

BAB II KAJIAN LITERATUR

2.1 Computer Vision

Computer vision merupakan disiplin ilmu yang mengeksplorasi pengembangan algoritma dan teknik agar sistem komputasi mampu menginterpretasikan informasi dari data visual, baik berupa citra statis maupun aliran video. Berbeda dengan pengolahan citra digital yang fokus pada transformasi piksel ke piksel, *computer vision* bertujuan untuk melakukan inferensi tingkat tinggi guna mereplikasi kompleksitas sistem visual manusia [18]. Dalam implementasinya, bidang ini melibatkan proses akuisisi, pemrosesan, dan analisis data multidimensi untuk menghasilkan informasi numerik atau simbolik yang mendukung pengambilan keputusan otonom. Transformasi fundamental dalam *computer vision* terjadi seiring dengan pergeseran dari metode ekstraksi fitur tradisional yang bersifat manual (*hand-crafted features*) menuju pendekatan pembelajaran mendalam (*deep learning*). Arsitektur berbasis *convolutional neural networks* (CNN) telah menjadi standar emas karena kemampuannya dalam mengekstraksi representasi fitur hierarkis, mulai dari deteksi tepi yang sederhana hingga identifikasi tekstur dan bentuk yang kompleks secara otomatis [19]. Di sektor agrikultur, penerapan visi komputer memungkinkan standarisasi mutu hasil panen melalui sistem pemeringkatan (*fruit grading system*) yang mampu bekerja konsisten melampaui limitasi fisik manusia [20].

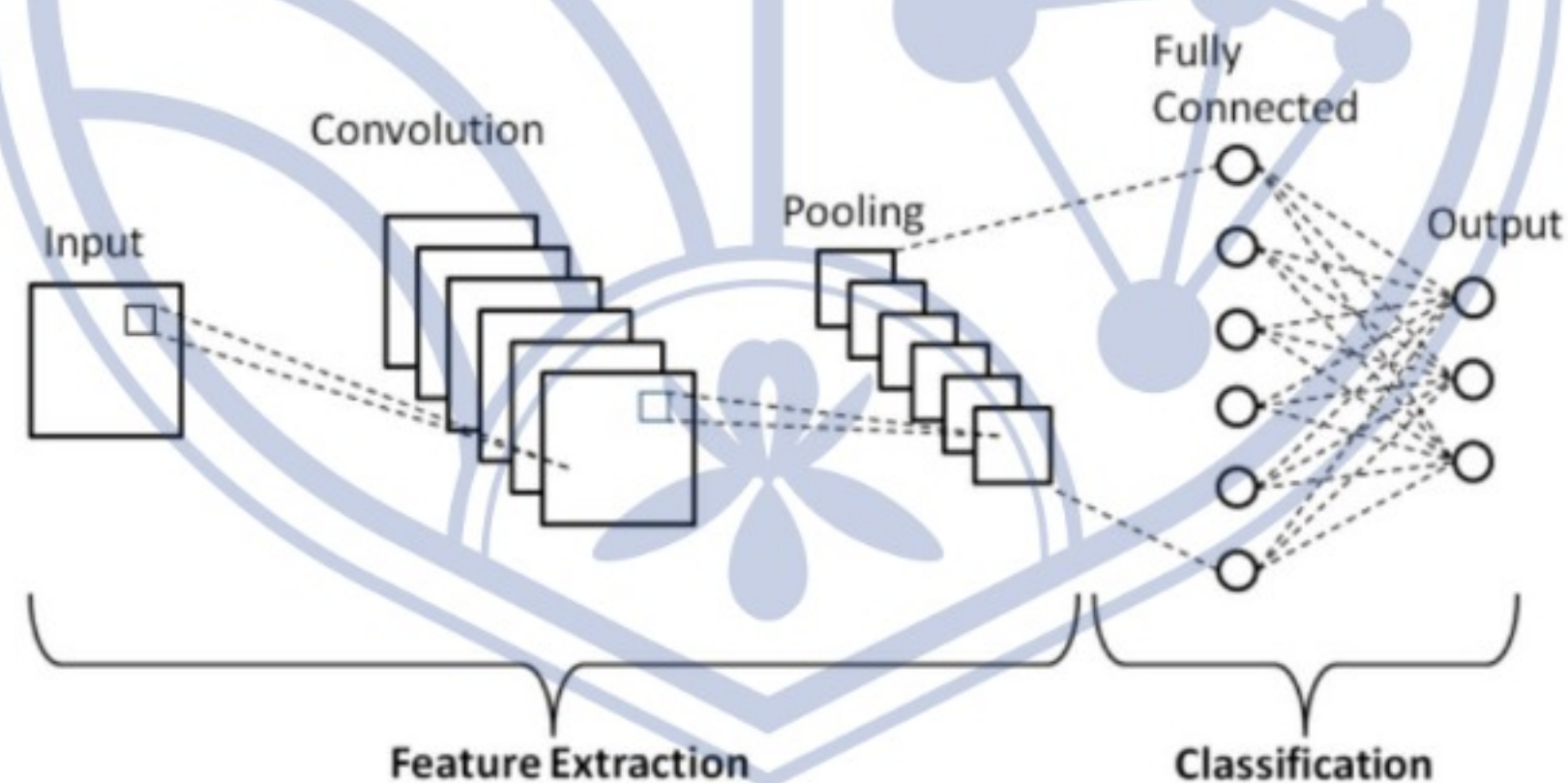
Salah satu komponen krusial dalam arsitektur visi komputer modern adalah mekanisme atensi (*attention mechanism*). Mekanisme ini berfungsi untuk mengarahkan sumber daya komputasi pada area atau fitur tertentu yang memiliki signifikansi tinggi dalam sebuah citra, sekaligus mereduksi pengaruh latar belakang yang tidak relevan atau bersifat *noise* [21]. Integrasi teknik pengolahan citra tradisional dengan model *deep learning* mutakhir memungkinkan sistem untuk mengenali anomali visual tingkat piksel, seperti pola penyakit pada permukaan kulit buah atau perubahan kromatik yang sangat tipis pada fase kematangan tertentu [22]. Keunggulan ini menjadi sangat vital dalam kasus *fine-grained visual classification*, di mana perbedaan antar kelas objek memiliki kemiripan visual yang sangat tinggi [16].

2.2 Deep Learning

Perkembangan kecerdasan buatan (*Artificial intelligence*) dalam beberapa tahun terakhir mengalami pergeseran signifikan menuju pendekatan berbasis *deep learning*, yang

memungkinkan sistem untuk mempelajari representasi data dari skala besar [19]. Berbeda dengan metode konvensional yang mengandalkan perancangan fitur manual (*handcrafted features*), *deep learning* memanfaatkan jaringan saraf tiruan berlapis (*deep neural networks*) untuk mengekstraksi fitur secara hierarkis dan adaptif. Pendekatan ini menjadi sangat relevan dalam pengolahan citra digital karena karakteristik data visual yang kompleks dan dinamis [19]. Dalam domain pengolahan citra, arsitektur yang paling dominan digunakan adalah *Convolutional Neural Network* (CNN), yang dirancang khusus untuk menangkap pola spasial pada citra [7]. CNN bekerja melalui operasi konvolusi menggunakan filter (*kernel*) yang bergerak di seluruh area citra untuk mengekstraksi fitur penting. Pada lapisan awal, model akan mendeteksi fitur sederhana seperti tepi (*edges*) dan gradasi warna, kemudian pada lapisan yang lebih dalam mampu mengenali pola yang lebih kompleks seperti bentuk objek dan tekstur permukaan [7]. Proses ini memungkinkan model membangun representasi fitur secara bertingkat tanpa intervensi manual.

CNN juga memanfaatkan operasi *pooling* untuk mereduksi dimensi fitur sekaligus mempertahankan informasi penting, sehingga meningkatkan efisiensi komputasi dan mengurangi risiko *overfitting* [9]. Kombinasi antara konvolusi dan *pooling* menjadikan CNN sangat efektif dalam mengenali objek pada berbagai kondisi, termasuk perubahan pencahayaan, variasi sudut pandang, serta keberadaan *noise* pada citra. gambar 2.1 akan memberikan hierarki dari penggunaan CNN.



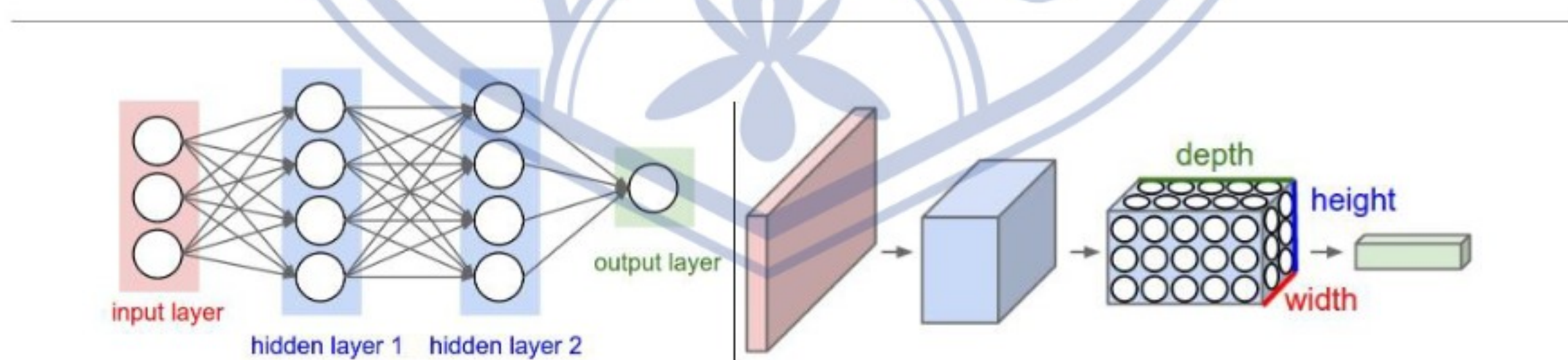
Gambar 2.1 Ekstraksi Fitur Hierarkis *Convolutional Neural Network* (CNN) [23]

Dalam sektor agrikultur, penerapan *deep learning* berbasis CNN telah banyak digunakan untuk meningkatkan akurasi dan konsistensi dalam proses evaluasi kualitas hasil panen [19]. Model mampu mengekstraksi fitur warna secara otomatis, yang menjadi indikator penting dalam menentukan tingkat kematangan buah [7]. Hal ini sangat relevan, di mana perbedaan tingkat kematangan pisang ditentukan oleh perubahan gradasi warna yang

halus serta pola permukaan kulit buah. perkembangan *deep learning* tidak hanya terbatas pada klasifikasi citra murni, tetapi berevolusi ke arah deteksi objek (*object detection*) yang mengharuskan sistem untuk mengenali kelas objek sekaligus melokalisasinya dalam bentuk *bounding box* secara simultan [22]. Deteksi objek dibagi menjadi dua pendekatan: *Two-Stage Detector* yang akurat namun lambat, dan *one-stage detector* yang mengintegrasikan proses ekstraksi fitur dan prediksi objek dalam satu alur komputasi. Pendekatan *deep learning* diimplementasikan melalui model *one-stage detector* YOLOv10 (*you only look once*) [22].

2.3 Object Detection

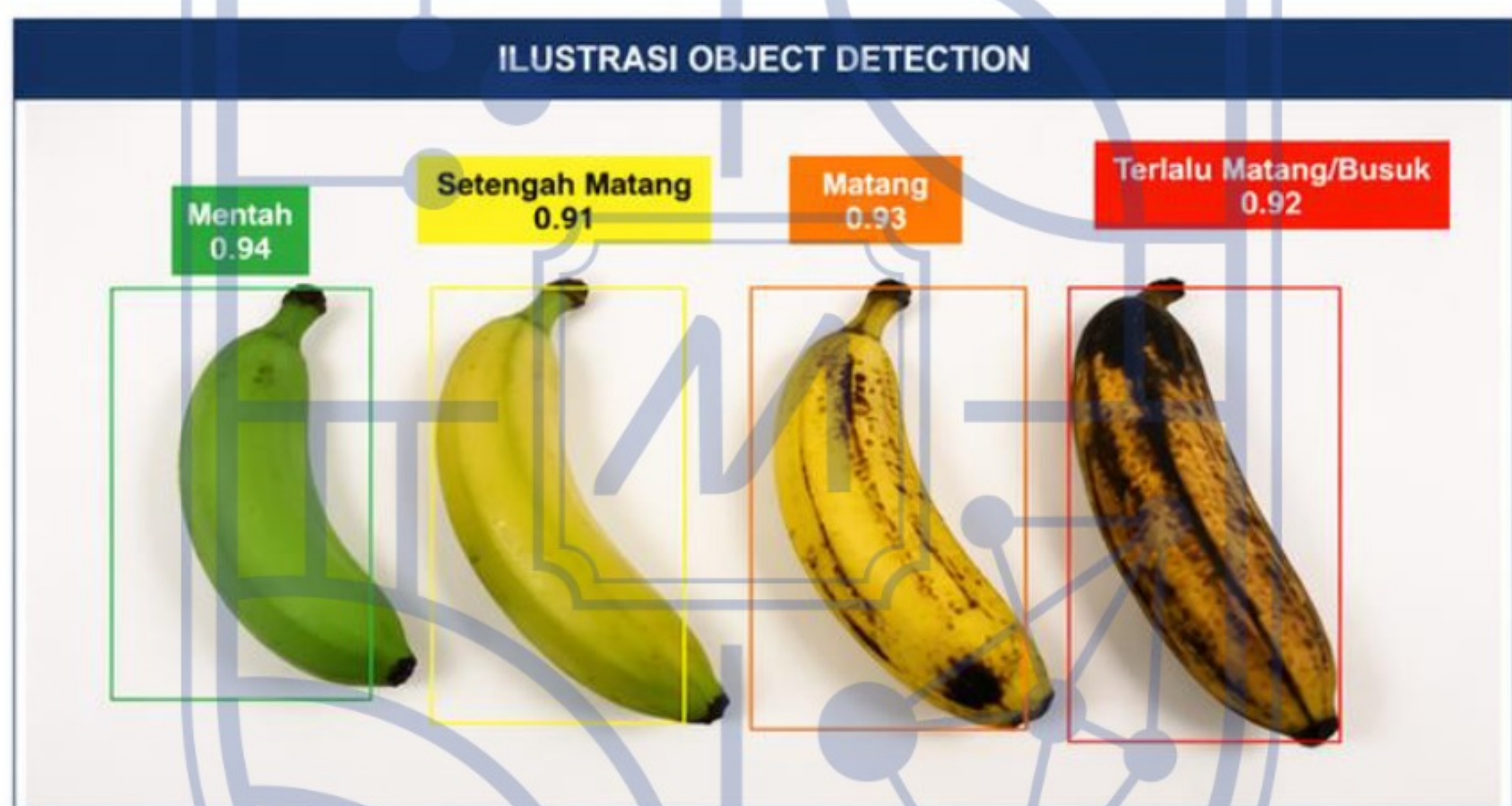
Object detection merupakan salah satu tugas fundamental dalam computer vision yang mengintegrasikan proses klasifikasi objek dengan lokalisasi spasial secara simultan dalam sebuah citra [14]. Berbeda dengan klasifikasi citra konvensional yang hanya menghasilkan label kelas secara global, deteksi objek mampu mengidentifikasi beberapa objek dalam satu gambar sekaligus menentukan posisi masing-masing objek menggunakan *bounding box*. Proses ini dimodelkan sebagai permasalahan regresi yang memprediksi koordinat pusat objek (x, y) serta dimensi objek berupa lebar (w) dan tinggi (h). Implementasi modern deteksi objek berbasis *deep learning* sangat bergantung pada arsitektur *convolutional neural network* (CNN). Arsitektur ini memungkinkan model untuk memproses citra sebagai volume tiga dimensi (lebar, tinggi, dan kedalaman warna) dan mengekstraksi fitur hierarkis melalui lapisan konvolusi, sebagaimana ditunjukkan pada Gambar 2.2.



Gambar 2.2 Arsitektur CNN yang Memproses Citra [24]

Dalam perkembangannya, arsitektur *deep learning* untuk *object detection* secara umum dibagi menjadi dua kategori utama, yaitu *two-stage detector* dan *one-stage detector* [25]. Pendekatan satu tahap (*one-stage detector*), seperti yang digunakan pada algoritma YOLOv10 melakukan proses deteksi dan klasifikasi secara langsung dalam satu tahap

inferensi. Berbeda dengan versi pendahulunya yang masih bergantung pada metode *Non maximum suppression* (NMS) saat inferensi, YOLOv10 memperkenalkan arsitektur *NMS-free training* yang signifikan mengurangi latensi inferensi dan meningkatkan efisiensi komputasi, menjadikannya sangat ideal untuk implementasi *real-time*. Pendekatan ini menawarkan keunggulan dalam hal kecepatan deteksi (*real-time performance*), yang sangat krusial untuk sistem deteksi kematangan pisang di kondisi dinamis. Keluaran atau *output* dari sistem deteksi ini memberikan informasi komprehensif berupa label kelas, lokasi objek, dan tingkat kepercayaan model (*confidence score*). Hasil deteksi tingkat kematangan pisang ditunjukkan pada gambar 2.3.



Gambar 2.3 *Output* Deteksi Objek Tingkat Kematangan Buah Pisang

Kinerja sistem deteksi objek umumnya dievaluasi berdasarkan akurasi lokalisasi *bounding box* terhadap *ground truth* menggunakan metrik *Intersection over Union* (IoU). Performa ini diukur menggunakan *mean Average Precision* (mAP) yang menjadi indikator utama dalam menentukan keandalan model untuk aplikasi dunia nyata. Metrik mAP digunakan karena memberikan gambaran komprehensif mengenai kemampuan model dalam melakukan deteksi pada berbagai tingkat ambang batas IoU, sehingga mencerminkan performa model yang stabil dalam berbagai kondisi objek.

2.4 Algoritma YOLO (*You Only Look Once*)

Algoritma *You Only Look Once* (YOLO) merupakan terobosan dalam domain deteksi objek yang memperkenalkan paradigma detektor satu tahap (*one-stage detector*).

YOLO merevolusi metode deteksi konvensional yang sebelumnya memisahkan tahap pengusulan wilayah (*region proposal*) dan tahap klasifikasi, menjadi satu proses regresi tunggal yang dilakukan secara simultan [14]. YOLO membagi citra masukan ke dalam jaringan sel (*grid*) berukuran $S \times S$. Jika pusat sebuah objek jatuh ke dalam suatu sel *grid*, maka sel tersebut bertanggung jawab untuk mendeteksi objek tersebut dengan memprediksi koordinat *bounding box* (x, y, w, h), skor kepercayaan (*confidence score*), serta probabilitas kelas objek [25]

Karakteristik utama yang membedakan YOLO dari algoritma deteksi lainnya adalah kemampuannya dalam citra selama proses pelatihan dan inferensi. Karena algoritma ini melihat keseluruhan citra dalam satu waktu, YOLO cenderung menghasilkan lebih sedikit kesalahan pada deteksi latar belakang (*background errors*) dibandingkan dengan metode berbasis wilayah [18]. Keunggulan ini, dikombinasikan dengan arsitektur jaringan yang efisien, memungkinkan YOLO mencapai performa *real-time* yang tangguh tanpa mengorbankan akurasi secara signifikan [26]. Sejak diperkenalkan pertama kali, keluarga algoritma YOLO telah mengalami evolusi arsitektural yang masif guna menyeimbangkan metrik kecepatan (*Frames Per Second*) dan akurasi (*mean Average Precision*). Iterasi pengembangan mulai dari YOLOv1 hingga generasi terbaru seperti YOLOv10 fokus pada pengoptimalan modul *backbone* untuk ekstraksi fitur yang lebih kuat, serta perbaikan pada mekanisme *feature fusion* untuk menangkap detail objek pada berbagai skala [15]. Pada implementasi praktis di lingkungan yang dinamis, model YOLO harus mampu menghadapi tantangan teknis seperti variasi pencahayaan yang ekstrim dan oklusi objek [10]. Kehadiran generasi terbaru seperti YOLOv10 membawa inovasi berupa *NMS-free training* yang mengeliminasi latensi pada tahap *post-processing*, sehingga memberikan solusi deteksi yang lebih cepat dan efisien untuk aplikasi industri dan agrikultur pintar [15].

2.5 YOLOv10

YOLOv10 (You Only Look Once version 10) merupakan evolusi terbaru dalam keluarga algoritma YOLO yang dirancang untuk mengatasi tantangan latensi dan akurasi pada sistem deteksi objek *real-time* [9]. Berbeda dengan generasi sebelumnya yang masih bergantung pada mekanisme *post-processing*, YOLOv10 mengusung pendekatan *end-to-end object detection*, yang memungkinkan model melakukan inferensi secara langsung tanpa memerlukan proses *Non-Maximum Suppression* (NMS) [15]. Pada model YOLO generasi terdahulu, NMS digunakan sebagai tahap krusial untuk mengeliminasi *bounding box* yang tumpang tindih. Namun, proses ini menambah beban komputasi dan berpotensi menurunkan

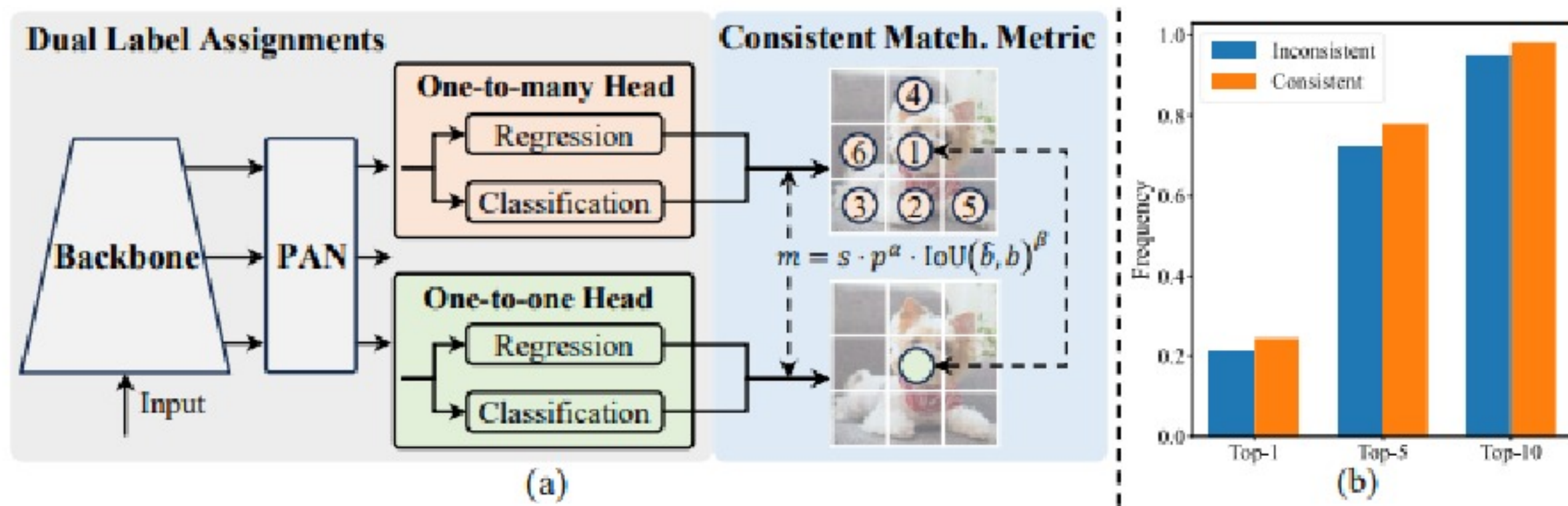
sensitivitas model dalam mendeteksi objek yang saling berdekatan atau mengalami *occlusion* kondisi yang umum terjadi dalam lingkungan pertanian [4]. YOLOv10 mengatasi keterbatasan ini dengan mengintegrasikan proses seleksi prediksi langsung ke dalam arsitektur jaringan, sehingga mampu meningkatkan efisiensi sekaligus mempertahankan akurasi deteksi [17]. YOLOv10 membawa berbagai inovasi pada arsitektur jaringan yang mendukung peningkatan performa deteksi objek, baik dari sisi kecepatan maupun akurasi.

2.5.1 Deskripsi YOLOv10

YOLOv10 dikembangkan dengan pendekatan arsitektur yang lebih efisien dan adaptif terhadap kompleksitas visual pada data dunia nyata. Model ini tetap mempertahankan struktur dasar *one-stage detector* yang terdiri dari *backbone*, *neck*, dan *head*, namun dengan berbagai inovasi yang meningkatkan performa secara signifikan.

YOLOv10n sebagai varian yang digunakan dalam penelitian ini mengawali proses ekstraksi fitur melalui serangkaian lapisan awal (*stem*) pada bagian *backbone*. Lapisan awal ini terdiri atas beberapa convolutional layer dengan ukuran kernel 3×3 dan stride 2 yang berfungsi mereduksi resolusi spasial citra input secara bertahap, sekaligus mengekstraksi fitur visual tingkat rendah (*low-level features*) seperti tepi (*edge*), tekstur, dan pola warna dasar dari permukaan kulit pisang [15]. Citra input berukuran 640×640 piksel diproses melalui lapisan *stem* sehingga menghasilkan *feature map* awal berukuran 320×320 piksel, yang selanjutnya diteruskan secara berjenjang ke tahap *downsampling* berikutnya hingga mencapai resolusi 20×20 piksel pada lapisan akhir *backbone*. Representasi fitur tingkat rendah yang dihasilkan pada lapisan awal ini menjadi fondasi bagi blok CIB dan PSA untuk membangun representasi fitur tingkat tinggi (*high-level features*) yang lebih kompleks, seperti pola gradasi warna dan *sugar spots* yang menjadi ciri pembeda antar kelas kematangan pisang.

Salah satu inovasi utama pada YOLOv10 adalah penerapan *dual assignment strategy*, yang memungkinkan model menghasilkan prediksi yang lebih stabil selama proses pelatihan. Pendekatan ini mengkombinasikan dua skema penugasan label yang berbeda untuk meningkatkan kualitas pembelajaran model terhadap objek target. Pada gambar 2.4 menunjukkan mekanisme *dual label assignment*.

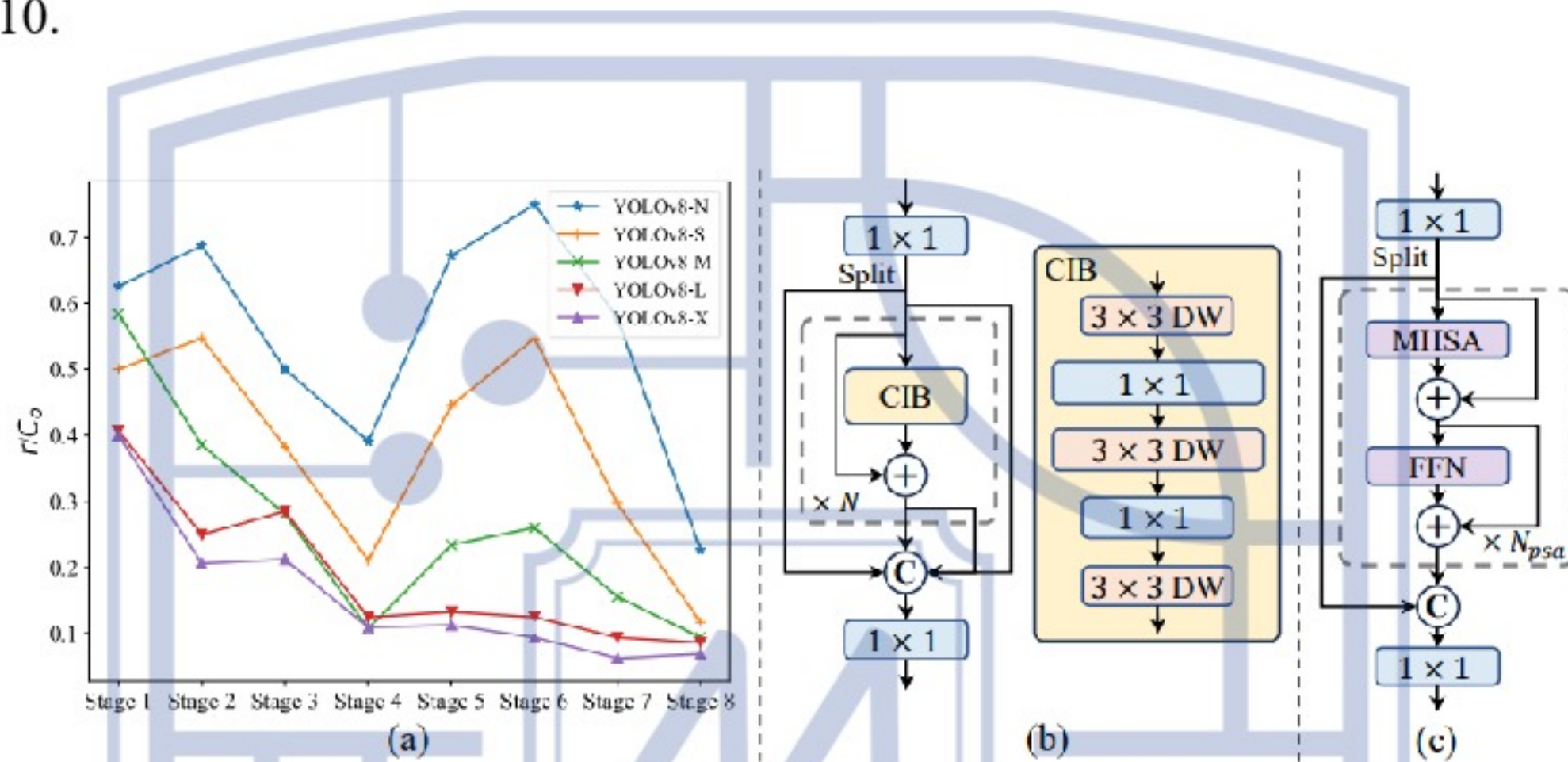


Gambar 2.4 Mekanisme *Dual Label Assignment* pada YOLOv10 [15]

Mekanisme *Dual Label Assignments* yang di perlihatkan pada gambar 2.8 (a) merupakan inovasi kunci pada arsitektur YOLOv10 untuk mencapai efisiensi tinggi melalui deteksi tanpa proses *non-maximum suppression* (NMS). Pada tahap pelatihan (*training*), model mengintegrasikan dua jenis *head* secara bersamaan: *one-to-many head* yang berfungsi untuk mengekstraksi fitur visual kaya dengan memberikan banyak kandidat prediksi untuk satu objek, serta *one-to-one head* yang dirancang khusus untuk menghasilkan hanya satu prediksi tunggal yang paling optimal bagi setiap objek. Sinkronisasi antara kedua jalur ini diatur oleh *consistent matching metric* (m), sebuah fungsi objektif yang memastikan bahwa strategi pemilihan kotak pembatas pada kedua *head* tersebut tetap selaras. Hal ini memungkinkan model untuk mempertahankan akurasi tinggi saat fase inferensi, di mana hanya *one-to-one head* yang digunakan sehingga latensi komputasi akibat proses NMS dapat dihilangkan sepenuhnya. Validasi empiris terhadap efektivitas mekanisme sinkronisasi tersebut ditunjukkan melalui analisis statistik pada gambar 2.8 (b). Grafik tersebut membandingkan tingkat konsistensi prediksi antara kondisi *inconsistent* dan *consistent* pada berbagai tingkatan akurasi (*Top-1*, *Top-5*, dan *Top-10*). Dominasi frekuensi pada bar berwarna oranye (*consistent*) di seluruh kategori membuktikan bahwa implementasi *consistent matching metric* berhasil meningkatkan keselarasan antara *one-to-one* dan *one-to-many head* secara signifikan. Tingkat konsistensi yang lebih tinggi ini menunjukkan bahwa model memiliki stabilitas yang lebih baik dalam mengidentifikasi objek target, yang secara langsung berkontribusi pada peningkatan performa deteksi tingkat kematangan pisang, baik dari aspek akurasi maupun kecepatan proses secara *real-time*.

Melalui pendekatan ini, model mampu mengurangi ambiguitas dalam pemilihan *bounding box* terbaik, sehingga menghasilkan deteksi yang lebih akurat tanpa memerlukan proses penyaringan tambahan pada tahap akhir. Selanjutnya, YOLOv10 mengadopsi konsep *NMS-free training*, yaitu mekanisme yang menghilangkan ketergantungan terhadap *non-*

maximum suppression (NMS). Dengan mengintegrasikan proses seleksi prediksi langsung ke dalam jaringan, YOLOv10 mampu melakukan inferensi lebih cepat dan efisien [15]. YOLOv10 juga memperkenalkan *rank-guided block design*, yaitu pendekatan desain arsitektur yang mengoptimalkan distribusi parameter pada jaringan guna mengurangi redundansi komputasi. Untuk meningkatkan kualitas representasi fitur, YOLOv10 dilengkapi dengan modul tambahan seperti CIB (*compact inverted block*) dan PSA (*partial self-attention*). Pada gambar 2.5 menggambarkan struktur Modul CIB dan PSA pada YOLOv10.

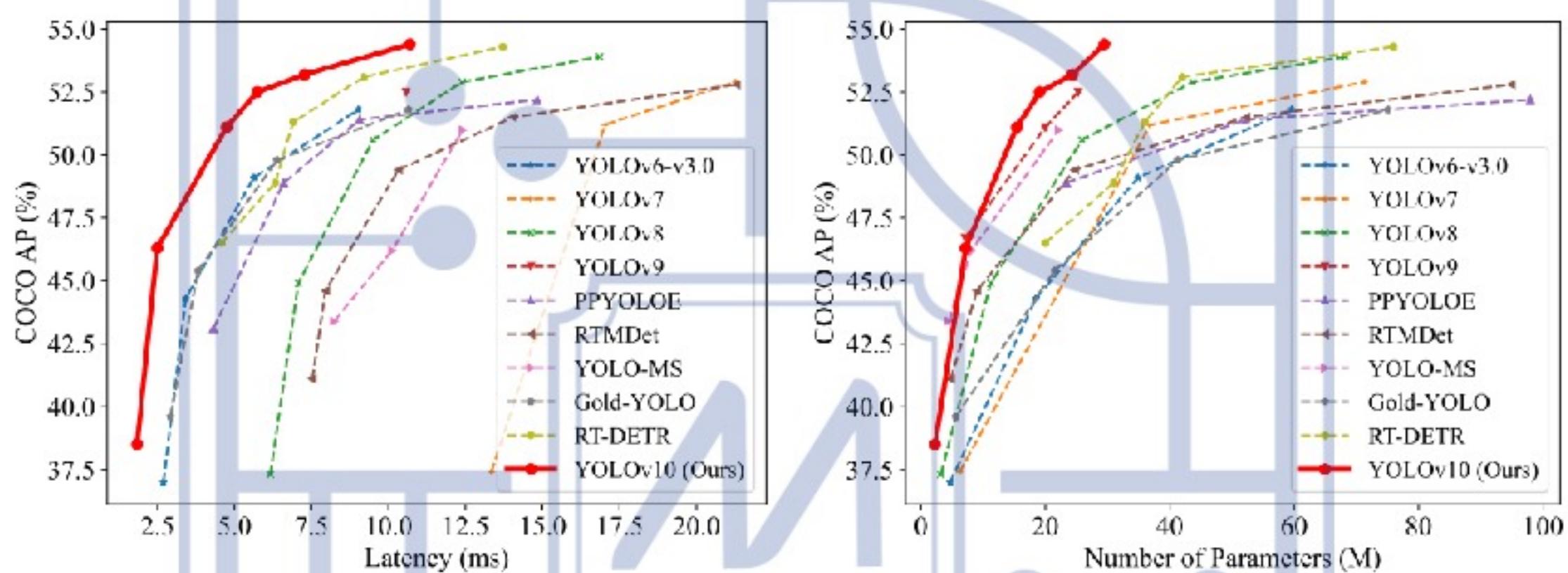


Gambar 2.5 Struktur Modul CIB dan PSA pada YOLOv10 [15]

Optimasi arsitektur pada YOLOv10 dilakukan melalui pendekatan desain yang berorientasi pada efisiensi dan akurasi, sebagaimana digambarkan dalam rangkaian analisis teknis pada gambar tersebut. Berdasarkan *rank analysis* pada gambar 2.9 (a), teridentifikasi adanya redundansi parameter yang signifikan pada berbagai tahapan (*stages*) arsitektur model deteksi objek generasi sebelumnya, yang menjadi landasan utama bagi pengembangan komponen yang lebih ramping. Sebagai solusi untuk meningkatkan efisiensi komputasi, YOLOv10 mengimplementasikan *compact inverted block* (CIB) pada gambar 2.9 (b) yang memanfaatkan kombinasi *depthwise convolution* (3×3 DW) dan konvolusi 1×1 untuk mereduksi beban FLOPs tanpa mengorbankan kapasitas ekstraksi fitur spasial. Guna mengimbangi aspek akurasi, diperkenalkan modul *partial self-attention* (PSA) pada gambar 2.9 (c) yang mengintegrasikan mekanisme *attention* melalui *multi-head self-attention* (MHSA) untuk menganalisis representasi visual citra secara utuh dan mendalam. Integrasi ketiga strategi ini memungkinkan sistem deteksi tingkat kematangan pisang untuk bekerja secara *real-time* dengan ketajaman klasifikasi yang tinggi, terutama dalam mengenali anomali tekstur kulit buah yang seringkali memiliki kemiripan visual yang sangat tipis antar kelasnya.

2.5.2 Evaluasi Model YOLOv10

Evaluasi performa merupakan tahapan krusial untuk mengukur efektivitas model YOLOv10 dalam melakukan deteksi dan klasifikasi tingkat kematangan pisang. Evaluasi difokuskan varietas pisang dengan empat kategori tingkat kematangan, yaitu mentah, setengah matang, matang, dan terlalu matang/busuk. Berbeda dengan klasifikasi citra konvensional, evaluasi pada *object detection* tidak hanya menilai ketepatan klasifikasi (*classification*), tetapi juga ketepatan lokasi objek (*localization*). Hal ini penting karena satu citra dapat mengandung lebih dari satu objek pisang dengan tingkat kematangan yang berbeda. Gambar 2.6 adalah kurva perbandingan performa dan *trade-off* YOLOv10.



Gambar 2.6 Kurva Perbandingan Performa dan *Trade-off* YOLOv10 [15]

Grafik tersebut menyajikan perbandingan performa komprehensif antara YOLOv10 (garis merah) dengan berbagai model deteksi objek mutakhir lainnya, yang menegaskan posisi YOLOv10 sebagai model dengan *trade-off* terbaik antara akurasi dan efisiensi. Pada grafik sebelah kiri, terlihat bahwa YOLOv10 konsisten mencapai nilai *average precision* (AP) yang lebih tinggi pada tingkat latensi yang lebih rendah, yang berarti model ini mampu memproses citra dengan lebih cepat namun tetap lebih akurat dibandingkan pendahulunya seperti YOLOv8 dan YOLOv9. Sementara itu, grafik di sebelah kanan menunjukkan efisiensi parameter yang luar biasa, di mana YOLOv10 mampu memberikan performa deteksi yang kompetitif meskipun menggunakan jumlah parameter yang jauh lebih sedikit dibandingkan model sekelasnya. Bagi sistem deteksi kematangan pisang, hasil *benchmark* ini membuktikan bahwa penggunaan YOLOv10 sangat ideal karena menjamin sistem yang sangat responsif untuk penggunaan *real-time* tanpa mengorbankan ketepatan identifikasi visual pada kondisi lingkungan yang dinamis.

Kurva *Precision–Recall* digunakan untuk menggambarkan hubungan antara ketepatan prediksi model dan kemampuannya dalam mendeteksi seluruh objek. Kurva ini

menjadi dasar dalam perhitungan nilai *Average Precision* (AP) dan *Mean Average Precision* (mAP). Untuk mengukur dan mengevaluasi performa deteksi objek secara menyeluruh hingga menghasilkan nilai *Average Precision* (AP) dan *Mean Average Precision* (mAP), proses evaluasi dilakukan secara bertahap menggunakan beberapa metrik dasar berikut:

1. *Intersection over Union* (IoU)

Intersection over Union (IoU) digunakan untuk mengukur tingkat kesesuaian antara *bounding box* hasil prediksi model (B_p) dengan *ground truth* (B_{gt}) yang telah dianotasi. IoU merepresentasikan akurasi geometris dari prediksi model. Nilai IoU dihitung dengan membagi area irisan dengan area gabungan kedua kotak:

$$IoU = \frac{Area(B_p \cap B_{gt})}{Area(B_p \cup B_{gt})} \dots\dots\dots(1)$$

Keterangan :

- a. B_p = bounding box hasil prediksi model
- b. B_{gt} = bounding box ground truth (hasil anotasi manual)
- c. $Area(B_p \cap B_{gt})$ = luas area irisan (overlap) antara bounding box prediksi dan ground truth
- d. $Area(B_p \cup B_{gt})$ = luas area gabungan (union) antara bounding box prediksi dan ground truth

Nilai IoU berada pada rentang 0 hingga 1. Ambang batas IoU yang digunakan adalah ≥ 0.5 untuk menentukan apakah suatu deteksi dianggap benar. IoU menjadi penting karena objek pisang dari varietas yang berbeda dapat memiliki ukuran dan posisi yang bervariasi dalam citra, sehingga diperlukan ketepatan lokalisasi yang baik untuk memastikan klasifikasi tingkat kematangan tidak keliru.

2. *Confusion Matrix* pada *Object Detection*

Berdasarkan nilai IoU dan *confidence score*, hasil deteksi dikategorikan menjadi:

- a. *True Positive* (TP): Keadaan di mana model berhasil mendeteksi objek pisang dan mengklasifikasikan tingkat kematangannya dengan benar.
- b. *False Positive* (FP): Keadaan di mana model menghasilkan deteksi yang keliru. Kondisi ini terjadi apabila model mendeteksi latar belakang sebagai objek pisang aktualnya bukan, atau model salah dalam memprediksi kelas tingkat kematangan pisang.

- c. *False Negative* (FN): Keadaan di mana objek pisang dan tingkat kematangan pisang yang sebenarnya ada di dalam citra gagal dideteksi sama sekali oleh model.
- d. *True Negative* (TN): Keadaan di mana objek pisang dan tingkat kematangan dengan benar tidak di deteksi sebagai objek apapun oleh model.

Kesalahan FP dapat menyebabkan mis-klasifikasi tingkat kematangan, misalnya pisang mentah terdeteksi sebagai matang. Sementara itu, FN dapat menyebabkan objek tidak terhitung dalam proses evaluasi, yang berdampak pada penurunan kualitas sistem secara keseluruhan.

3. Accuracy

Akurasi merupakan metrik yang digunakan untuk mengukur tingkat ketepatan model secara keseluruhan dalam melakukan prediksi.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \dots\dots\dots (2)$$

Keterangan :

- a. TP (True Positive) = jumlah deteksi benar, model tepat mendeteksi objek pisang dan mengklasifikasikan tingkat kematangannya
- b. TN (True Negative) = jumlah kondisi di mana objek yang bukan pisang/bukan kelas kematangan tertentu memang benar tidak dideteksi oleh model
- c. FP (False Positive) = jumlah deteksi keliru, baik model mendeteksi latar belakang sebagai pisang maupun salah memprediksi kelas kematangan
- d. FN (False Negative) = jumlah objek pisang yang sebenarnya ada dalam citra namun gagal dideteksi oleh model

Nilai akurasi berada pada rentang 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan performa model yang semakin baik. Namun, penggunaan akurasi memiliki keterbatasan. Hal ini disebabkan oleh dominasi area latar belakang dalam citra yang menghasilkan nilai TN yang sangat besar. Akibatnya, nilai akurasi dapat terlihat tinggi meskipun model belum optimal dalam mendeteksi objek utama. Oleh karena itu, akurasi digunakan sebagai metrik pendukung, sedangkan evaluasi utama tetap difokuskan pada metrik Precision, Recall, F1-Score, dan mAP yang lebih representatif dalam mengukur performa deteksi objek.

4. *Precision, Recall, dan F1-Score*

Untuk mengevaluasi kualitas prediksi model YOLOv10, digunakan tiga metrik utama:

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

Keterangan :

- a. TP = jumlah deteksi benar
- b. FP = jumlah deteksi keliru (salah deteksi objek atau salah klasifikasi kelas kematangan)

Precision mengukur tingkat ketepatan model dalam memberikan prediksi. Nilai *precision* yang tinggi menunjukkan bahwa sebagian besar hasil deteksi adalah benar.

$$Recal = \frac{TP}{TP+FN} \dots\dots\dots(4)$$

Keterangan :

- a. TP = jumlah deteksi benar
- b. FN = jumlah objek pisang yang sebenarnya ada dalam citra namun tidak berhasil dideteksi model

Recall mengukur kemampuan model dalam mendeteksi seluruh objek pisang yang ada dalam citra. Nilai recall yang tinggi menunjukkan bahwa model mampu menemukan hampir semua objek yang relevan.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \dots\dots\dots(5)$$

Keterangan :

- a. Precision = tingkat ketepatan model dalam memberikan prediksi
- b. Recall = kemampuan model dalam mendeteksi seluruh objek pisang yang relevan

F1-Score merupakan rata-rata harmonik dari *precision* dan *recall*, yang digunakan untuk mengevaluasi keseimbangan performa model, terutama pada kondisi distribusi data yang tidak seimbang antar kelas kematangan.

5. *Mean Average Precision (mAP)*

Mean Average Precision (mAP) merupakan metrik utama dalam evaluasi model *object detection*. Nilai mAP dihitung sebagai rata-rata dari *Average Precision (AP)* untuk seluruh kelas kematangan:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \dots \dots \dots (6)$$

Keterangan :

- a. N = jumlah total kelas kematangan pisang yang dievaluasi
- b. AP_i = nilai Average Precision untuk kelas kematangan ke i
- c. i = indeks kelas kematangan, dengan i=1,2,.....N

Evaluasi dilakukan menggunakan dua pendekatan:

- i. mAP@0.5 → evaluasi dengan *threshold* IoU sebesar 0.5
- ii. mAP@0.5:0.95 → evaluasi lebih ketat dengan variasi *threshold* IoU

Penggunaan mAP memungkinkan penilaian performa model secara menyeluruh dalam mendeteksi dan mengklasifikasikan seluruh kategori kematangan. Metrik-metrik evaluasi yang digunakan dirangkum pada Tabel 2.1

Tabel 2.1 Metrik Evaluasi

Metrik	Formula	Keterangan
<i>Precision</i>	$TP / (TP + FP)$	Proporsi prediksi positif yang benar
<i>Recall</i>	$TP / (TP + FN)$	Proporsi objek terdeteksi dari total objek nyata
<i>F1-Score</i>	$2 \times (P \times R) / (P + R)$	Rata-rata harmonik <i>Precision</i> dan <i>Recall</i>
mAP@0.5	mean AP pada IoU <i>threshold</i> 0.5	Akurasi deteksi pada <i>threshold</i> longgar
mAP@0.5:0.95	mean AP pada IoU 0.5 s.d. 0.95	Akurasi deteksi komprehensif (metrik utama)
IoU	$Area(Intersection) / Area(Union)$	Kesesuaian <i>bounding box</i> prediksi vs <i>ground truth</i>

6. Efisiensi Komputasi Model

Karena menggunakan citra digital statis (*image-based*) dan tidak mencakup deteksi video secara *real-time*, evaluasi efisiensi difokuskan pada kemampuan model dalam memproses citra secara cepat dan ringan.

Parameter yang digunakan meliputi:

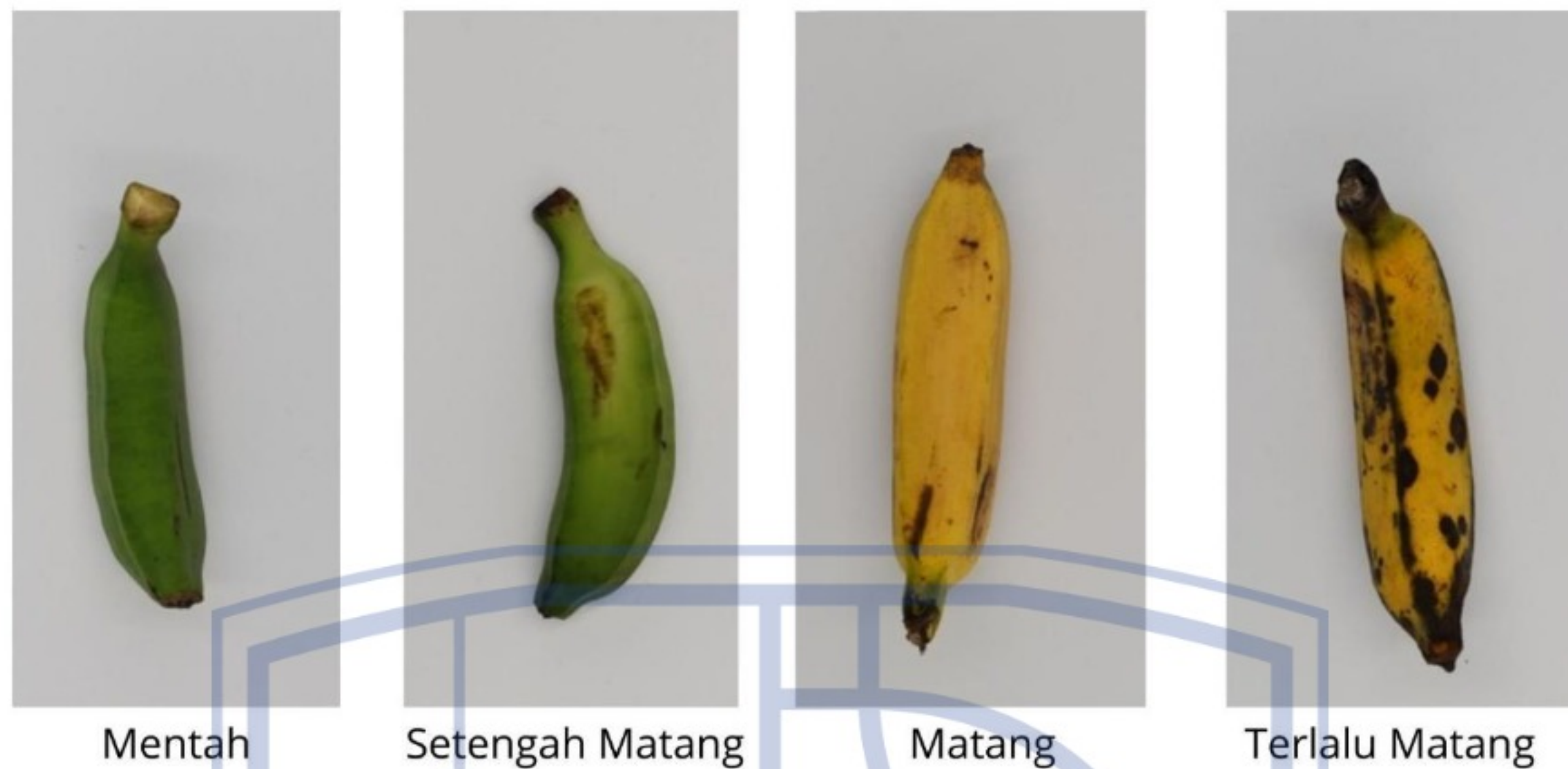
- a. *Inference time* (ms) waktu yang dibutuhkan model untuk memproses satu citra hingga menghasilkan prediksi.
- b. *Model parameters* menunjukkan ukuran model dan kebutuhan memori.
- c. GFLOPs (*Giga Floating Point Operations*) menggambarkan kompleksitas komputasi model.

Efisiensi ini penting untuk memastikan bahwa model YOLOv10 dapat diimplementasikan dengan optimal pada sistem berbasis citra digital tanpa memerlukan perangkat keras berspesifikasi tinggi.

2.6 Karakteristik Kematangan Buah Pisang

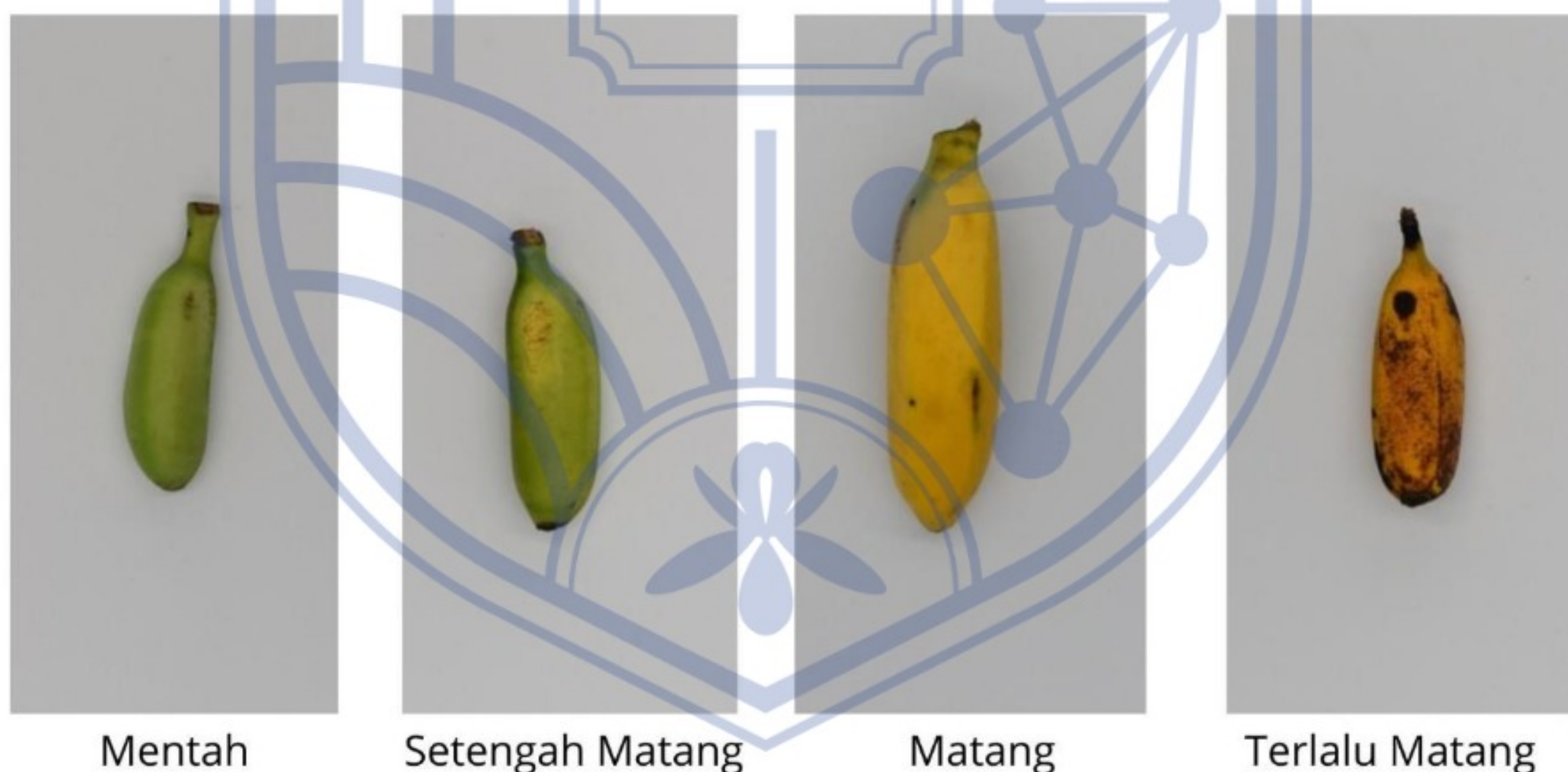
Pisang merupakan komoditas hortikultura yang kaya akan nutrisi dan memiliki peranan penting dalam pemenuhan kebutuhan vitamin dan mineral masyarakat [1]. Sebagai buah klimakterik, pisang tetap mengalami proses pematangan setelah dipanen yang dipicu oleh peningkatan produksi gas etilen [2]. Tingkat kematangan pisang umumnya diklasifikasikan ke dalam tujuh fase utama, mulai dari hijau pekat (mentah), hijau kekuningan, kuning merata, hingga kuning dengan bercak cokelat (*sugar spots*) sebagai indikator tingkat kematangan lanjut [3]. Tantangan utama dalam penanganan pasca-panen adalah mengidentifikasi perbedaan visual antar fase kematangan yang sangat halus, karena kesalahan klasifikasi dapat mempercepat proses pembusukan dan menurunkan umur simpan buah secara signifikan [4].

Pisang Raja memiliki ukuran buah yang relatif besar dengan kulit tebal dan permukaan yang halus. Saat mentah, kulit pisang raja berwarna hijau tua merata. Seiring proses pematangan, warna kulit berubah menjadi hijau kekuningan, kemudian kuning merata disertai kemunculan *sugar spots* (bintik coklat), dan akhirnya mendominasi warna coklat kehitaman pada fase terlalu matang. Gambar 2.7 menunjukkan proses kematangan pisang raja.



Gambar 2.7 Proses Matang Pisang Raja

Pisang Mas memiliki ukuran yang lebih kecil dan ramping dengan kulit yang lebih tipis. Warna kulitnya saat mentah adalah hijau cerah yang berubah menjadi kuning cerah merata saat matang, dengan *sugar spots* yang muncul lebih awal dan lebih banyak dibandingkan pisang Raja. Gambar 2.8 menunjukkan proses kematangan pisang mas.



Gambar 2.8 Proses Matang Pisang Mas

Penentuan kematangan pisang mengacu pada skala *Von Loesecke* yang membagi kematangan menjadi tujuh fase berdasarkan perubahan warna kulit secara progresif, dari hijau penuh (fase 1) hingga kuning berbercak coklat (fase 7) [3]. Perubahan warna ini terjadi akibat degradasi klorofil dan pembentukan karotenoid yang dikendalikan oleh hormon etilen

yang diproduksi buah itu sendiri. Ketujuh tahap tersebut disederhanakan menjadi empat fase kematangan untuk kepraktisan anotasi *dataset* dan relevansi aplikasi di lapangan [4]. Fase pertama adalah mentah, ditandai oleh warna kulit hijau tua yang mendominasi seluruh permukaan pada tiap citra buah pisang, tekstur daging yang keras dan padat, kandungan pati sangat tinggi, serta kadar gula yang masih rendah. Buah pada fase ini belum layak dikonsumsi langsung. Fase kedua adalah setengah matang, dicirikan oleh perpaduan warna hijau dan kuning (*hijau kekuningan*) pada permukaan kulit, menandai proses aktif konversi pati menjadi gula sehingga tekstur daging mulai melunak dan rasa sedikit manis mulai terasa. Fase ketiga adalah matang, merupakan kondisi optimal konsumsi yang ditandai oleh warna kulit kuning merata pada seluruh permukaan disertai kemunculan *sugar spots* berupa bintik-bintik coklat kecil yang merupakan tanda konsentrasi gula telah mencapai puncaknya, dengan tekstur daging yang lunak dan rasa yang manis penuh [7]. Fase keempat adalah terlalu matang atau busuk, yang ditandai oleh perubahan warna kulit menjadi coklat kehitaman secara masif akibat proses oksidasi dan fermentasi lanjut; pada fase ini buah tidak lagi layak dikonsumsi secara langsung namun masih dapat dimanfaatkan sebagai bahan olahan makanan.

