

BAB I PENDAHULUAN

1.1 Latar Belakang

Padi merupakan komoditas pangan utama di Indonesia yang memiliki tahapan pertumbuhan spesifik dan terbagi ke dalam empat fase utama, yaitu *vegetatif awal* (V1), *vegetatif akhir* (V2), *reproduktif* (G1), dan *pematangan* (G2). Fase pertumbuhan padi menjadi indikator penting dalam menentukan strategi pemupukan, pengendalian hama, serta penjadwalan panen yang tepat. Kesalahan dalam pemantauan fase dapat menyebabkan pemupukan yang tidak sesuai, meningkatkan risiko serangan hama, serta menurunkan hasil panen. Oleh karena itu, identifikasi fase pertumbuhan padi secara akurat menjadi aspek krusial dalam sistem budidaya yang efisien dan produktif [1], [2].

Pentingnya pemantauan fase pertumbuhan ini semakin menonjol seiring dengan meningkatnya kebutuhan akan sistem pertanian yang adaptif dan responsif terhadap dinamika lingkungan serta ketersediaan sumber daya. Dalam era digitalisasi pertanian, penggunaan teknologi untuk mendukung proses pengambilan keputusan agronomis menjadi langkah strategis yang tidak dapat diabaikan. Salah satu tantangan besar dalam pengelolaan pertanian modern adalah bagaimana memperoleh data lapangan secara cepat, akurat, dan berkelanjutan, terutama pada skala lahan yang luas dan beragam kondisi [3].

Penyuluh pertanian berperan penting dalam membantu petani mengidentifikasi fase-fase pertumbuhan tersebut [4]. Di Balai Penyuluhan Pertanian (BPP) Karang Tengah, Kabupaten Cianjur, para penyuluh secara rutin memantau lahan sawah untuk menentukan langkah budidaya yang tepat, termasuk waktu panen yang optimal. Salah satu indikator yang mereka gunakan adalah fase pertumbuhan padi, karena informasi tersebut dapat dijadikan dasar dalam menentukan kebutuhan dan perlakuan terhadap tanaman pada waktu tertentu.

Namun, dalam pelaksanaannya, para penyuluh di BPP Karang Tengah masih menghadapi sejumlah tantangan di lapangan. Berdasarkan hasil diskusi dengan BPP Karang Tengah, Kabupaten Cianjur (terlampir pada Lampiran 1), terdapat beberapa keluhan utama yang dihadapi penyuluh, salah satunya adalah banyaknya lahan sawah yang harus mereka tangani. Kondisi ini membuat penyuluh kesulitan untuk mengingat secara detail umur tanam dari masing-masing lahan. Padahal, informasi umur tanam sangat krusial untuk menentukan fase pertumbuhan padi secara akurat. Hal ini menyebabkan proses identifikasi fase menjadi kurang konsisten dan berisiko menimbulkan kesalahan dalam penentuan perlakuan budidaya.

Keterbatasan tersebut mendorong kebutuhan akan solusi inovatif yang mampu mendukung kerja penyuluh secara efisien dan berbasis teknologi. Solusi ini tidak hanya harus mampu mengenali kondisi tanaman secara visual, tetapi juga mudah diakses dan terintegrasi dengan sistem pendukung keputusan pertanian yang sudah ada.

Pengembangan fitur baru pada *platform* KawalTani.id dilakukan untuk merespons kebutuhan tersebut melalui deteksi fase pertumbuhan padi berbasis citra digital. KawalTani.id merupakan *platform* hasil kolaborasi antara *Center of Excellence* (CoE) *Smart City* dan PT. Rastek Inovasi Digital (ID), yang bertujuan menghadirkan dukungan pertanian berbasis data secara *real-time*. Fitur ini dirancang untuk membantu penyuluh pertanian dalam mengidentifikasi fase pertumbuhan padi secara lebih akurat dan efisien, serta menjadi bahan pendukung dalam pengambilan keputusan budidaya yang lebih tepat dan berbasis data

Teknologi yang digunakan pada fitur ini memanfaatkan algoritma YOLOv8 (*You Only Look Once version 8*), yang dikenal memiliki kemampuan deteksi objek dengan tingkat akurasi tinggi. Seiring dengan kemajuan teknologi, beberapa penelitian menunjukkan efektivitas YOLOv8 dalam aplikasi pertanian, khususnya dalam deteksi fase pertumbuhan tanaman. Dalam penelitian yang dilakukan oleh Hayati dkk. (2023), YOLOv8 digunakan untuk mendeteksi dan menghitung kendaraan dengan hasil akurasi sebesar 86%, *Precision* 86%, *Recall* 86%, dan *F1-Score* 85%, membuktikan keandalan algoritma ini dalam mengidentifikasi objek bergerak di lingkungan kompleks secara *real-time* [5]. Sementara itu, Firdaus dkk. (2024) menerapkan YOLOv8 untuk mendeteksi lima jenis penyakit pada daun padi dan berhasil mencapai akurasi deteksi hingga 92% untuk penyakit blas daun [6]. Perbandingan ini menunjukkan bahwa YOLOv8 tidak hanya efektif dalam bidang transportasi, tetapi juga sangat potensial untuk diterapkan dalam sektor pertanian, terutama dalam mendeteksi kondisi tanaman berbasis citra digital.

Penerapan YOLOv8 di KawalTani.id dilakukan dengan mengunggah gambar padi dari lapangan, di mana sistem secara otomatis mengidentifikasi fase pertumbuhan, apakah itu V1, V2, G1, atau G2. Selain itu, sistem juga memberikan rekomendasi pemupukan, pengendalian hama, dan penjadwalan panen yang sesuai dengan fase yang terdeteksi. Prototipe fitur ini diharapkan dapat meringankan beban kerja penyuluh, mempercepat proses pemantauan, serta menjadi bahan pendukung dalam pengambilan keputusan agronomis yang lebih presisi dan berbasis teknologi digital.

1.2 Rumusan Masalah dan Solusi

Berdasarkan latar belakang tersebut, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana meningkatkan efisiensi dan menghitung akurasi dalam mendeteksi fase pertumbuhan padi yang selama ini dilakukan secara manual oleh penyuluh pertanian?
2. Bagaimana menyediakan akses atas fitur pengenalan fase pertumbuhan padi agar dapat digunakan oleh penyuluh pertanian

Dari permasalahan di atas, maka terdapat solusi sebagai berikut:

1. Menggunakan algoritma YOLOv8 untuk mendeteksi fase pertumbuhan padi secara cepat, sehingga hasil deteksi dapat langsung ditampilkan setelah foto diunggah, memungkinkan pemantauan *real-time* yang efisien.
2. Merancang prototipe deteksi visual yang terintegrasi dengan KawalTani.id, memungkinkan pengguna mengunggah gambar dan memperoleh hasil deteksi fase secara informatif dan praktis.

1.3 Tujuan

Berdasarkan rumusan masalah, tujuan yang akan dicapai adalah:

1. Membantu penyuluh pertanian dalam memperoleh informasi fase pertumbuhan padi secara efisien melalui deteksi berbasis citra digital.
2. Mengoptimalkan proses identifikasi fase pertumbuhan padi dengan memanfaatkan teknologi *deep learning*, sehingga dapat mengurangi ketergantungan pada metode observasi manual.

1.4 Batasan Masalah

Penelitian ini memiliki batasan-batasan untuk menjaga fokus dan cakupan tetap sesuai dengan tujuan yang telah ditetapkan. Batasan masalah yang ditetapkan adalah sebagai berikut:

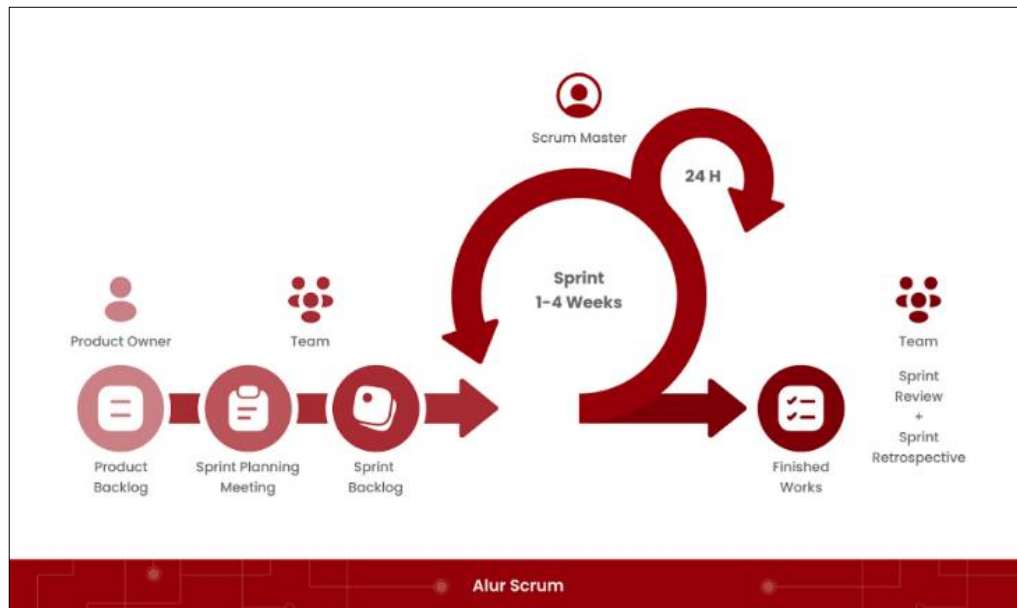
1. Penelitian ini hanya berfokus pada deteksi fase pertumbuhan tanaman padi dan tidak mencakup jenis tanaman pertanian lainnya.
2. *Dataset* penelitian ini dibatasi pada citra yang dikumpulkan menggunakan tiga perangkat kamera saja, yaitu dua unit kamera CCTV dan satu unit drone.

1.5 Metode Pengerjaan

Scrum adalah salah satu metode pengembangan perangkat lunak yang termasuk dalam pendekatan *Agile*. Metode ini dirancang agar proses pengembangan bisa berjalan secara bertahap, fleksibel, dan berfokus pada kolaborasi tim. Dalam *scrum*, proyek dibagi menjadi beberapa bagian kecil yang dikerjakan dalam waktu tertentu

yang disebut *sprint*. Tujuannya adalah agar pengembangan sistem bisa lebih mudah dipantau, cepat dievaluasi, dan cepat disesuaikan jika ada perubahan kebutuhan.

Scrum cocok digunakan untuk proyek-proyek yang membutuhkan perubahan atau penyesuaian berkelanjutan. Selain itu, metode ini juga sangat membantu dalam memastikan bahwa sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna, karena setiap hasil pengembangan selalu dievaluasi secara berkala [7].



Gambar 1. 1 Metode *Scrum*

Gambar 1. 1 menunjukkan alur kerja proses *Scrum*, yang menggambarkan bagaimana ide produk dikembangkan secara iteratif melalui tahapan-tahapan *Scrum* seperti perencanaan, pelaksanaan, evaluasi, hingga perbaikan berkelanjutan. Dalam penelitian ini, tahapan *scrum* yang diterapkan mencakup:

1. *Product Backlog*

Pada tahap ini, disusun daftar lengkap mengenai fitur, kebutuhan, dan fungsionalitas yang ingin dikembangkan dalam sistem. Daftar ini dikenal sebagai *product backlog*. Setiap item dalam *backlog* diurutkan berdasarkan prioritas, dan dapat diperbarui sewaktu-waktu jika ada perubahan kebutuhan.

2. *Sprint Planning*

Setelah *backlog* tersusun, dilakukan *sprint planning* untuk menentukan item-item yang akan dikerjakan dalam satu *sprint*. Di tahap ini, dilakukan identifikasi fitur atau komponen utama yang menjadi fokus pengembangan. Tugas-tugas dirinci agar lebih mudah dikerjakan secara bertahap.

3. *Sprint*

Sprint adalah periode waktu pengembangan aktif, biasanya berlangsung selama satu hingga dua minggu. Selama *sprint*, fokus diarahkan untuk menyelesaikan item yang telah direncanakan dalam *sprint planning*. Proses pengembangan dilakukan secara mandiri dengan tetap mengacu pada rencana dan target yang telah ditentukan sebelumnya.

4. *Sprint Review*

Setelah *Sprint* selesai, dilakukan *sprint review* untuk meninjau hasil pekerjaan selama *Sprint*. Pada tahap ini, dilakukan pengujian terhadap fitur yang telah dikembangkan, sekaligus mencatat umpan balik dan perbaikan yang mungkin diperlukan untuk iterasi berikutnya. Evaluasi ini menjadi dasar dalam menyusun *backlog* untuk *Sprint* selanjutnya.

1.6 Penjadwalan Kerja

Jadwal kerja disusun menggunakan metode *scrum* untuk mengarahkan pengembangan prototipe. Tabel 1. 1 memperlihatkan rincian tahapan kegiatan yang akan dilaksanakan selama periode tersebut.

Tabel 1. 1 Tabel Pelaksanaan Kerja

No	Deskripsi Kerja	Februari				Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Product Backlog</i>																				
	Menyusun <i>Product Backlog</i>																				
2	<i>Sprint Planning</i>																				
	Membuat <i>Sprint Planning</i>																				
3	<i>Sprint</i>																				
	<i>Sprint 1</i>																				
	<i>Sprint 2</i>																				
	<i>Sprint 3</i>																				
	<i>Sprint 4</i>																				
	<i>Sprint 5</i>																				
4	<i>Sprint Review</i>																				
	Meninjau Hasil Pekerjaan																				
5	Pengujian																				
	Pengujian Sistem																				
6	Dokumentasi																				
	Penyusunan Dokumentasi TA																				