

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian terdahulu

Terdapat beberapa penelitian terkait dengan metode implementasi *deep learning* untuk melakukan deteksi dan hitung otomatis pada sebuah objek sebagai referensi perancangan dan pembangunan sistem deteksi dan hitung otomatis pada objek lalat buah yang menyerang salak.

2.1.1 YOLOv5, YOLO-X, YOLO-R, YOLOv7 Performance Comparison:

A Survey

Penelitian dengan judul “*YOLOv5, YOLO-X, YOLO-R, YOLOv7 Performance Comparison: A Survey*” yang dilakukan oleh Ismat Saira Gillani, Muhammad Rizwan Munawar, Muhammad Talha, Salman Azhar, Yousra Mashkoor, Muhammad Sami uddin dan Usama Zafar yang bertujuan untuk membandingkan kinerja hasil algoritma YOLOv5, YOLO-X, YOLO-R, dan YOLOv7 pada *dataset* SOTA video *game* sepak bola dalam mendeteksi pemain dan pergerakan secara *real-time*. Pada penelitian ini melakukan perbandingan pada aspek kecepatan (FPS) dan akurasi (AP).

Tabel 2. 1 Perbandingan Arsitektur YOLOv5, YOLO-X, YOLO-R, dan YOLOv7

	YOLOv5	YOLO-X	YOLO-R	YOLOv7
<i>Bacbone</i>	<i>CSPDarknet 53</i>	<i>Darknet 53</i>	<i>CSPDarknet 53</i>	<i>Extended CSPDarknet</i>
<i>Neck</i>	<i>PANet (Path Aggregation Network)</i>	<i>PANet</i>	<i>PANet</i>	<i>E-ELAN (Extended Efficient Layer)</i>

				<i>Aggregation Network)</i>
Head	<i>YOLO Layer</i>	<i>Decoupled head</i>	<i>YOLO layer with multi-task learning</i>	<i>Dual layer: Lead Layer dan Auxiliary Layer</i>
Activation Function	<i>Leaky ReLU, Sigmoid</i>	<i>Leaky ReLU, Sigmoid</i>	<i>Leaky ReLU, Sigmoid</i>	<i>SiLU (Sigmoid Linear Unit)</i>
Ekstraksi Fitur	<i>CSPDarknet</i>	<i>Advanced FPN dengan Decoupled Head</i>	<i>Explicit dan Implicit Knowledge Learning</i>	<i>Re-parameterized Convolutional Network</i>
Fitur Spesial	Dioptimalkan untuk kecepatan dan akurasi yang seimbang	<i>Anchor-free, decoupled head</i> untuk klasifikasi dan regresi	<i>Multi-task learning, kernel space alignment</i>	Peningkatan model dengan peningkatan kapasitas pembelajaran

Berdasarkan Tabel 2.3 masing-masing algoritma memiliki arsitektur yang berbeda-beda. Model YOLOv5 menjadi pilihan yang tepat untuk aplikasi *real-time* deteksi dan hitung otomatis yang memerlukan performa ringan dengan akurasi tinggi [17]. Poin penting yang digunakan pada penelitian tersebut adalah pemilihan model deteksi dan hitung otomatis menggunakan YOLOv5 dengan alasan sebagai berikut:

1. YOLOv5 menggunakan kombinasi fungsi aktivasi *Leaky ReLU* dan *Sigmoid*, yang dikenal efektif dalam menangani masalah *vanishing*

gradient dan membantu model belajar dari berbagai pola dalam data, termasuk dalam kondisi pencahayaan yang beragam.

2. Ekstraksi fitur berupa *CSPDarknet* sehingga mampu melakukan pengelolaan gradien dan mengurangi redundansi komputasi, yang memungkinkan model untuk mengekstraksi fitur lebih efisien. Hal ini penting dalam meningkatkan akurasi model terutama dalam mendeteksi objek kecil atau yang kompleks.
3. YOLOv5 memiliki fitur spesial untuk melakukan optimasi kecepatan dan akurasi yang seimbang memberikan bukti bahwa model menjadi pilihan terbaik untuk aplikasi yang ringan, *real-time* dengan akurasi tinggi seperti deteksi objek kecil.

2.1.2 A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture

Pada penelitian dengan judul “*A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture*” yang dilakukan oleh Yuanhong Zhong, Junyuan Gao, Qilun Lei dan Yao Zhou dengan tujuan untuk membantu manajemen hama dengan menyediakan informasi yang akurat dan tepat waktu tentang populasi serangga. Hal ini yang sangat penting untuk strategi pengendalian hama agar efektif. Model yang dikembangkan dalam penelitian ini adalah YOLO dan SVM. YOLO digunakan untuk melakukan deteksi dan perhitungan kasar serangga. Sedangkan, SVM melakukan klasifikasi dan perhitungan detail dari gambar yang telah diekstraksi.

Metode deteksi dan *counting object* yang digunakan di penelitian ini adalah kombinasi dua model YOLO dan SVM. Model ini menggabungkan kecepatan dan akurasi YOLO dalam deteksi awal dengan keakuratan klasifikasi dari SVM untuk menghasilkan sistem yang efektif dalam pemantauan hama di sektor Pertanian [18]. Poin penting yang digunakan

pada penelitian tersebut adalah penjelasan pengembangan model deteksi dan *counting object* yang dilakukan oleh YOLO sebagai berikut:

- *Dataset* yang digunakan pada penelitian ini memiliki resolusi yang cukup besar 3280x2464 sehingga perlu dilakukan pemrosesan. Pada tahap pemrosesan *dataset* dilakukan *resize* menjadi 448x448 untuk mengurangi jumlah komputasi. Dengan skema pengembangan model seperti penjelasan tersebut maka performa YOLO akan menghasilkan akurasi perhitungan 92,50%.
- Membagi *dataset* dengan komposisi 80:10:10, di mana 80% digunakan untuk pelatihan (*training*), 10% untuk validasi (*validation*), dan 10% untuk pengujian (*testing*) model.

2.1.3 Smart insect monitoring based on YOLOV5 case study: Mediterranean fruit fly *Ceratitis capitata* and Peach fruit fly *Bactrocera zonata*

Penelitian dengan “*Smart insect monitoring based on YOLOV5 case study: Mediterranean fruit fly *Ceratitis capitata* and Peach fruit fly *Bactrocera zonata**” yang dilakukan oleh S.O. Slim, I.A. Abdelnaby, M.S. Moustafa, M.B. Zahran, H.F. Dahi, dan M.S. Yones bertujuan untuk mengembangkan pendekatan berbasis *deep learning* yang bertujuan untuk mendeteksi dan mengkuantifikasi hama, khususnya *Bactrocera zonata* (lalat buah persik) dan *Ceratitis capitata* (lalat buah Mediterania), yang merupakan hama utama yang merusak buah-buahan secara global.

Pada penelitian ini menggunakan model YOLOv5 untuk melakukan klasifikasi, lokalisasi, dan penghitungan jumlah hama. Dalam melakukan monitoring digunakannya aplikasi ponsel cerdas yang dikembangkan untuk membantu petani dan profesional pertanian dalam manajemen dan pengobatan hama, dengan harapan dapat meningkatkan hasil pertanian

dengan memprediksi pilihan pengobatan secara *real-time*, mengurangi waktu dan sumber daya yang dibutuhkan untuk inspeksi lahan. Hasil penelitian menunjukkan peningkatan performa yang signifikan dengan akurasi rata-rata berbobot mencapai 84%, serta peningkatan presisi, *mean average precision* (mAP), dan *F1-score* hingga masing-masing 15%, 18%, dan 7% [19]. Poin penting yang digunakan dalam penelitian ini pemrosesan *dataset* sebagai berikut:

1. Penggunaan teknik transfer *learning* model YOLOv5s untuk melatih *dataset* dan menghasilkan performa model yang baik. Transfer *learning* adalah teknik yang efektif dalam *machine learning* agar model yang telah dilatih sebelumnya pada *dataset* besar dapat digunakan kembali untuk tugas yang berbeda, dengan melakukan sedikit pelatihan tambahan pada *dataset* baru.

2.1.4 A Smart Trap for Counting Olive Moths Based on the Internet of Things and Deep Learning

Penelitian dengan judul “*A Smart Trap for Counting Olive Moths Based on the Internet of Things and Deep Learning*” yang dilakukan oleh Afef Mdhaffar, Bechir Zalila, Racem Moalla, Ayoub Kharrat, Omar Rebai, Mohamed Melek Hsairi, Ahmed Sallemi, Hsouna Kobbi, Amel Kolsi, Dorsaf Chatti, Mohamed Jmaiel, and Bernd Freisleben bertujuan untuk mengembangkan dan menguji perangkat pintar yang dirancang untuk mendeteksi dan menghitung ngengat zaitun (*Prays oleae*) menggunakan teknologi *Internet of Things* (IoT) dan algoritme pembelajaran mendalam (*deep learning*). Pada penelitian ini menggunakan model YOLOv5 dan YOLOv7. Poin penting yang digunakan dalam penelitian ini adalah metrik evaluasi deteksi *object* yang dilakukan oleh sistem [20]. Metrik evaluasi untuk performa model dalam melakukan deteksi *object* meliputi :

1. *Precision*

Precision mengukur ketepatan model dalam mendeteksi objek. Objek yang dideteksi oleh model akan dihitung banyaknya yang benar dan sesuai.

2. *Recall*

Recall mengukur seberapa baik model dalam menemukan semua objek yang ada di gambar, sehingga memastikan model tidak melewatkan objek yang seharusnya dideteksi.

3. *Mean Average Precision (mAP)*

mAP cara untuk mengevaluasi keseluruhan kinerja model. Metrik ini menggabungkan nilai precision dan recall untuk melihat seberapa baik model mendeteksi semua objek dengan akurat.

2.1.5 A Deep-Learning Approach for Automatic Counting of Soybean Insect Pests

Penelitian dengan judul “*A Deep-Learning Approach for Automatic Counting of Soybean Insect Pests*” yang dilakukan oleh Everton Castelão Tetila , Bruno Brandoli Machado , Geazy Vilharva Menezes Nicolas Alessandro de Souza Belete , Gilberto Astolf, dan Hemerson Pistori bertujuan untuk membandingkan performa tiga model *deep learning* untuk mengetahui diagnosis dan jumlah hama pada lahan pertanian agar dapat melakukan pengendalian dengan tepat. Pada penelitian ini menggunakan tiga model yaitu *Dense-Net 201*, *Inception Resnet v2*, dan *Resnet 50*. Poin penting yang digunakan dalam penelitian ini adalah metode evaluasi hasil perhitungan otomatis yang dilakukan oleh sistem [21]. Metrik evaluasi untuk performa model dalam melakukan *counting object* meliputi :

1. *Average Detection Accuracy (ADA)*

ADA mengukur rata-rata akurasi model dalam mendeteksi objek yang ada di gambar. Hal ini untuk memastikan bahwa jumlah objek yang dihitung oleh model sesuai dengan jumlah sebenarnya.

2. *Average Precision (AP)*

AP untuk mengevaluasi kemampuan model untuk menghitung objek dengan benar tanpa terlalu banyak kesalahan atau kelalaian.

3. *F1-Score*

F1-Score untuk memastikan bahwa model tidak hanya menghitung objek dengan tepat, tetapi juga tidak melewatkan objek yang ada di gambar.

4. *Mean Absolute Error (MAE)*

MAE menunjukkan tingkat kesalahan dalam perhitungan jumlah objek. Semakin kecil nilai MAE, semakin akurat model dalam menghitung objek.

2.2 Tinjauan Teori

2.2.1 Lalat Buah Salak

Salak, atau lebih dikenal dengan nama "*snake fruit*" dalam bahasa Inggris, adalah salah satu buah tropis yang sangat khas dari Indonesia dan beberapa negara Asia Tenggara lainnya. Buah salak dikenal luas karena kulitnya yang bersisik menyerupai ular, yang menjadi asal mula nama "*snake fruit*" dalam bahasa Inggris. Daging buahnya berwarna krem atau kekuningan, dengan rasa manis yang khas disertai sedikit asam, menciptakan sensasi rasa yang unik dan memikat, mirip dengan perpaduan antara nanas dan apel.



Gambar 2. 1 Tanaman Salak Pondoh

Tanaman salak memiliki akar serabut dan menyerupai pohon palem dengan penampilan yang seolah-olah tidak berbatang, berukuran rendah dan tegak dengan tinggi antara 1,5 hingga 7 meter. Daun Salak Pondoh tersusun dalam bentuk roset, bersirip terputus-putus, dengan panjang antara 2,5 hingga 7 meter. Bagian bawah dan tepi tangkai daun dipenuhi duri tajam. Seperti yang terlihat pada Gambar 2.1, tanaman Salak Pondoh adalah tanaman berumah dua, di mana terdapat tanaman jantan dan betina. Buah Salak Pondoh umumnya lebih kecil dibandingkan dengan jenis salak lainnya. Buah ini memiliki variasi warna kulit mulai dari coklat kehitaman, coklat kemerahan, coklat kekuningan, hingga merah gelap kehitaman, dengan rasa yang secara konsisten manis [22].

Salah satu kendala utama dalam meningkatkan produksi salak adalah serangan lalat buah. Lalat buah dengan famili *Tephritidae* merupakan serangga yang sangat merak dan merugikan secara ekonomi pada produksi salak. Perdagangan internasional telah menyatakan bahwa lalat buah merupakan ancaman utama sebagai hama kontaminan dan bersifat spesies yang *invasif*. Keberadaan lalat buah menjadi perhatian utama dalam

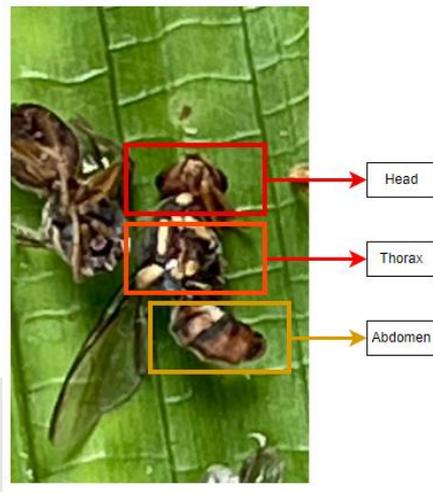
globalisasi perdagangan internasional. Telah banyak kasus penolakan ekspor salak oleh suatu negara pengimpor dikarenakan adanya gejala serangan lalat buah. Deteksi hama lalat buah ini sangat merugikan secara ekonomi nasional.



Gambar 2. 2 Lalat Buah

Pada gambar 2.2 dapat dilihat visual lalat buah yang menyerang tanaman salak. Lalat buah yang menyerang salak berbeda dengan lalat pada umumnya. Lalat buah ini memiliki ciri khas siluet warna kuning emas pada bagian tubuhnya. Dapat dilihat pada gambar 2.3 terdapat 3 bagian utama tubuh lalat buah yaitu kepala, *thorax*, dan abdomen atau perut. Toraks bagian tubuh lalat buah yang terletak antara kepala dan abdomen tersusun tulang dada, ruas tulang belakang, dan tulang rusuk dilengkapi dengan sayap [23].

2.2.2 Gejala Salak Busuk



Gambar 2. 3 Bagian Tubuh Lalat Buah

Lalat buah merupakan salah satu hama utama yang menyerang tanaman salak, yang berdampak signifikan terhadap kualitas dan kuantitas produksi salak, terutama yang ditujukan untuk ekspor. Serangan lalat buah menyebabkan percepatan proses pembusukan pada buah salak, yang dapat menurunkan kualitas dan nilai jualnya secara drastis. Salak yang terinfeksi oleh lalat buah umumnya menunjukkan gejala yang jelas, seperti tekstur buah yang menjadi lunak dan berair. Salah satu tanda yang paling mencolok adalah munculnya serbuk putih pada permukaan luar buah salak dapat dilihat Gambar 2.4, yang merupakan hasil dari aktivitas larva lalat buah di dalam buah.



Gambar 2. 4 Tampilan Luar Salak Gejala Lalat Buah



Gambar 2. 5 Tampilan Dalam Salak Gejala Lalat Buah

Ketika salak yang terinfeksi dibuka, akan terlihat tanda-tanda pembusukan yang lebih parah. Daging buah salak yang sehat biasanya berwarna putih krem, namun pada buah yang terinfeksi, daging buah berubah menjadi coklat pekat dan sering kali terdapat lubang-lubang kecil yang disebabkan oleh larva yang memakan daging buah dari dalam seperti pada Gambar 2.5. Kondisi ini tidak hanya mempengaruhi penampilan buah, tetapi juga membuatnya tidak layak untuk dikonsumsi atau dijual.

Penyebaran gejala lalat buah pada buah salak dapat terjadi dengan sangat cepat. Jika satu buah salak sudah terinfeksi dan bergesekan dengan buah lain atau jika serbuk putih dari buah yang terinfeksi menyebar ke buah yang sehat, maka proses kontaminasi akan berlangsung lebih cepat. Kondisi lingkungan yang lembab juga dapat mempercepat penyebaran dan perkembangan larva lalat buah, sehingga memperburuk kondisi tanaman salak secara keseluruhan. Dampak dari serangan lalat buah ini sangat serius, terutama bagi petani dan eksportir salak. Buah yang terinfeksi tidak hanya mengurangi hasil panen secara signifikan, tetapi juga dapat menyebabkan penolakan produk di pasar ekspor, yang pada akhirnya berdampak pada pendapatan petani dan reputasi produk salak di pasar internasional. Oleh karena itu, pengendalian lalat buah menjadi sangat penting untuk menjaga kualitas dan keberlanjutan produksi salak, khususnya yang ditujukan untuk ekspor.

2.2.3 Trapping

Lalat buah mampu menyebar dengan cepat. Salah satu pengendalian yang dapat dilakukan adalah dengan *trapping*. *Trapping* adalah pemerangkapan lalat buah pada suatu wadah dengan memanfaatkan zat pemikat *Methyl Eugenol*. Penggunaan zat pemikat ini merupakan cara yang praktis untuk mendapatkan spesimen lalat buah dalam jumlah yang cukup banyak pada waktu yang relatif singkat. Zat pemikat ini disebut sebagai *paraferomon* yang cara kerjanya menyerupai feromon yaitu dapat di deteksi oleh lalat buah jantan dalam jarak yang cukup jauh [24]. *Trapping* atau perangkap yang dipasang oleh petani sangat beragam seperti yang ditunjukkan pada Gambar 2.6. Lalat buah yang berhasil masuk ke perangkap maka akan segera mati.



Gambar 2. 6 Trapping Lalat Buah

2.2.4 Deep Learning

Deep learning atau pembelajaran mendalam merupakan cabang *machine learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. *Deep learning* cocok diterapkan pada aplikasi *supervised learning*, *unsupervised learning* dan *semi-supervised learning* maupun untuk *reinforcement learning* dalam berbagai aplikasi seperti pengenalan citra, suara, klasifikasi teks, dan sebagainya. *Deep Learning* adalah pembelajaran yang menggunakan fitur berbentuk hierarki,

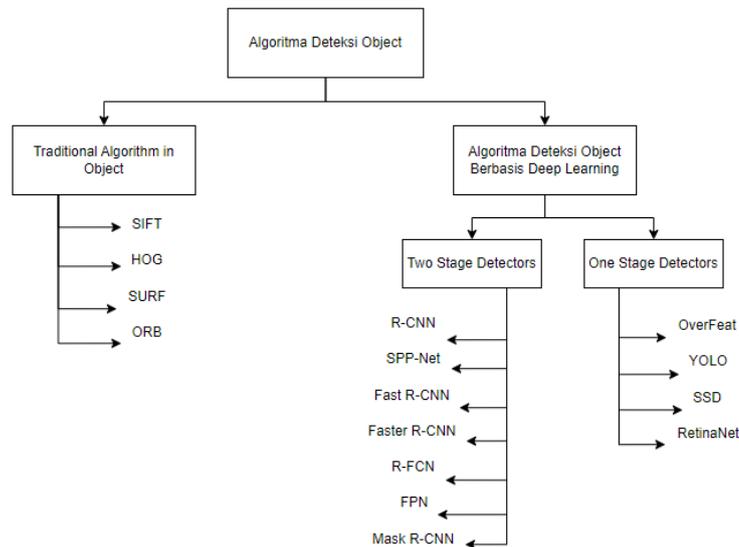
yang dapat diskalakan sesuai dengan kasus yang diproses. Algoritma *Deep Learning* mampu melakukan ekstraksi fitur otomatis dari data mentah dengan sangat detail, karena menggunakan struktur eksploitasi yang mengungkap fitur-fitur yang tidak terlihat secara kasat mata. Proses ini melibatkan transformasi fitur level tinggi menjadi fitur level rendah yang mudah dipahami oleh mesin pembelajaran, menjadikan *Deep Learning* efektif dalam menghasilkan representasi fitur yang baik dan optimal. *Deep Learning* juga menggabungkan pendekatan eksploitasi, yang fokus pada menggali fitur secara mendalam, dan eksplorasi, yang mengumpulkan sebanyak mungkin target *space* atau region.

Prinsip kerja dari algoritma *Deep Learning* melibatkan beberapa langkah penting yang harus diperhatikan. Pertama, penting untuk membuat arsitektur yang tepat, yang terdiri dari layer *input*, *hidden* (juga dikenal sebagai *darknet* atau layer tersembunyi), dan *output*, yang disesuaikan dengan kompleksitas kasus yang akan diolah. Selanjutnya, perlu disiapkan data atau item yang diperlukan untuk proses pelatihan dan pengujian *Deep Learning*. Pertimbangan juga harus diberikan pada tempat implementasi, apakah akan dilakukan secara lokal atau menggunakan *cloud*, serta bagaimana kode program akan dibuat apakah dari awal (*from scratch*), dengan menggabungkan beberapa kode dari *library*, atau menggunakan *library* sepenuhnya. Pengujian juga harus dilakukan sesuai dengan standar algoritma *Deep Learning*, mulai dari parameter hingga bentuk arsitekturnya, baik dengan pendekatan *waterfall* (sekuensial tanpa *loop*) atau *recycle* (sekuensial dengan *loop* tertentu). Pengujian *recycle* ini melibatkan pengulangan beberapa kali siklus untuk mengoptimalkan parameter sampai nilai akurasi yang stabil tercapai. Terakhir, penting untuk memikirkan cara untuk mengoptimalkan kembali algoritma *Deep Learning*

berdasarkan hasil yang telah diperoleh, guna mencapai performa yang lebih baik [25].

2.2.5 Object Detection

Object detection adalah salah satu tugas *computer vision* untuk menemukan objek tertentu dalam visual gambar atau video. Tujuan dari deteksi objek adalah untuk mengembangkan model atau teknik komputasi yang menyediakan informasi sederhana dibutuhkan oleh aplikasi *computer vision* seperti informasi jenis *object*, letak, dan lainnya. Akhir-akhir ini, penemuan objek menjadi bidang penelitian yang aktif dikembangkan. Teknik deteksi objek mengusulkan banyak algoritma yang dapat digunakan sesuai dengan kebutuhan, seperti yang ditunjukkan pada Gambar 2.7. Metode deteksi objek ada secara tradisional dan berbasis *deep learning* seperti yang ditunjukkan pada Gambar 2.7.



Gambar 2. 7 Algoritma Deteksi Objek

Pada *single stage detector*, deteksi dilakukan dalam satu langkah. *Single-stage detectors* umumnya lebih cepat karena mereka hanya memproses gambar sekali untuk melakukan prediksi. Salah satu algoritma termasuk

single stage detectors adalah YOLO, seperti yang ditunjukkan pada Gambar 2.7. Algoritma ini pertama kali diperkenalkan tahun 2015. Konsep utama YOLO adalah membagi gambar menjadi beberapa sel grid. Adanya algoritma klasifikasi dan lokalisasi yang diterapkan pada setiap sel retina memungkinkan label kelas untuk disesuaikan dengan visual objek dalam jaringan yang telah diidentifikasi oleh objek pusat. Tujuan dari pendekatan ini adalah untuk meningkatkan efisiensi pendeteksian dengan melatih model secara langsung pada seluruh gambar [26].

2.2.6 Object Counting

Salah satu aplikasi utama pada pemrosesan gambar digital adalah menghitung jumlah jenis objek tertentu dalam sebuah gambar. Menyelesaikan masalah perhitungan objek memiliki tantangan tersendiri. Kesulitan masalah ini antara lain kontras antara objek dengan latar belakang, tingkat pengelompokan objek, objek memiliki tekstur dan ukuran yang bervariasi, dan lainnya. Beragamnya tantangan yang dihadapi namun, dapat dipecahkan karena objek memiliki karakteristik yang sama. Alhasil, pemilihan strategi yang akan dikembangkan menjadi solusi tepat dalam menyelesaikan masalah ini. Ada beberapa kriteria yang dapat digunakan untuk mengklasifikasikan metode penghitungan objek secara otomatis, seperti jenis segmentasi atau jenis pengenalan pola yang diadopsi. Teknik yang digunakan bergantung pada kebutuhan pengembangan sistem khususnya karakteristik objek dan implementasinya. Salah satu contoh teknik *Artificial Neural Network* (RNN) cocok diimplementasikan untuk mendeteksi dan menghitung objek kecil. RNN membuat jenis khusus yang dinamakan dengan CNN. CNN (*Convolutional Neural Network*) juga banyak diimplementasikan dalam *counting object*. CNN sangat berguna dalam pemrosesan gambar digital karena fleksibilitas dan komputasinya yang cepat. Metode ini mampu mengenali pola yang rumit dan memisahkan

objek kecil dari latar belakang yang beragam dengan akurasi yang tinggi [27].

2.2.7 YOLOv5s

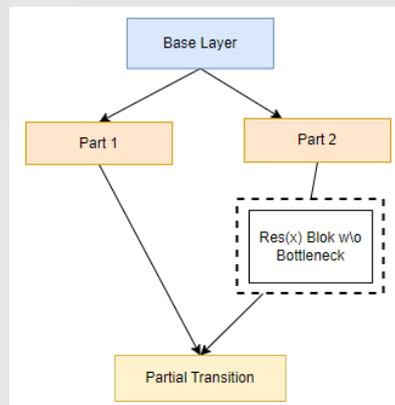
Algoritma YOLO menjadi seri model yang banyak digunakan dalam bidang deteksi objek karena memiliki performa akurasi tinggi dan kecepatan deteksi yang *relative* cepat. Terdapat beberapa algoritma YOLO sampai saat ini, salah satunya YOLOv5. YOLOv5 memiliki beberapa versi salah satunya YOLOv5s. Perbedaan YOLOv5s dengan model lainnya terletak pada jumlah layer, filter, dan parameter. Hal ini dapat dilihat pada Tabel 2.2 yang menunjukkan perbandingan antara YOLOv5s dan model lainnya.

Tabel 2.2 Perbandingan Model YOLOv5

Indikator		Model				
		YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Jumlah Layer		150	233	319	364	476
Jumlah Filter	Awal	32-64	32-64	64-128	128-256	128-256
	Akhir	128	512	1024	2048	2048
Jumlah Parameter		1.9 Juta	7 Juta	21 Juta	46 Juta	87 Juta

YOLOv5s memiliki sekitar 233 layer, yang mencakup berbagai jenis layer konvolusi, *batch normalization*, dan aktivasi yang digunakan dalam arsitektur model. Layer-layer ini membentuk *backbone* dan *head* untuk deteksi objek. Filter digunakan di berbagai layer konvolusi untuk mengekstrak fitur dari gambar, dengan layer-layer konvolusi awal menggunakan sekitar 32 atau 64 filter, yang kemudian meningkat menjadi 512 filter di layer-layer terakhir. YOLOv5s memiliki sekitar 7 juta parameter, yang mencakup bobot yang dipelajari selama proses pelatihan dan digunakan untuk mengonfigurasi model agar dapat mendeteksi objek dengan akurasi yang tinggi [28].

Arsitektur YOLOv5s, seperti halnya YOLOv5 biasa, terdiri dari beberapa komponen utama: *backbone*, *neck*, dan *head*. *Backbone* bertugas untuk mengekstrak fitur dari gambar *input*, *neck* digunakan untuk menggabungkan fitur-fitur tersebut, dan *head* berfungsi untuk memprediksi kotak pembatas (*bounding box*), kelas objek, dan skor kepercayaan. Pada arsitektur YOLOv5 terdapat bagian *CSP-Darknet 53* yang dirancang untuk mengurangi beban komputasi sehingga meningkatkan efisiensi kinerja model. Efisiensi kinerja model dilakukan dengan membagi *feature map* menjadi dua bagian dan memprosesnya secara paralel tanpa mengurangi hasil akurasi, seperti yang ditunjukkan pada Gambar 2.8. *CSP-Darknet 53* memanfaatkan blok *residual* yang ada dalam *Darknet* dengan cara yang lebih optimal. Blok *residual* membantu model dalam menangkap informasi yang lebih dalam dan kompleks tanpa memperlambat proses pelatihan atau inferensi.



Gambar 2. 8 CSPNet Model Structure

Setiap konvolusi Darknet-53 menggunakan struktur DarknetConv2D yang unik, dan regularisasi l2 dilakukan selama setiap konvolusi. Setelah konvolusi selesai, *BatchNormalization* distandarisasi dan fungsi aktivasi *Leaky Relu* dilakukan. Fungsi aktivasi Relu mengatur semua penugasan

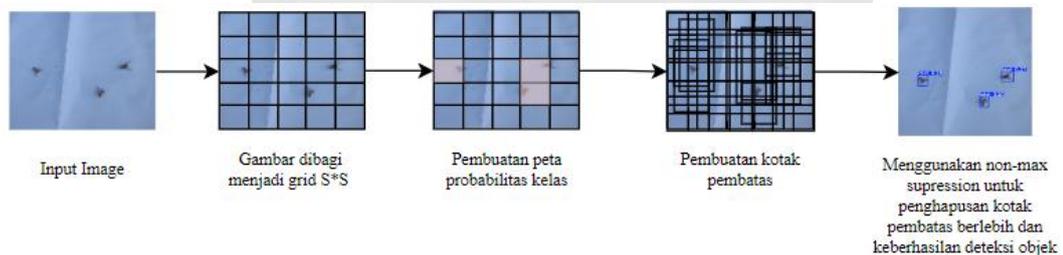
menjadi zero, dan *Leaky Relu* memberikan semua penugasan kemiringan bukan nol. Berikut rumus matematika yang digunakan:

$$y = \begin{cases} x_i, & \text{Jika } x_i \geq 0 \\ \frac{x_i}{a_i}, & \text{Jika } x_i < 0 \end{cases}$$

a_i adalah parameter dengan interval $(1, \infty)$. Sesuai dengan rumus matematika tersebut, maka *CSP-Darknet 53* menggunakan struktur *CSPNet* untuk membagi penumpukan menjadi dua bagian yaitu bagian utama untuk melanjutkan *stack original residual* dengan cepat dan bagian lainnya sama dengan *residual side* [13].

2.2.8 Deteksi dan Perhitungan Sistem

Metode deteksi YOLOv5 membagi gambar dengan *bounding box* ukuran $s \times s$ secara merata pada sumbu X dan sumbu Y, yang kemudian dibagi menjadi beberapa daerah dan melakukan prediksi berdasarkan *confidence value* yakni nilai kepercayaan prediksi objek dan probabilitas kelas. Pada tahap ini, teknik *non-max suppression* digunakan untuk menghilangkan kotak pembatas yang tumpang tindih (*redundan*) dengan menjaga hanya satu kotak pembatas dengan skor probabilitas tertinggi untuk setiap objek yang terdeteksi. Seperti pada Gambar 2.9 [29].



Gambar 2. 9 Proses Deteksi

Proses perhitungan objek dilakukan dengan menghitung label kelas hasil deteksi dari model YOLOv5 yang telah dilatih sebelumnya. Hasil label

kelas diperoleh dengan menyimpan file .txt selama proses deteksi. File .txt tersebut berisi hasil deteksi dalam bentuk koordinat kotak pembatas (bounding box) dan label kelas.

2.3 Simpulan

Berdasarkan studi dari referensi penelitian sebelumnya dan juga tinjauan dasar teori, maka penulis memiliki kesimpulan bahwa sistem deteksi dan hitung otomatis pada lalat buah salak yang akan dibuat memiliki acuan dan spesifikasi sebagai berikut:

1. Menjadikan rangkuman pada penelitian sebelumnya menjadi acuan dan referensi penelitian sistem yang akan dibangun
2. Penulis hanya melakukan deteksi dan hitung pada objek hama lalat buah. Selain objek tersebut tidak diteliti karena urgensi penelitian dilakukan untuk melakukan pengendalian hama lalat buah.
3. Dalam membangun sistem deteksi dan perhitungan otomatis penulis menggunakan algoritma YOLOv5 karena pilihan model yang memberikan hasil terbaik dan sesuai dengan kebutuhan sistem ringan dan cepat.
4. Sebelum masuk ke *training* model dilakukan *preprocessing* model dengan melakukan *resize* menjadi 448x448.
5. Membagi *dataset* dengan komposisi 80:10:10.
6. Pada proses pelatihan model YOLOv5 menggunakan teknik *transfer learning* model YOLOv5s sebelumnya.
7. Model yang dikembangkan akan dievaluasi dengan menggunakan *metric* evaluasi untuk deteksi objek dan *counting* objek.
8. Penggunaan *dataset* internal dan eksternal untuk meningkatkan kinerja model.

Berdasarkan daftar referensi berupa penelitian terdahulu dan tinjauan dasar teori yang telah penulis bahas pada bab 2 tinjauan teori, penulis menentukan teknologi

dan metode sesuai dengan sistem yang akan penulis rancang dan bangun. Penulis melakukan penimbangan dan penentuan solusi terbaik yang dipilih berdasarkan batasan penelitian ini. Penulis hanya mempertimbangkan pada keberhasilan hasil penelitian yang diukur dari kinerja performa model YOLOv5 yang dikembangkan untuk sistem deteksi dan hitung otomatis sesuai dengan penulisan latar belakang masalah dan batasan penelitian.

