BAB II

LANDASAN TEORI

2.1 Arsitektur MATO

Arsitektur MATO merupakan kerangka sistem deteksi objek kematangan tomat yang dikembangkan dengan mengintegrasikan model YOLOv11 sebagai basis utama (*baseline*) yang telah dimodifikasi. Nama MATO merepresentasikan fokus penelitian pada deteksi kematangan tomat melalui efisiensi arsitektur. Arsitektur ini dirancang untuk menyeimbangkan antara akurasi deteksi dan efisiensi komputasi, sehingga relevan untuk diimplementasikan pada perangkat dengan sumber daya terbatas.

Struktur utama arsitektur MATO terdiri dari komponen-komponen yakni *Input* dan *Praproses*, citra RGB buah tomat sebagai input awal diproses melalui tahap *preprocessing* dengan mengubah ukuran (*resize*) menjadi 640x480 *piksel*. Hal ini dilakukan untuk menjaga konsistensi dimensi pada *feature map* dan menstabilkan proses pelatihan. *Backbone* (Ekstraksi Fitur), bagian ini berfungsi untuk mengekstraksi fitur *multiskala* dari citra *input*. Dalam arsitektur MATO, blok standar C3k2 pada setiap tahapan (*stage*) *backbone* digantikan dengan blok usulan yaitu C3k2_Lite. Penggantian ini bertujuan untuk memperkuat ekstraksi fitur kontekstual pada lingkungan yang kompleks tanpa menambah beban parameter secara signifikan. *Neck* (Penggabungan Fitur), MATO menggunakan struktur *Neck* untuk menggabungkan fitur dari berbagai skala melalui mekanisme *fusion* (FPN/PAN). Sama halnya dengan bagian *backbone*, blok C3k2_Lite diintegrasikan ke dalam jalur *top-down* dan *bottom-up* untuk memastikan interaksi antar fitur tetap

dinamis dan efisien. *Head* (Prediksi), bagian akhir arsitektur berfungsi menghasilkan himpunan prediksi yang mencakup lokasi objek (*bounding box*), klasifikasi kelas kematangan (matang atau mentah), dan skor kepercayaan (*confidence score*).

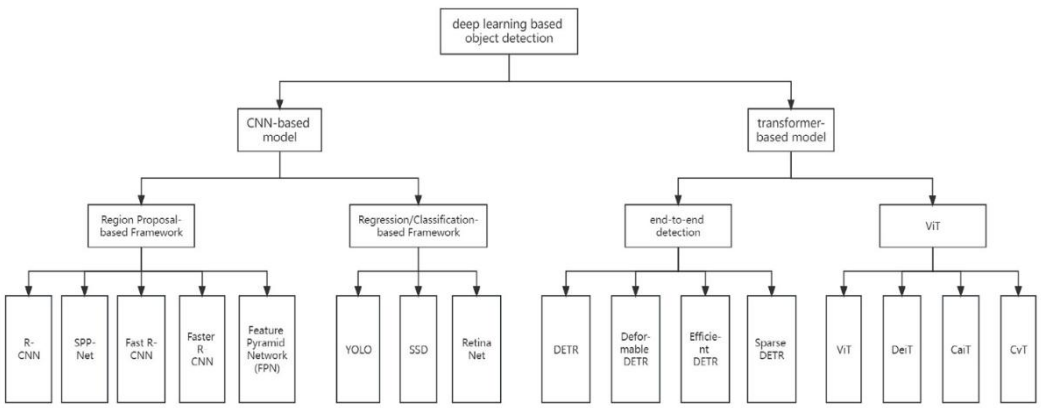
Inti dari kebaruan arsitektur MATO terletak pada penggunaan blok C3k2_Lite yang mengintegrasikan mekanisme *Fast Context Unit* (FCU) untuk ekstraksi fitur berkecepatan tinggi dan *Cross-Gating Enhancement* (CGE) untuk meningkatkan selektivitas fitur terhadap objek target sembari menekan gangguan latar belakang (*noise*). Dengan desain ini, arsitektur MATO mampu mencapai peningkatan presisi dan pengurangan waktu inferensi dibandingkan dengan model standar.

2.2 Deteksi Objek (*Object Detection*)

Deteksi objek adalah salah satu bidang paling penting yang termasuk ke dalam bidang kecerdasan buatan dan cabang keilmuan visi komputer. Cara kerja deteksi objek yaitu dapat menemukan dan menentukan posisi yang tepat dari objek yang dituju dalam gambar atau pun video (Y. Sun dkk., 2024) dengan memberikan *bounding box* di sekeliling objeknya. *Bounding box* sendiri adalah kotak persegi panjang yang digunakan dalam deteksi objek untuk menentukan lokasi dan ukuran suatu objek dalam gambar atau video. *Bounding box* direpresentasikan dengan koordinat (x, y, w, h), yakni (x, y) sebagai koordinat sudut kiri atas kotak, w (*width*) sebagai lebar *bounding box*, h (*height*) sebagai tinggi *bounding box* (Redmon dkk., 2015).

Deteksi objek juga dapat melakukan proses klasifikasi yakni dengan memberi informasi bahwa objek yang ditemukan termasuk ke dalam kelas atau kategori tertentu. Deteksi objek sudah banyak digunakan pada berbagai tugas seperti pengawasan video (Ma dkk., 2021), deteksi wajah (Y. Chen & Joo, n.d.), pengemudian otonom pada kendaraan (Hu dkk., 2022), serta pada lingkup pertanian untuk mendeteksi tingkat kematangan buah (Rizzo dkk., 2023). Model deteksi objek, terdapat dua metrik utama yang digunakan untuk mengukur kinerja model, yaitu akurasi dan kecepatan deteksi (Zou dkk., 2019).

Metode deteksi objek tradisional pada mulanya menggunakan teknik perhitungan *sliding window* untuk memilih area kandidat tempat objek berada, yakni dengan cara mencari semua kemungkinan area dan melakukan pengukuran pada setiap gambar yang ditangkap untuk mengidentifikasi objek. Namun, metode ini menghadapi tantangan dalam mendeteksi objek dengan berbagai ukuran dan rasio aspek, yang dapat mengakibatkan ketidakakuratan serta tingginya beban komputasi akibat pencarian yang ekstensif (Sapkota dkk., 2024). Selain itu, teknik ini dinilai tidak *robust* ketika menangani tugas untuk mendeteksi



objek yang berbeda-beda dan *background* area objek yang kompleks (Y. Sun dkk., 2024).

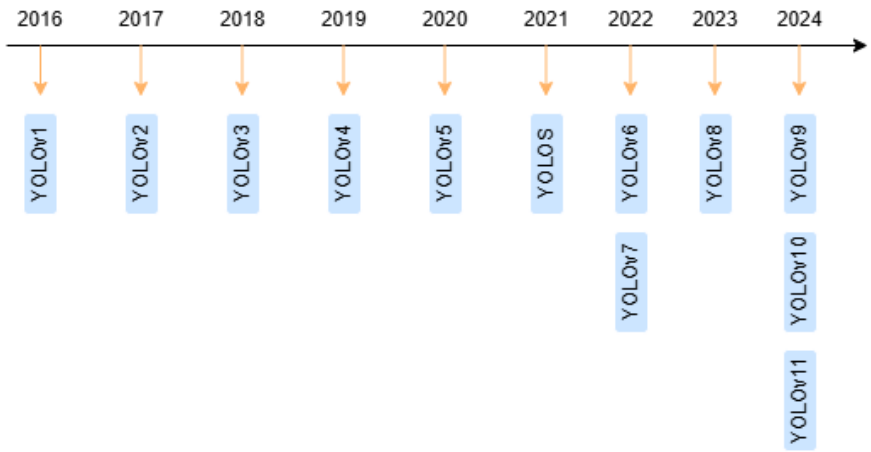
Gambar 2. 1 Klasifikasi metode deteksi objek (Y. Sun dkk., 2024)

Perkembangan *deep learning* telah melahirkan algoritma yang lebih *robust* untuk menangani fitur *multiscale* dan mengatasi keterbatasan metode deteksi objek tradisional. Tahun 2012, *Convolutional Neural Network* (CNN) menggantikan metode tradisional dengan kemampuan representasi fitur yang lebih baik, dengan cara menghilangkan kebutuhan akan fitur buatan (Xie dkk., 2021).

Gambar 2.1 terlihat bahwa algoritma atau arsitektur deteksi objek sendiri di klasifikasikan menjadi dua, yaitu model yang berbasiskan *Convolutional Neural Network* (CNN) dan juga berbasiskan *Transformer*. Model berbasiskan *Convolutional Neural Network* khususnya untuk tugas klasifikasi terdapat model yang sedang pesat perkembangannya saat ini yaitu YOLO (*You Only Look Once*) yang dikembangkan pertama kali oleh (Redmon dkk., 2016). Terdapat juga metode berbasiskan *transformer*, di mana pada arsitekturnya menggunakan *self-attention mechanism* untuk menangani *dataset* berkapasitas besar dan kompleks. Sehingga, membawa kemampuan deteksi objek ke tingkat lebih maju dengan arsitektur yang lebih fleksibel dan adaptif terhadap data yang kompleks. Menurut (Rizzo dkk., 2023) dengan mengombinasikan algoritma berbasis CNN dan berbasis *transformer* untuk tugas tertentu seperti mendeteksi tingkat kematangan pada buah, akan menghasilkan kinerja model yang menjanjikan.

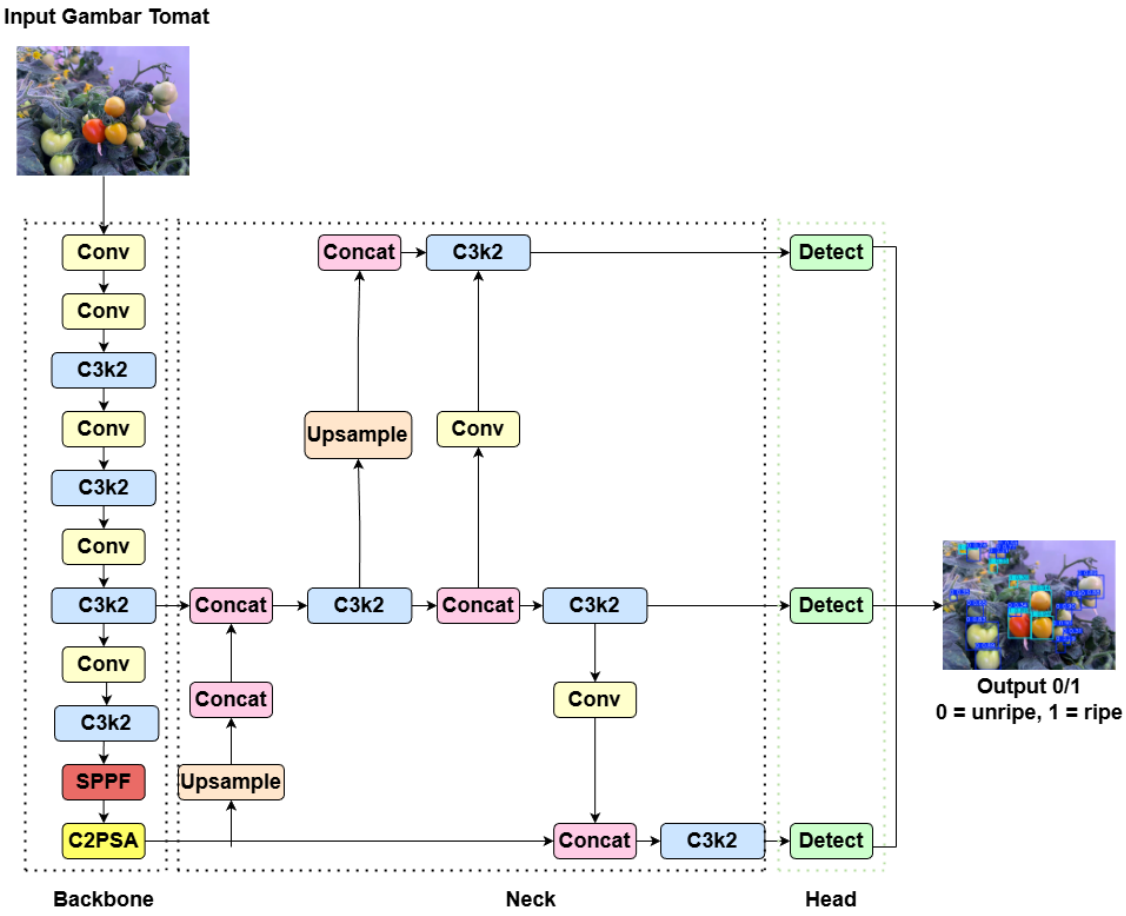
2.3 YOLO

Tahun 2016, (Redmon dkk., 2016) memperkenalkan algoritma deteksi objek kepada publik yang diberi nama YOLO (*You Only Look Once*). Penemuan ini melahirkan revolusi baru di bidang deteksi objek secara *realtime*, yakni dengan mengombinasikan proposal *region* dan klasifikasi ke dalam *single neural network* yang mana hal tersebut mampu mereduksi waktu komputasi (Sapkota dkk., 2024). Model YOLO bekerja dengan membagi gambar menjadi beberapa kotak kecil (*grid*), lalu memprediksi lokasi objek (*bounding box*) dan jenis objeknya secara langsung di setiap kotak. Pendekatan ini memungkinkan YOLO belajar secara langsung dari data tanpa memerlukan proses tambahan, sehingga lebih cepat dan efisien (Redmon dkk., 2016). YOLO sering digunakan pada berbagai bidang keilmuan yang menitik beratkan akurasi dan kecepatan dalam deteksi seperti pada bidang agrikultur (Badgujar dkk., n.d.), kesehatan (Prinzi dkk., 2024), industri kendaraan otonom (Tang dkk., 2024), dan bidang-bidang lainnya. Gambar 2.2 menunjukkan evolusi dari berbagai versi YOLO yang hadir atau rilis setiap tahunnya dimulai dari YOLO versi pertama di tahun 2016.



Gambar 2. 2 Evolusi YOLO terinspirasi dari (Sapkota dkk., 2024)

Dikarenakan banyaknya pengadopsian YOLO untuk tugas di visi komputer, setiap tahunnya selalu rilis versi terbaru dengan penambahan fitur yang dapat menjawab limitasi versi sebelumnya. Dimulai pada versi YOLOv1 dengan memiliki kecepatan tinggi dan akurasi yang baik karena langsung memproses seluruh gambar dalam satu langkah, tanpa memerlukan tahap pemrosesan tambahan seperti metode sebelumnya. Sampai pada akhir tahun 2024 tim Ultralytics mengembangkan YOLOv11 yang hadir dengan struktur model yang lebih canggih untuk menangkap detail objek lebih akurat. Model ini dirancang agar lebih cepat, efisien, dan performa tinggi, berkat arsitektur serta teknik pelatihan yang telah dioptimalkan. Salah satu keunggulan utamanya adalah kemampuannya dalam menjaga keseimbangan antara akurasi dan efisiensi komputasi, sehingga dapat digunakan untuk berbagai keperluan, mulai dari perangkat kecil seperti IoT hingga sistem AI berskala besar. Dari perbaikan versi sebelumnya, YOLOv11 mampu mendeteksi objek lebih akurat dan memproses gambar secara real-time dalam berbagai kondisi lingkungan. Model ini juga tersedia dalam lima versi berbeda (YOLOv11 n, s, m, l, x), yang dibedakan berdasarkan tingkat kompleksitas jaringannya.



Gambar 2. 3 Arsitektur YOLO11 (Sapkota dkk, 2024)

YOLOv11 membawa peningkatan besar dalam struktur modelnya, sehingga lebih efisien dan akurat dalam berbagai tugas visi komputer. Salah satu perubahan penting adalah penggunaan blok C3k2, yang menggantikan C2f dari versi sebelumnya, menjadikan pemrosesan lebih ringan (Sapkota dkk., 2024). Model ini juga mempertahankan fitur unggulan SPPF dan menambahkan modul C2PSA, yang membantu menangkap fitur gambar dengan lebih baik, terutama untuk objek kecil atau yang tertutup sebagian. Inovasi ini, YOLOv11 menjadi lebih fleksibel dan

dapat digunakan untuk deteksi objek, segmentasi, klasifikasi gambar, hingga deteksi *oriented bounding box* (OBB).

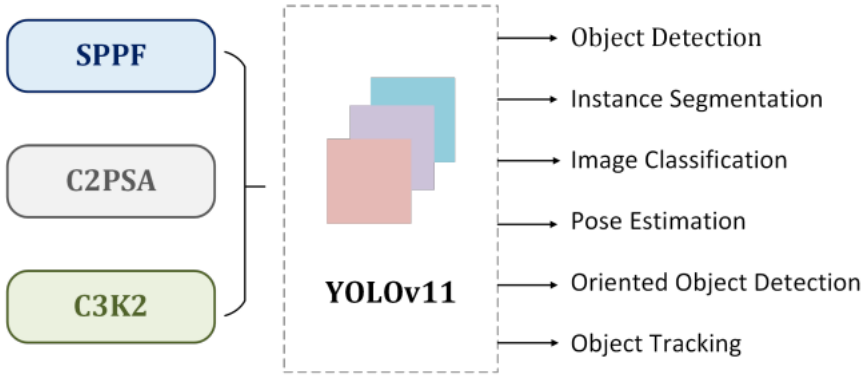
Hasil uji coba menunjukkan bahwa YOLOv11 lebih akurat dan lebih efisien dibandingkan versi sebelumnya yaitu YOLOv10. Varian YOLOv11m terbukti memiliki presisi lebih tinggi pada *dataset* COCO, sekaligus menggunakan lebih sedikit parameter (22% lebih ringan) dibanding YOLOv8m, sehingga model berjalan lebih cepat tanpa kehilangan akurasi yang signifikan. Selain itu, YOLOv11 lebih cepat sekitar 2% dibandingkan YOLOv10, menjadikannya pilihan terbaik untuk aplikasi *real time*. Namun, diantara semua keunggulan tersebut YOLOv11 masih memiliki kekurangan yaitu menemukan keseimbangan antara kinerja yang cepat dan hasil yang akurat ketika mendeteksi pada lingkungan yang kompleks (Sapkota dkk., 2024).

2.4 YOLOv11

Konferensi *YOLO Vision 2024 (YV24)*, hadirnya YOLOv11 menandai kemajuan besar dalam bidang *real-time object detection*. Model ini memperkenalkan berbagai penyempurnaan pada arsitektur dan pendekatan pelatihan yang digunakan, sehingga mampu memberikan performa yang lebih unggul dalam hal akurasi, kecepatan deteksi, serta efisiensi komputasi.

Arsitektur inovatif YOLOv11 dirancang dengan memanfaatkan teknik ekstraksi fitur tingkat lanjut yang memungkinkan model mengenali detail visual secara lebih mendalam tanpa meningkatkan kompleksitas parameter. Pendekatan efisien ini berdampak pada peningkatan akurasi di berbagai tugas *computer vision*, seperti

object detection dan *image classification*. Tak hanya itu, efisiensi rancangan tersebut juga membuat YOLOv11 mampu mencapai kecepatan pemrosesan yang jauh lebih tinggi, menjadikannya sangat andal untuk aplikasi yang menuntut performa *real-time*. Adapun pada Gambar 2.4 merupakan kunci kebaruan dari arsitektur YOLOv11 yang diambil dari (Khanam & Hussain, 2024).



Gambar 2. 4 Fitur utama YOLOv11

Arsitektur YOLOv11 masih sama dengan versi YOLO lainnya terbagi menjadi 3 bagian yaitu *backbone, neck,* dan *head.* Arsitektur YOLO, *backbone* berfungsi sebagai inti utama yang bertugas mengekstraksi informasi visual dari citra masukan pada berbagai skala. Mekanisme ini dilakukan melalui susunan berlapis dari *convolutional layers* dan blok pemrosesan khusus yang dirancang untuk menghasilkan *feature maps* dengan resolusi yang bervariasi, sehingga memungkinkan model memahami struktur dan pola objek secara lebih mendalam. Peningkatan utama pada YOLOv11 terletak pada penggantian blok C2f dengan blok baru bernama C3k2. Blok ini dikembangkan sebagai bentuk optimasi dari *Cross Stage Partial (CSP) Bottleneck* agar lebih efisien dalam penggunaan sumber

daya komputasi. Jika pada YOLOv8 proses ekstraksi fitur dilakukan melalui satu konvolusi besar, maka C3k2 memecahnya menjadi dua konvolusi kecil untuk menekan beban komputasi. Notasi "k2" merepresentasikan penggunaan *kernel* berukuran kecil yang mampu mempercepat proses inferensi, sekaligus menjaga stabilitas dan kualitas performa model secara keseluruhan (Khanam & Hussain, 2024). Arsitektur pada YOLOv11, blok *Spatial Pyramid Pooling – Fast (SPPF)* dari versi sebelumnya tetap dipertahankan karena efektivitasnya dalam menggabungkan informasi dari berbagai skala fitur. Namun, pembaruan penting muncul dengan penambahan blok baru, yaitu *Cross Stage Partial with Spatial Attention (C2PSA)*, yang ditempatkan setelah SPPF. Blok C2PSA memungkinkan model untuk secara adaptif memberikan perhatian lebih besar pada area yang relevan dalam citra, sehingga dapat meningkatkan ketepatan deteksi terhadap objek dengan ukuran, bentuk, maupun posisi yang beragam (Khanam & Hussain, 2024).

Bagian *neck* YOLOv11 berfungsi menggabungkan fitur dari berbagai tingkat skala melalui proses *upsampling* dan *concatenation* untuk diteruskan ke *head* dalam tahap prediksi. Pembaruan utama pada versi ini adalah penggantian blok C2f dengan C3k2, yang dirancang lebih cepat dan efisien untuk meningkatkan kinerja agregasi fitur tanpa mengorbankan akurasi. Selain itu, YOLOv11 memperkenalkan modul C2PSA (*Cross Stage Partial with Spatial Attention*) yang memperkuat kemampuan model dalam memusatkan perhatian pada area penting dalam citra. Mekanisme *spatial attention* ini membantu meningkatkan ketepatan deteksi, terutama terhadap objek berukuran kecil atau yang tertutupi sebagian,

sekaligus menjadi keunggulan utama yang membedakan YOLOv11 dari versi sebelumnya, YOLOv8 (Khanam & Hussain, 2024).

Bagian *head* pada YOLOv11 berperan dalam menghasilkan prediksi akhir untuk tugas *object detection* dan *classification*. Komponen ini memproses *feature maps* yang diteruskan dari bagian *neck* dan kemudian menghasilkan keluaran berupa *bounding boxes* serta label kelas untuk setiap objek yang terdeteksi dalam citra. Maka dari itu, *head* menjadi tahap akhir yang mengonversi representasi fitur menjadi hasil deteksi yang dapat diinterpretasikan secara langsung. Bagian *head*, YOLOv11 memanfaatkan beberapa blok C3k2 untuk memproses dan menyempurnakan *feature maps* secara efisien. Blok-blok C3k2 ini ditempatkan pada beberapa jalur di dalam *head* untuk menangani fitur multi-skala pada berbagai tingkat kedalaman. Setelah proses tersebut, *head* YOLOv11 dilengkapi dengan beberapa lapisan CBS (*Convolution–BatchNorm–SiLU*) (Shigematsu & Kato, 2024) yang berfungsi sebagai komponen dasar dalam ekstraksi fitur dan proses deteksi. Lapisan CBS memastikan bahwa *feature maps* yang telah disempurnakan dari blok C3k2 diteruskan ke lapisan berikutnya secara optimal untuk menghasilkan prediksi *bounding box* dan klasifikasi objek dengan akurasi tinggi.

YOLOv11 merepresentasikan kemajuan besar dalam bidang *computer vision (CV)* dengan menghadirkan kombinasi yang kuat antara peningkatan kinerja dan fleksibilitas. Model ini menggabungkan peningkatan pada proses ekstraksi fitur, optimalisasi kinerja komputasi, serta kemampuan adaptasi terhadap beragam tugas visual. Kombinasi tersebut menjadikan YOLOv11 sebagai model yang

tangguh dan serbaguna dalam menyelesaikan berbagai tantangan pengenalan objek, baik untuk kebutuhan riset akademik maupun aplikasi dunia nyata.

2.5 Blok C3k2

Blok C3k2 merupakan salah satu komponen kunci yang diperkenalkan dalam arsitektur YOLOv11 sebagai bentuk optimasi dari blok C2f yang digunakan pada versi sebelumnya. Pengembangan blok ini didasarkan pada konsep *Cross Stage Partial* (CSP) *Bottleneck* yang dirancang untuk meningkatkan efisiensi penggunaan sumber daya komputasi tanpa mengorbankan kedalaman fitur yang diekstraksi.

Secara struktural, perbedaan utama C3k2 dibandingkan dengan mekanisme ekstraksi fitur pada model pendahulunya terletak pada strategi pemrosesan konvolusinya. Jika pada YOLOv8 proses ekstraksi dilakukan melalui satu operasi konvolusi besar, maka pada blok C3k2, proses tersebut dipecah menjadi dua konvolusi yang lebih kecil. Penggunaan notasi "k2" dalam blok ini merepresentasikan pemanfaatan *kernel* berukuran kecil yang bertujuan untuk mempercepat proses inferensi, di mana penggunaan *kernel* kecil secara signifikan mengurangi beban komputasi per lapisan. Menjaga stabilitas performa, meskipun lebih ringan, pemecahan konvolusi ini tetap mampu menjaga kualitas fitur yang dihasilkan sehingga akurasi model tetap terjaga.

Dalam arsitektur YOLOv11, blok C3k2 diintegrasikan secara luas pada bagian *Backbone* untuk ekstraksi fitur awal, serta pada bagian *Neck* untuk membantu proses agregasi fitur multiskala. Implementasi C3k2 pada bagian *Head* juga berperan penting dalam menyempurnakan *feature maps* sebelum dilakukan prediksi

akhir terhadap lokasi (*bounding box*) dan klasifikasi objek. Meskipun blok C3k2 standar telah membawa peningkatan efisiensi sebesar 22% lebih ringan dibandingkan YOLOv8m , dalam skenario lingkungan yang kompleks, blok ini masih menghadapi tantangan dalam menyeimbangkan kecepatan dan ketepatan deteksi

2.6 Evaluasi Kinerja Model

Proses menganalisis dan mengevaluasi apakah model yang kita buat meningkat secara nilai akurasi dan mengurangi waktu inferensi, maka kita perlu mengevaluasinya menggunakan metrik yang relevan untuk mengukurnya. Di mana metrik ini akan menjadi tolak ukur perbandingan hasil dari model yang sudah ditambahkan blok C3k2_Lite. Di dalam penelitian ini metrik yang digunakan adalah mengikuti rekomendasi dari penelitian (Sapkota dkk., 2024) yang merupakan *survey paper* untuk algoritma YOLO untuk tugas visi komputer salah satunya deteksi objek. Metrik yang digunakan adalah mAP50, *recall, precision, inference time,* dan beban memori.

2.7 Penelitian Terkait

Saat ini sudah banyak perangkat atau sistem yang membantu para petani dalam menyelesaikan tugas dalam menentukan tingkat kematangan pada buah tomat. Beberapa sistem dibuat dengan menggunakan YOLO dalam berbagai versi. Algoritma YOLO itu digunakan semata-mata untuk menyelesaikan permasalahan pada bidang deteksi objek yaitu membuat model memiliki ketahanan yang tinggi ketika dihadapkan pada situasi atau lingkungan yang kompleks (X. Wang dkk., 2023). Maka, dibuatlah *state of the art* terkait dengan penelitian yang dilakukan

serta melihat kesesuaian pemanfaatan dari *object detection* pada sistem yang membantu petani dalam mendeteksi tingkat kematangan buah tomat.

Tabel 2. 1 *State of The Art* deteksi tingkat kematangan buah tomat menggunakan YOLO

Penulis	Metode Deteksi Objek	Teknik Modifikasi Arsitektur	Keterangan
Tomato Ripening Detection in Complex Environments Based on Improved BiAttFPN Fusion and YOLOv11-SLBA Modeling (Hao dkk., 2025)	YOLOv11-SLBA	Kombinasi SPPF-LSKA + BiAttFPN + DetectAux	**Hasil:** Model YOLOv11-SLBA diuji pada dataset tomat dengan kondisi kompleks seperti variasi pencahayaan, *occlusion*, dan latar belakang alami. Dibandingkan dengan model lain (Faster R-CNN, SSD, RT-DETR, YOLOv7, YOLOv8, dan YOLOv11), model ini menunjukkan hasil terbaik dengan Precision 92.0%, Recall 83.5%, mAP_{50} 91.3%, $mAP_{50–95}$ 64.6%, dan F1-Score 87.5%. Selain akurat, YOLOv11-SLBA juga efisien dengan waktu inferensi 2.3 ms, ukuran model 10.9 MB, dan parameter 2.72 juta, menunjukkan kinerjanya yang optimal untuk deteksi kematangan tomat secara *real-time*. ***Gap:*** Beban model dan *inference time* bertambah dibandingkan model YOLOv11 *baseline*. Hal ini terjadi karena adanya penambahan *attention mechanism* baru.
YOLO-PGC: A Tomato Maturity Detection Algorithm Based on Improved YOLOv11 (Q. Wu dkk., 2025)	YOLOv11-PGC	Ditingkatkan dengan tiga modul baru (PSSD, GHVC, CIF2M)	**Hasil:** Model YOLO-PGC diuji untuk mendeteksi tingkat kematangan tomat dan menunjukkan peningkatan performa signifikan dibandingkan model deteksi objek sebelumnya. Hasil pengujian menunjukkan bahwa model ini mencapai mAP sebesar 81.6%, Precision 80.4%, dan Recall 74.6%, dengan jumlah parameter 3.8 juta serta FLOPs sebesar 8.7 G. Dibandingkan dengan model lain seperti YOLOv3, YOLOv5, YOLOv8, dan YOLOv11 standar, YOLO-PGC unggul dalam hal akurasi sekaligus efisiensi komputasi. Selain itu, eksperimen membuktikan bahwa YOLO-PGC mampu mendeteksi tomat pada berbagai tahap kematangan, kondisi pencahayaan berbeda, dan situasi *occlusion* dengan stabil serta dapat bekerja secara *real-*

			time di lingkungan pertanian yang kompleks. ***Gap:*** Nilai *GFLOPs* masih tinggi yaitu pada angka 8.7 dibandingkan dengan YOLOv11 *baseline.*
URT-YOLOv11: A Large Receptive Field Algorithm for Detecting Tomato Ripening Under Different Field Conditions (Mu dkk., 2025)	URT-YOLOv11	Iintegrasi UniRepLKNet, RFCBAMConv	**Hasil:** URT-YOLOv11 memperluas arsitektur YOLOv11 dengan integrasi UniRepLKNet, RFCBAMConv, dan TADDH, menghasilkan model yang lebih efisien dan akurat dalam mendeteksi tingkat kematangan tomat di lingkungan nyata. Model ini meningkatkan mAP sebesar 2.2%, menurunkan parameter 16%, serta mempertahankan deteksi stabil di bawah pencahayaan dan occlusion yang kompleks—menjadikannya solusi unggul untuk *real-time intelligent agriculture*.
Tomato ripeness detection method based on FasterNet block and attention mechanism (M. Chen dkk., 2025)	YOLOv11	Integrasi C3k2-Faster-EMA	**Hasil:** Penelitian ini mengembangkan YOLOv11n yang ditingkatkan melalui integrasi C3k2-Faster-EMA, SimAM attention, dan GIoU loss, menghasilkan model deteksi tomat yang lebih cepat, ringan, dan akurat. Model mencapai mAP 86.0%, meningkatkan kecepatan inferensi hingga 102.6 FPS, serta memperbaiki akurasi deteksi pada kondisi occlusion dan pencahayaan alami. Dengan performa ini, model menjadi solusi efektif untuk sistem pemanenan dan grading otomatis dalam pertanian cerdas (*intelligent agriculture*).
Comparing YOLOv11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment (Sapkota & Karkee, 2024; Zhang, 2023)	YOLOv8	-	**Hasil:** Menggunakan metode *transfer learning* dengan model YOLOv8 dan berhasil mendapatkan nilai mAP: 99.1% *Precision*: 98.1% *Recall*: 98% ***Gap*:** Model ini masih perlu diuji dalam kondisi pencahayaan dan lingkungan yang lebih bervariasi agar lebih *robust* dalam skenario dunia nyata.
Fruit ripeness identification using YOLOv8 model. Multimedia Tools and	YOLOv8, CenterNet	-	**Hasil:** Hasil menunjukkan bahwa model YOLOv8 lebih cepat dan lebih efisien dibandingkan dengan CenterNet dalam hal deteksi buah dengan nilai akurasi mencapai 99.5%.

Applications (Xiao dkk., 2024)			***Gap*:** Penelitian ini tidak mencakup semua variasi buah dan atau kondisi yang dapat terjadi di lapangan.
YOLO11: A Maturity Detection Model for Multi-Variety Tomato (Wei dkk., 2024)	YOLOv11	SPPFELAN	**Hasil:** Model GFS-YOLO11 menunjukkan peningkatan signifikan dalam kinerja dibandingkan dengan model asli, dengan peningkatan dalam metrik presisi (P), *recall* (R), mAP50, dan MAP50-95 sebesar 5.8%, 4.9%, 6.2%, dan 5.5%. Model ini juga berhasil mengurangi jumlah parameter dan beban komputasi sebesar 35.9% dan 22.5%, menjadikannya lebih efisien untuk aplikasi deteksi kematangan tomat secara *real-time*. Berikut adalah data hasil deteksi GFS-YOLO11 *precision* 92%, *recall* 86.8%, mAP50 93.4%, mAP50-95 83.6% ***Gap*:** Model ini perlu disesuaikan lebih lanjut untuk meningkatkan kinerjanya dalam kondisi lingkungan yang lebih variatif, seperti pencahayaan yang sangat rendah atau tumpang tindih yang lebih parah.
CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism (Appe dkk., 2023)	YOLOv5	CBAM (*Convolutional Block Attention Module*)	**Hasil:** YOLOv5 yang ditingkatkan dengan CBAM (*attention mechanism*) dan DIoU untuk deteksi kematangan tomat. Model ini berhasil meningkatkan mAP (88.1%), *precision* (87.3%), dan *recall* (86.9%) dibandingkan YOLOv5 standar ***Gap*:** Keterbatasan utamanya adalah *dataset* yang terbatas dan ketergantungan hanya pada analisis warna untuk menentukan kematangan.
Enhanced tomato detection in greenhouse environments: a lightweight model based on S-YOLO with high accuracy (X. Sun, 2024)	YOLOv8s	SE (*Squeeze-and-Excitation) Attention Module*	**Hasil:** Akurasi 96.60%, mAP@0.5 sebesar 92.46%, dan FPS 74.05, lebih baik dari YOLOv8s. ***Gap*:** Belum diuji pada *dataset* besar, masih bisa dioptimalkan untuk perangkat *edge computing*, dan belum mempertimbangkan fitur non-visual seperti tekstur atau kedalaman.
Performance Comparison of Cherry Tomato Ripeness Detection Using Multiple YOLO Models	YOLOv5, YOLOv8 (dengan *backbone* ResNet50)	-	**Hasil:** YOLOv8 (ResNet50) lebih unggul, dengan mAP 0.757, *precision* 0.707, dan *recall* 0.775, lebih tinggi dibandingkan YOLOv5. ***Gap*:** *Dataset* terbatas, belum menggunakan *segmentation*, dan belum diuji dalam kondisi dunia nyata.

(Yang & Ju, 2024)			
Tomato ripeness and stem recognition based on improved YOLOX (Li dkk., 2025)	YOLOX	SE (*Squeeze-and-Excitation) Attention Module*	**Hasil:** mAP 92.17%, *Precision* 87.07%, *Recall* 87.74%, dan F1 *Score* 87.25%, lebih unggul dibandingkan YOLOv4, YOLOv5, dan YOLOv7. ***Gap*:** Belum diuji di ladang terbuka, masih mengalami kesulitan dengan oklusi, tidak menggunakan data non-visual, dan perlu optimasi untuk perangkat *edge*.
Revolutionizing Agriculture: Real-Time Ripe Tomato Detection With the Enhanced Tomato-YOLOv7 System (J. Guo dkk., 2023)	YOLOv7	ODConv (*Omni-Dimensional Dynamic Convolution)*	**Hasil:** Menggunakan Tomato-YOLOv7 (versi YOLOv7 yang ditingkatkan) dengan struktur ReplkDext, FasterNet, dan ODConv. mAP@0.5 meningkat menjadi 89.3%, dengan FPS 42, tetap mendukung deteksi *real-time*. ***Gap*:** *Dataset* terbatas pada satu kondisi lingkungan, masih memiliki tantangan dalam mendeteksi objek yang sangat tertutup, dan belum diuji pada perangkat *edge*.
RSR-YOLO: a real-time method for small target tomato detection based on improved YOLOv8 network (Yue dkk., 2024)	YOLOv8n	*Gather and Distribute* (GD) *Mechanism*	**Hasil:** mAP@0.5 sebesar 90.7%, *Precision* 91.6%, *Recall* 85.9%, FPS 76, menunjukkan keseimbangan antara akurasi dan kecepatan inferensi. ***Gap*:** Kesulitan dalam deteksi objek yang sangat kecil, penurunan akurasi dalam kondisi pencahayaan ekstrem, dan belum diuji pada perangkat *edge*.
Real-time detection of fruit ripeness using the YOLOv4 algorithm (Widyawati & Febriani, 2021)	YOLOv4	-	**Hasil:** YOLOv4 dengan menggunakan PANet untuk fusi fitur *multi*-skala. Berhasil mendapatkan nilai mAP 87.6%, Akurasi rata-rata 87.6%, FPS 5, dengan deteksi pisang matang lebih akurat (93.33%) dibandingkan pisang mentah (81.9%). ***Gap*:** FPS masih rendah untuk aplikasi *real-time*, model belum diuji dalam kondisi pencahayaan ekstrem, akurasi deteksi pisang mentah masih kurang optimal, dan tidak ada segmentasi objek.
YOLOv8n-CA: Improved YOLOv8n Model for Tomato Fruit Recognition at Different Stages of	YOLOv8n	*Coordinate Attention (CA) Mechanism*	**Hasil:** *Coordinate Attention* (CA) *Mechanism* untuk meningkatkan ekstraksi fitur dan memperbaiki deteksi objek kecil. Digunakan di *backbone* dan *neck* untuk memperkuat deteksi fitur yang berhubungan dengan kematangan. CARAFE *Up-sampling* Operator Memperluas *receptive field* untuk

*Ripeness (*Gao dkk., 2025)			meningkatkan fusi fitur. SIoU *Loss Function* Mengatasi kekurangan CIoU dalam *loss function* asli YOLOv8n mAP@0.5 sebesar 97.3%, *Precision* 94.3%, *Recall* 92.5%, ukuran model hanya 4.90 MB, lebih ringan dari YOLOv8n tetapi lebih akurat. ***Gap*:** Kesulitan dalam mendeteksi objek dengan oklusi ekstrem, performa belum diuji dalam pencahayaan ekstrem, dan belum diuji dalam perangkat *edge computing*.
Detection, counting, and maturity assessment of blueberries in canopy images using YOLOv8 and YOLOv9(Deng dkk., 2024)	YOLOv8-l dan YOLOv9-c	-	**Hasil:** YOLOv9-c mengandalkan PGI dan GELAN untuk *propagation* informasi gradien. Nilai mAP@50 YOLOv8l mencapai 92.7%, dengan akurasi *counting error* 6.67% dan estimasi kematangan *error* 9.65%. ***Gap*:** Kesulitan dalam menangani *occlusion*, bias terhadap buah mentah, FPS rendah di perangkat *edge*, dan dataset masih terbatas.

Hasil telaah terhadap berbagai penelitian pada Tabel 2.1 menunjukkan bahwa mayoritas model deteksi kematangan tomat berbasis YOLO meningkatkan performanya melalui pemanfaatan modul *attention* (CA, SE, ECA, SimAM), *kernel* selektif (LSK, GHVC), perbaikan fusi fitur (BiAttFPN, HS-FPAN, FRM), serta penguatan *backbone* seperti UniRepLKNet dan C3k2-Faster-EMA. Pendekatan tersebut berhasil meningkatkan kemampuan ekstraksi fitur spasial, *multiscale*, dan kontekstual sehingga menghasilkan peningkatan *mAP*, *precision*, serta stabilitas deteksi. Namun, sebagian besar metode tersebut menambah jumlah parameter, meningkatkan nilai *GFLOPs*, atau memperlambat inference time, sehingga kurang sesuai untuk aplikasi real-*time* pada perangkat *edge* ataupun lingkungan pertanian yang membutuhkan efisiensi komputasi.

Berdasarkan gap tersebut, penelitian ini berkontribusi dengan menghadirkan blok C3k2_Lite sebagai varian ringan dari blok C3k2 pada YOLOv11. Blok ini mengombinasikan *Fast Context Unit* yang mampu menangkap informasi *multiscale* secara efisien dengan mekanisme *Cross-Gating Enhancement* yang memperkuat fitur relevan sekaligus menekan *noise* dari latar belakang kompleks. Pendekatan ini memperluas penelitian terdahulu dengan menawarkan peningkatan kapasitas selektivitas fitur tanpa menambah parameter secara signifikan. Dengan demikian, penelitian ini mengisi kesenjangan antara akurasi tinggi dan efisiensi komputasi, serta menjadi alternatif arsitektur yang lebih kompatibel untuk implementasi pada *edge device*. Berikutnya pada Tabel 2.2 memaparkan perbandingan algoritma yang digunakan penelitian sebelumnya menggunakan YOLO.

Tabel 2. 2 Perbandingan algoritma deteksi tingkat kematangan pada buah tomat

Penelitian	YOLOv4	YOLOv5	YOLOv7	YOLOv8	YOLOv9	YOLOX	YOLOv11
Tomato Ripening Detection in Complex Environments Based on Improved BiAttFPN Fusion and YOLOv11-SLBA Modeling (Hao dkk., 2025)	-	-	-	-	-	-	√
YOLO-PGC: A Tomato Maturity Detection Algorithm Based on Improved YOLOv11 (Q. Wu dkk., 2025)	-	-	-	-	-	-	√
URT-YOLOv11: A Large Receptive Field Algorithm for Detecting Tomato Ripening Under Different Field Conditions (Mu dkk., 2025)	-	-	-	-	-	-	√
Tomato ripeness detection method based on FasterNet block and attention mechanism (M. Chen dkk., 2025)	-	-	-	-	-	-	√
CAM-YOLO: tomato detection and classification based on improved YOLOv5 using combining attention mechanism (Appe dkk., 2023)	-	√	-	-	-	-	-

Enhanced tomato detection in greenhouse environments: a lightweight model based on S-YOLO with high accuracy (X. Sun, 2024)	-	-	-	√	-	-	-
Performance Comparison of Cherry Tomato Ripeness Detection Using Multiple YOLO Models (Yang & Ju, 2024)	-	√	-	√	-	-	-
Tomato ripeness and stem recognition based on improved YOLOX (Li dkk., 2025)	-	-	-	-	-	√	-
Revolutionizing Agriculture: Real-Time Ripe Tomato Detection With the Enhanced Tomato-YOLOv7 System (J. Guo dkk., 2023)	-	-	√	-	-	-	-
RSR-YOLO: a real-time method for small target tomato detection based on improved YOLOv8 network (Yue dkk., 2024)	-	-	-	√	-	-	-
Real-time detection of fruit ripeness using the YOLOv4 algorithm (Widyawati & Febriani, 2021)	√	-	-	-	-	-	-
*YOLOv8n-CA: Improved YOLOv8n Model for Tomato Fruit Recognition at Different Stages of Ripeness (*Gao dkk., 2025)	-	-	-	√	-	-	-
Detection, counting, and maturity assessment of blueberries in canopy images using YOLOv8 and YOLOv9(Deng dkk., 2024)	-	-	-	√	√	-	-