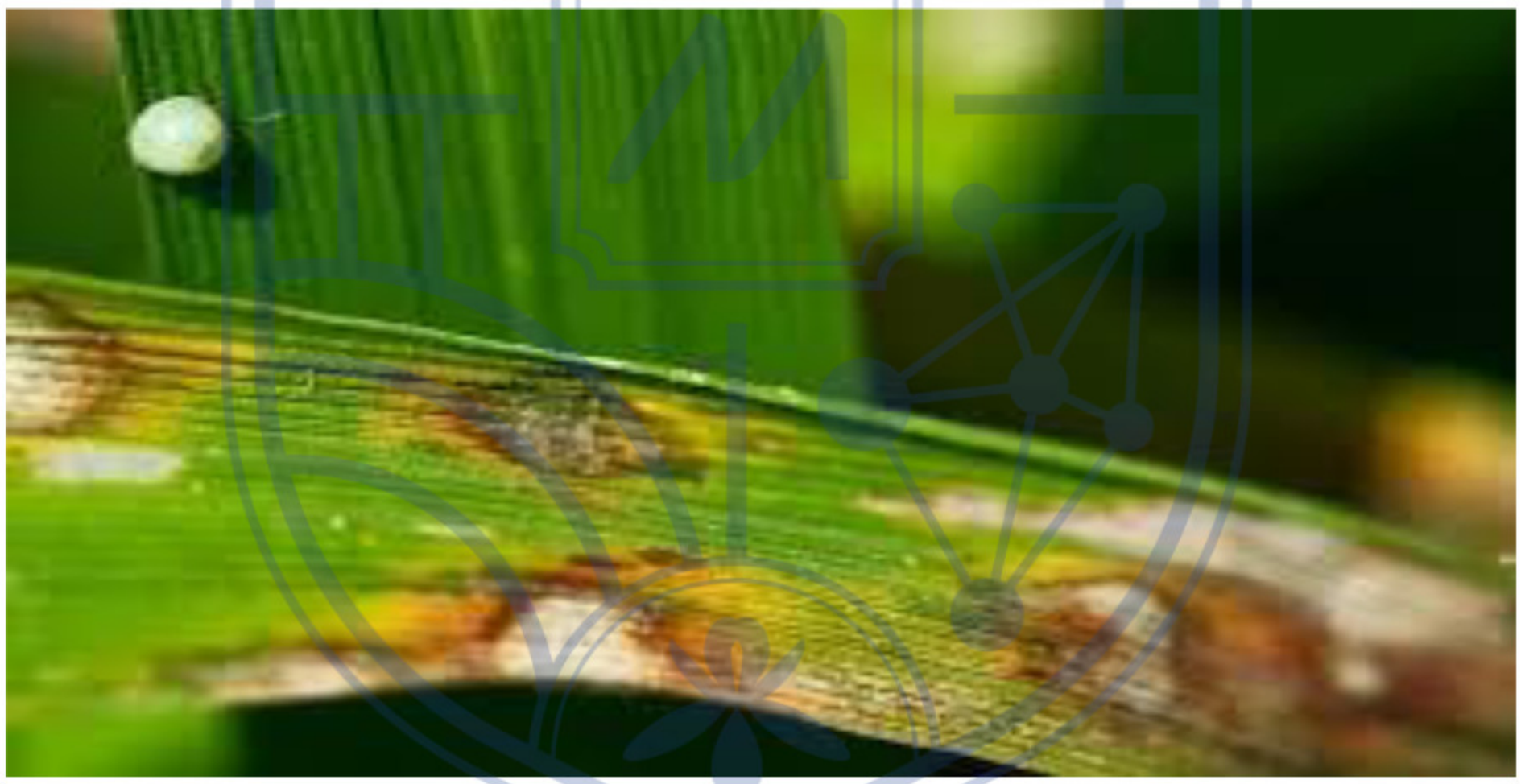
1. Bercak Daun Coklat (*Brown Spot*)

Bercak daun coklat (*Brown Spot*) pada padi yang disebabkan oleh *Cochliobolus miyabeanus* diketahui terjadi di Jepang sejak tahun 1900. Penyakit ini juga disebut sebagai layu bibit, bercak daun wijen, dan *Helminthosporium*. Penyakit ini lebih parah pada padi yang ditanam kering atau langsung. Penyakit ini terutama terjadi dilingkungan dimana pasukan air langka dikombinasikan dengan ketidakseimbangan nutrisi, khususnya kekurangan nitrogen [10]. Penyakit bercak coklat yang disebabkan oleh jamur adalah infeksi yang paling umum dan muncul sebagai lesi berbentuk oval dan bulat pada daun padi. Jika tidak ditangani tepat waktu, penyakit ini dapat mengakibatkan kehilangan hasil panen yang serius . Gejala khas bercak coklat pada daun dan lembaran bunga meliputi lesi berwarna coklat kemerahan ringan atau lesi dengan pusat abu-abu yang dikelilingi oleh margin coklat tua hingga kemerahan, dan kemudian dikelilingi oleh bercak kuning cerah. Bercak memiliki bentuk dan ukuran khas pada bagian tanaman yang berbeda tergantung pada kondisi lingkungan [11].

Gambar 2.1 Penyakit Daun Padi Bercak Daun Coklat (Brown Spot) [10]

2. Blast (Blast Disease)

Penyakit *Blast* (*Riceblast*) disebabkan oleh jamur *Pyricularia oryzae*. Penyakit ini bersifat endemik di beberapa sentra padi Indonesia, seperti Jawa Barat, yang artinya serangan terus terjadi setiap tahun bahkan saat cuaca kemarau panjang atau fenomena *El Niño* sekalipun. Gejala fisik penyakit ini pada daun sangat khas dan mudah dikenali untuk keperluan deteksi visual. Bercak penyakit berbentuk seperti belah ketupat (*diamond-shaped*) dengan ujung yang meruncing. Bagian pinggir bercak berwarna cokelat, sedangkan bagian tengahnya berwarna abu-abu atau keputihan. Pada bagian tengah lesi tersebut sering terlihat adanya serbuk putih, yang sebenarnya adalah kumpulan spora jamur yang siap menyebar [12].



Gambar 2.2 Penyakit Daun Padi Blast (Blast Disease) [12]

3. Hawar Daun Bakteri (*Bacterial Leaf Blight*)

Penyakit hawar daun bakteri merupakan penyakit yang tersebar luas di pertanaman padi sawah dan bisa menurunkan hasil sampai 36%. Penyakit ini pada umumnya terjadi pada musim hujan atau lembab >75%, terutama pada lahan sawah yang selalu tergenang dengan pemupukan N yang tinggi. Hawar daun bakteri merupakan penyakit yang disebabkan oleh bakteri *Xanthomonas campestris pv. Oryzae Dye*. yang dapat menginfeksi tanaman padi pada berbagai stadium pertumbuhan [13]. Deteksi dini terhadap penyakit-penyakit tersebut sangat

penting agar dapat dilakukan pengendalian secara cepat dan tepat. Pendekatan berbasis teknologi pengolahan citra dan kecerdasan buatan (*AI*) menjadi solusi efektif untuk mendukung deteksi penyakit tanaman secara otomatis dan akurat [5].

Gambar 2.3 Penyakit Daun Padi Hawar Daun Bakteri (Bacterial Leaf Blight) [13]

2.2 Deteksi Citra Berbasis *Deep Learning*

Deteksi citra merupakan proses untuk mengidentifikasi dan mengklasifikasikan objek tertentu dalam suatu gambar. Pada bidang pertanian digital, deteksi citra digunakan untuk mengenali kondisi daun tanaman berdasarkan pola warna, tekstur, dan bentuk yang muncul akibat penyakit. Metode tradisional pengolahan citra, seperti *thresholding*, *edge detection*, atau *color segmentation*, memiliki keterbatasan dalam mengenali pola kompleks. Oleh karena itu, pendekatan *Deep Learning* khususnya *Convolutional Neural Network* (CNN) menjadi pilihan utama karena kemampuannya mengekstraksi fitur visual secara otomatis dan terstruktur [4].

Deep learning adalah metode *machine learning* yang melibatkan sejumlah besar lapisan dalam jaringan saraf tiruan, yang digunakan untuk mengolah data secara berurutan. *Deep learning* merupakan subbagian dari *machine learning* yang memanfaatkan *neural network* untuk menyelesaikan berbagai masalah [14]. *Deep learning* memiliki keunggulan dalam belajar langsung dari data visual melalui proses training pada dataset besar. Dalam konteks pertanian, CNN menjadi arsitektur paling banyak digunakan. CNN mampu mengekstrak fitur sederhana seperti tepi hingga fitur kompleks seperti pola bercak penyakit daun. Berbagai penelitian pada penyakit tanaman seperti singkong, gandum, jagung, dan

padi menunjukkan bahwa *deep learning* memberikan akurasi sangat tinggi bahkan pada kondisi lingkungan lapangan yang tidak terkontrol [15].

Model *deep learning* sangat relevan untuk deteksi penyakit daun padi karena bercak penyakit memiliki variasi bentuk, ukuran, dan warna. Penggunaan *deep learning* memungkinkan sistem beradaptasi dengan variasi tersebut tanpa perlu pembuatan fitur manual. Selain itu, *deep learning* juga dapat digunakan tidak hanya untuk klasifikasi, tetapi juga untuk deteksi objek yang menyediakan informasi lokasi bercak pada daun.

2.3 Model Object Detection

Object detection merupakan salah satu bidang pada *computer vision* yang bertujuan untuk mendeteksi keberadaan objek sekaligus menentukan lokasi objek tersebut di dalam citra menggunakan *bounding box*. Berbeda dengan klasifikasi citra yang hanya menghasilkan informasi mengenai kelas objek, *object detection* mampu memberikan informasi mengenai jenis objek serta posisi objek secara bersamaan. Kemampuan tersebut menjadikan *object detection* banyak diterapkan pada berbagai bidang, seperti sistem pengawasan, kendaraan otonom, robotika, kesehatan, serta pertanian cerdas [15].

Secara umum, algoritma object detection dibedakan menjadi dua kelompok, yaitu *two-stage detector dan single-stage detector*. *Two-stage detector*, seperti *Faster* R-CNN, melakukan proses pencarian wilayah objek terlebih dahulu sebelum proses klasifikasi sehingga memiliki tingkat akurasi yang tinggi, tetapi membutuhkan waktu inferensi yang lebih lama. Sebaliknya, *single-stage detector*, seperti YOLO dan SSD, melakukan proses klasifikasi dan lokalisasi objek dalam satu kali proses inferensi sehingga mampu menghasilkan kecepatan deteksi yang lebih tinggi. Oleh karena itu, pendekatan single-stage detector lebih sesuai digunakan pada aplikasi real-time, terutama pada perangkat bergerak yang memiliki keterbatasan sumber daya komputasi [16].

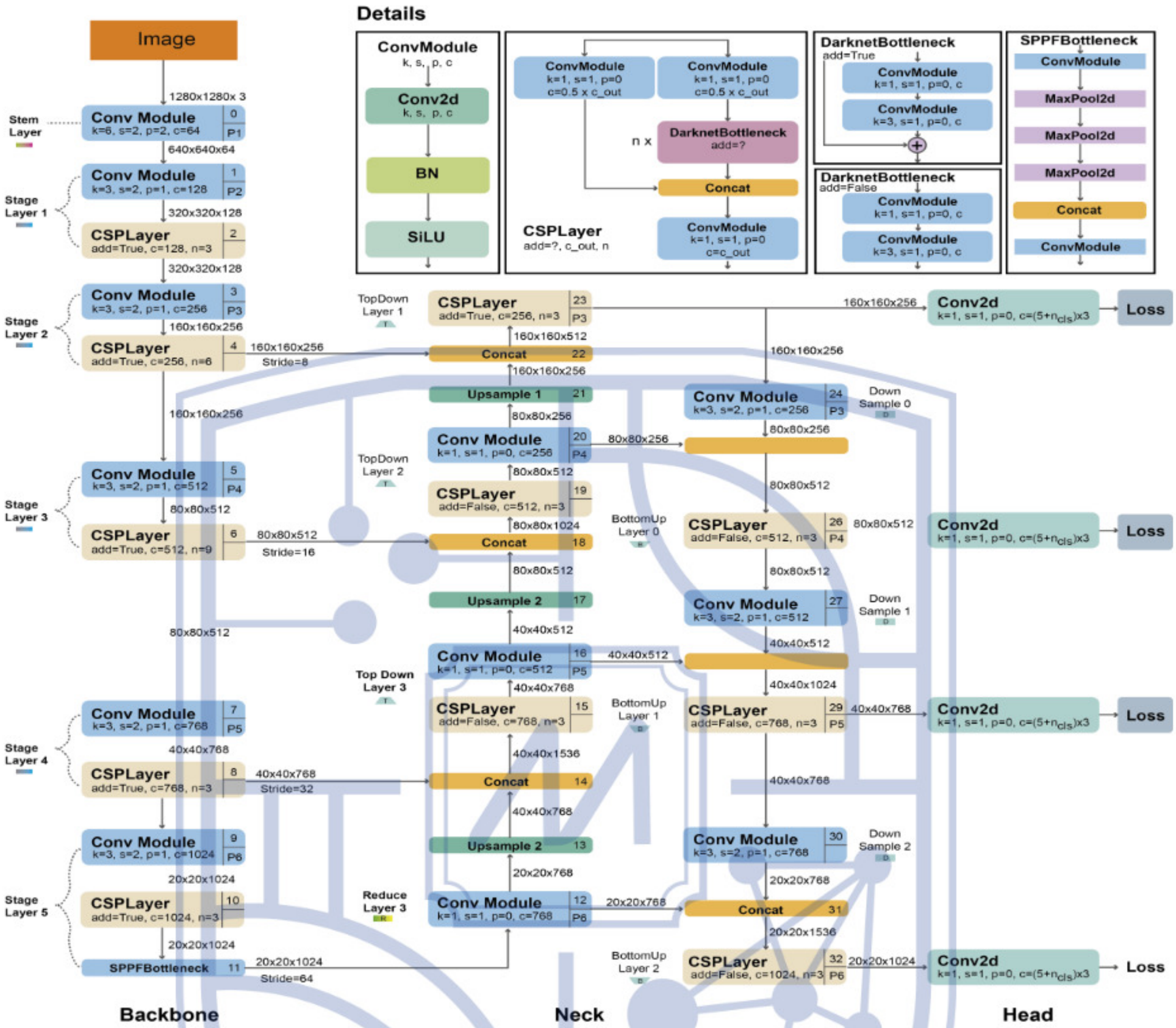
Sebaliknya, *single-stage detector*, seperti SSD dan YOLO, melakukan proses klasifikasi dan lokalisasi objek secara bersamaan dalam satu kali proses inferensi sehingga memiliki kecepatan deteksi yang jauh lebih tinggi. Karakteristik tersebut menjadikan *single-stage detector* lebih sesuai untuk aplikasi *real-time*, termasuk implementasi pada perangkat *mobile* yang memiliki keterbatasan sumber daya komputasi [17]. Selain algoritma deteksi, performa model *object detection* juga dipengaruhi oleh arsitektur *backbone* yang digunakan pada proses ekstraksi fitur. *Backbone* berperan dalam mengekstraksi informasi visual penting dari citra, seperti warna, tekstur, bentuk, dan pola objek. Pada penelitian ini, digunakan *EfficientNet-B4* sebagai *backbone* karena memiliki kemampuan ekstraksi fitur

yang baik dengan jumlah parameter yang relatif efisien dibandingkan arsitektur CNN konvensional [18]. YOLO memiliki performa yang baik dalam mendeteksi penyakit tanaman, termasuk penyakit daun padi, terutama dalam mengenali objek berukuran kecil dan kompleks. Kombinasi antara algoritma YOLO dan *backbone EfficientNet-B4* diharapkan mampu menghasilkan model *object detection* yang tidak hanya memiliki akurasi yang baik, tetapi juga tetap efisien dari sisi komputasi. Dengan kemampuan tersebut, model diharapkan mampu mendeteksi penyakit daun padi secara tepat sehingga dapat mendukung implementasi pada aplikasi berbasis Android [19].

Pada penelitian ini dipilih algoritma YOLOv3 sebagai model object detection karena mampu menghasilkan keseimbangan antara kecepatan inferensi dan akurasi deteksi. Selain itu, performa YOLOv3 didukung oleh penggunaan EfficientNet-B4 sebagai backbone yang berfungsi mengekstraksi fitur visual penting dari citra daun padi. Kombinasi kedua arsitektur tersebut diharapkan mampu meningkatkan kemampuan sistem dalam mengenali penyakit daun padi sekaligus menentukan lokasi lesi penyakit secara akurat melalui informasi bounding box.

2.4 Algoritma YOLO

You Only Look Once (YOLO) merupakan algoritma *single-stage object detection* yang melakukan proses klasifikasi dan lokalisasi objek secara bersamaan dalam satu kali proses inferensi. Berbeda dengan metode *two-stage detector* yang memisahkan proses pencarian objek dan klasifikasi, YOLO memproses seluruh citra secara langsung menggunakan satu jaringan saraf sehingga mampu menghasilkan deteksi dengan kecepatan tinggi tanpa mengurangi akurasi secara signifikan. Oleh karena itu, YOLO banyak digunakan pada berbagai aplikasi yang membutuhkan deteksi objek secara *real-time*, seperti sistem pengawasan, kendaraan otonom, robotika, pertanian cerdas, dan diagnosis penyakit tanaman [20]. Pada penelitian ini, algoritma YOLO digunakan untuk mendeteksi penyakit daun padi berdasarkan citra yang diperoleh dari kamera maupun galeri perangkat Android. Proses deteksi dimulai dengan menerima citra masukan yang kemudian diproses oleh *EfficientNet-B4* sebagai *backbone* untuk mengekstraksi fitur-fitur visual penting, seperti warna daun, tekstur, bentuk bercak, dan pola lesi penyakit. *Feature* map yang dihasilkan *backbone* selanjutnya diteruskan ke YOLO *Detection Head* untuk menghasilkan prediksi berupa kelas penyakit, nilai *confidence score*, serta koordinat *bounding box* yang menunjukkan lokasi penyakit pada daun.



Gambar 2. 4 Algoritma Yolo [21]

Algoritma YOLO terdiri atas dua komponen utama yang digunakan dalam penelitian ini, yaitu *backbone* dan *head*. *Backbone* berfungsi untuk melakukan ekstraksi fitur dari citra masukan dengan mempelajari informasi visual seperti warna, tekstur, bentuk, dan pola objek. *Feature map* yang dihasilkan oleh *backbone* kemudian diteruskan ke bagian *head* untuk menghasilkan prediksi akhir berupa kelas objek, *confidence score*, dan koordinat *bounding box*. Kombinasi kedua komponen tersebut memungkinkan model melakukan klasifikasi dan lokalisasi objek secara bersamaan dalam satu kali proses inferensi.

1. Backbone merupakan bagian awal dari arsitektur YOLO yang bertugas melakukan ekstraksi fitur dari citra masukan. Komponen ini tersusun atas beberapa lapisan konvolusi yang mampu mempelajari informasi visual penting, seperti tekstur, warna, bentuk, tepi, dan pola objek pada berbagai tingkat abstraksi. *Feature map* yang dihasilkan oleh *backbone* diteruskan ke YOLO *Detection Head* untuk menghasilkan

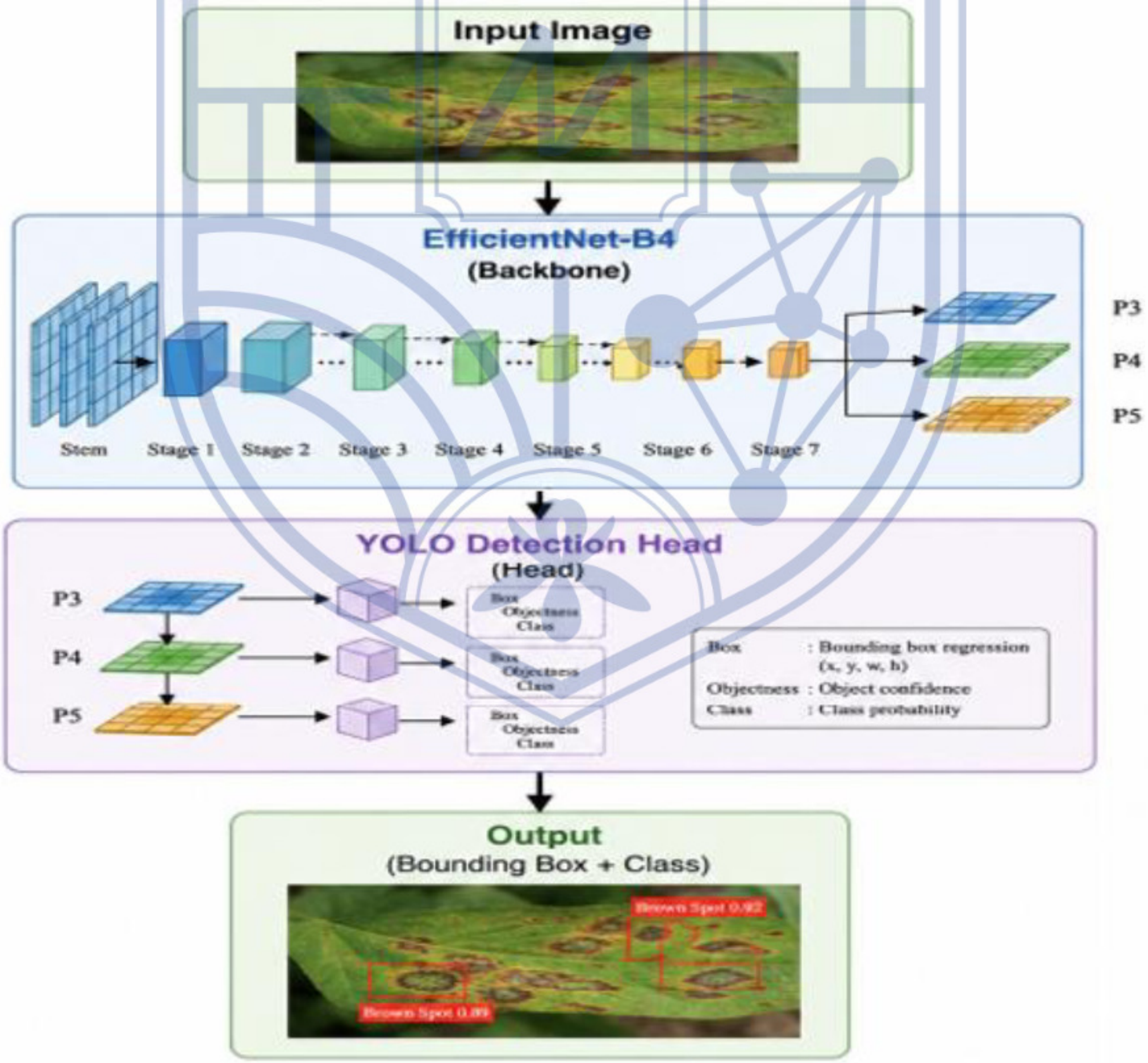
prediksi objek. Kualitas *feature map* yang dihasilkan *backbone* sangat memengaruhi performa deteksi objek secara keseluruhan [20].

2. Neck merupakan komponen yang berada di antara *backbone* dan *detection head* pada arsitektur *object detection*. Fungsi utama *neck* adalah menggabungkan *feature map* yang dihasilkan oleh *backbone* pada berbagai tingkat resolusi sehingga informasi fitur menjadi lebih kaya sebelum diteruskan ke *detection head*. Pada beberapa arsitektur YOLO, neck umumnya dibangun menggunakan *Feature Pyramid Network* (FPN), *Path Aggregation Network* (PAN), atau kombinasi keduanya untuk meningkatkan kemampuan model dalam mendeteksi objek dengan ukuran yang bervariasi. Pada penelitian ini, komponen neck tidak digunakan. *Feature map* yang dihasilkan oleh *backbone EfficientNet-B4* diteruskan secara langsung ke YOLO *Detection Head* tanpa melalui proses penggabungan fitur menggunakan FPN maupun PAN. Pendekatan ini dipilih untuk menyederhanakan arsitektur model sehingga jumlah parameter dan kompleksitas komputasi menjadi lebih rendah. Dengan arsitektur yang lebih sederhana, proses inferensi pada perangkat Android menjadi lebih cepat dan penggunaan memori lebih efisien, sehingga sesuai dengan tujuan penelitian yang berfokus pada implementasi sistem deteksi penyakit daun padi berbasis *mobile* [20],[21] .
3. *Head* merupakan komponen akhir yang bertugas menghasilkan prediksi *bounding box*, *confidence score*, dan klasifikasi objek berdasarkan *feature map* yang dihasilkan *backbone*. Bagian ini melakukan dua tugas utama, yaitu klasifikasi objek dan regresi *bounding box*. Selain itu, *head* juga menghasilkan nilai *confidence score* yang menunjukkan tingkat keyakinan model terhadap keberadaan objek pada lokasi tertentu. Melalui komponen ini, model dapat menentukan jenis penyakit sekaligus lokasi area yang terinfeksi pada citra daun padi [20], [21].
4. *Non-Maximum Suppression* (NMS) merupakan tahap pascaproses yang umum digunakan pada sistem *object detection* untuk menghilangkan prediksi *bounding box* yang saling tumpang tindih. Algoritma ini bekerja dengan mempertahankan *bounding box* yang memiliki nilai *confidence score* tertinggi dan menghapus prediksi lain yang memiliki tingkat kemiripan tinggi berdasarkan nilai *Intersection over Union* (IoU). Dengan menerapkan NMS, model dapat menghasilkan deteksi objek yang lebih akurat dan mengurangi kemunculan prediksi ganda pada objek yang sama [20], [21]. Secara umum, arsitektur YOLO menggunakan *Non-Maximum Suppression* (NMS) untuk menyaring hasil deteksi. Akan tetapi, pada penelitian ini proses evaluasi model dilakukan tanpa menerapkan NMS secara *eksplisit*. Prediksi terbaik pada setiap citra dipilih

berdasarkan nilai *confidence score* tertinggi menggunakan pendekatan *single best prediction per image*.

2.5 Arsitektur *YOLO dengan Backbone EfficientNet-B4*

EfficientNet-B4 merupakan salah satu arsitektur *Convolutional Neural Network (CNN)* yang banyak digunakan dalam berbagai tugas pengolahan citra karena memiliki kemampuan ekstraksi fitur yang tinggi dengan kebutuhan komputasi yang relatif efisien [22]. Pada penelitian ini, *EfficientNet-B4* digunakan sebagai *backbone* pada arsitektur YOLO. *Backbone* bertugas mengekstraksi fitur dari citra daun padi sehingga diperoleh representasi visual berupa warna, tekstur, bentuk, dan pola lesi penyakit. *Feature map* yang dihasilkan oleh *EfficientNet-B4* kemudian diproses oleh YOLO *Detection Head* untuk menghasilkan prediksi lokasi objek, *confidence score*, dan kelas penyakit. Arsitektur model terdiri atas empat tahapan utama, yaitu input image, backbone *EfficientNet-B4*, YOLO *Detection Head*, dan *output detection*.



Gambar 2. 5 Arsitektur EfficientNet-B4 dengan YOLO Hybrid

1. Input Image

Tahap awal pada arsitektur ini adalah pemberian citra masukan berupa citra daun padi yang telah melalui proses *preprocessing*, yaitu *image resizing* menjadi ukuran 380 × 380 piksel dan normalisasi nilai piksel menggunakan parameter *mean* dan *standard deviation* dari *ImageNet*. Citra masukan kemudian diteruskan ke *backbone EfficientNet-B4* untuk dilakukan proses ekstraksi fitur. Kualitas citra masukan sangat memengaruhi kemampuan model dalam mengenali pola penyakit pada daun padi [23].

2. EfficientNet-B4 Backbone

EfficientNet-B4 digunakan sebagai *backbone* utama untuk melakukan ekstraksi fitur dari citra daun padi. Arsitektur ini tersusun atas beberapa blok *Mobile Inverted Bottleneck Convolution* (MBConv) yang mampu mengekstraksi informasi visual penting, seperti tekstur, warna, bentuk, serta pola lesi penyakit. Pada penelitian ini, *EfficientNet-B4* menggunakan bobot awal (*pre-trained weights*) dari *ImageNet* untuk membantu model dalam mempelajari karakteristik citra secara lebih efektif. Feature map yang dihasilkan oleh backbone kemudian diteruskan secara langsung ke YOLO *Detection Head* untuk proses deteksi objek [22], [24].

3. YOLO Detection Head

Hasil dari *feature map* diteruskan ke YOLO *Detection Head* untuk menghasilkan prediksi akhir berupa kelas penyakit dan lokasi objek pada citra. *Detection head* terdiri atas dua proses utama, yaitu klasifikasi objek dan regresi *bounding box*.

a. Classification Head

Classification Head bertujuan menentukan kelas penyakit berdasarkan feature map dengan menghasilkan probabilitas setiap kelas penyakit, seperti *Brown Spot*, *Blast Disease*, *Bacterial Leaf Blight*, maupun *Healthy*, beserta nilai *confidence score* yang menunjukkan tingkat keyakinan model terhadap hasil prediksi.

b. Bounding Box Regression Head

Bounding Box Regression Head bertugas memprediksi lokasi objek penyakit pada citra daun padi. Keluaran dari bagian ini berupa koordinat *bounding box* yang menunjukkan posisi lesi penyakit secara spesifik pada citra sehingga memudahkan pengguna dalam mengidentifikasi area daun yang terinfeksi penyakit [20], [21].

Pada arsitektur *object detection*, *EfficientNet-B4* digunakan sebagai *backbone* karena memiliki kemampuan ekstraksi fitur yang baik dengan jumlah parameter yang relatif efisien. Penggunaan *EfficientNet-B4* dapat membantu model dalam mengenali karakteristik visual penyakit daun padi yang memiliki variasi warna, bentuk, dan tekstur. Selain itu, lesi penyakit pada daun padi umumnya berukuran kecil dan tersebar pada bagian tertentu dari daun. Maka diperlukan *backbone* yang mampu menghasilkan representasi fitur yang lebih detail agar proses deteksi dapat dilakukan dengan lebih akurat. Hasil ekstraksi fitur dari *EfficientNet-B4* kemudian diteruskan ke YOLO *Detection Head* untuk melakukan proses klasifikasi penyakit dan prediksi *bounding box*. Melalui kombinasi tersebut model tidak hanya mampu menentukan jenis penyakit, tetapi juga dapat menunjukkan lokasi area daun yang terinfeksi.

2.6 Deteksi Berbasis Mobile

Deteksi penyakit berbasis *mobile* menjadi kebutuhan penting dalam pertanian modern. meningkatnya penggunaan *smartphone* oleh petani membuat aplikasi *mobile* menjadi solusi yang efektif. *Smartphone* memungkinkan deteksi penyakit dilakukan secara *real-time* di lokasi lahan tanpa memerlukan perangkat mahal seperti laptop atau server berbasis GPU [7].

Penelitian *Nguyen et al*. menunjukkan bahwa model *EfficientNet* yang dikonversi ke *TensorFlow Lite* dapat berjalan dalam waktu inferensi rata-rata 1.7 detik pada perangkat Android tanpa penurunan akurasi signifikan, membuktikan kelayakan deteksi berbasis *mobile* dalam pertanian cerdas [6]. Studi lain dalam bidang *smart agriculture* juga menunjukkan bahwa aplikasi *mobile* meningkatkan akses informasi petani serta mempercepat proses pengambilan keputusan, terutama terkait pengendalian penyakit tanaman [5].

Selain itu, deteksi berbasis *mobile* memungkinkan monitoring intensif di lapangan meskipun jaringan internet terbatas, karena model dapat berjalan secara *offline* pada perangkat [5], [6]. Hal ini sangat relevan untuk daerah pedesaan di Indonesia yang belum sepenuhnya memiliki koneksi stabil. Deteksi berbasis *mobile* yang memungkinkan pemantauan secara terus menerus di lapangan meskipun jaringan internet terbatas, karena model dapat berjalan secara *offline* pada perangkat. Sebagai kesimpulan, penggunaan aplikasi *mobile* sangat cocok untuk deteksi penyakit padi karena memberikan kemudahan bagi petani untuk mendapatkan hasil analisis penyakit pada padi.

2.7 Metrik Evaluasi

Evaluasi performa model deteksi penyakit daun padi menggunakan beberapa metrik utama. *Precision* mengukur ketepatan prediksi positif model (1), sementara *Recall* mengukur kemampuan model menangkap seluruh objek penyakit yang ada (2). Keduanya penting karena penyakit daun padi seringkali memiliki pola bercak yang kompleks sehingga model perlu menyeimbangkan tingkat kesalahan *false positive* dan *false negative*.

F1-Score merupakan harmonisasi antara *Precision* dan *Recall*, yang bermanfaat untuk menilai performa keseluruhan model (3). Metrik paling digunakan dalam deteksi objek adalah *mean Average Precision* (mAP) yang mengukur akurasi lokasi *bounding box* dan klasifikasi pada berbagai *threshold Intersection over Union* (IoU).

Pada penelitian *YOLOv7-tiny* terhadap penyakit daun padi, mAP digunakan sebagai metrik utama untuk menilai performa model deteksi dan membandingkan kualitasnya dengan YOLOv9 dan YOLOv10n. Begitu juga pada penelitian *EfficientNet-B4* terhadap FAW dan penyakit tanaman lain, metrik mAP menjadi indikator utama kinerja model deteksi ringan [25].

$$Precision = \frac{TP}{TP + FP} \tag{2. 1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2. 2}$$

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{2. 3}$$

$$mAP = \frac{1}{C} \sum_{C=1}^{C} AP_c \tag{2. 4}$$

1. *Precision* = Mengukur ketepatan prediksi positif model. Semakin tinggi, semakin sedikit prediksi salah (*false positive*).
2. *Recall* = Mengukur kemampuan model menemukan seluruh kasus penyakit yang ada. Semakin tinggi, semakin sedikit kasus yang terlewat (*false negative*).
3. F1-*Score* = Merupakan rata-rata harmonik *Precision* dan *Recall* untuk menilai performa model secara seimbang.

4. mAP (*mean Average Precision*) = Digunakan dalam deteksi objek untuk mengukur akurasi model baik dalam klasifikasi maupun lokasi *bounding box* pada berbagai ambang IoU (*Intersection over Union*).