BAB II LANDASAN TEORI

Bab ini membahas landasan teori dan penelitian terdahulu yang menjadi dasar dalam pelaksanaan penelitian. Pembahasan mencakup konsep-konsep yang berkaitan dengan penyakit daun teh, segmentasi citra, pendekatan *Human-in-the-Loop*, Segment Anything Model (SAM), RF-DETR-Seg, serta metode pendukung yang digunakan dalam proses pelatihan dan evaluasi model. Landasan teori tersebut digunakan untuk memberikan pemahaman terhadap pendekatan dan metode yang diterapkan dalam penelitian.

2.1 Landasan Teori

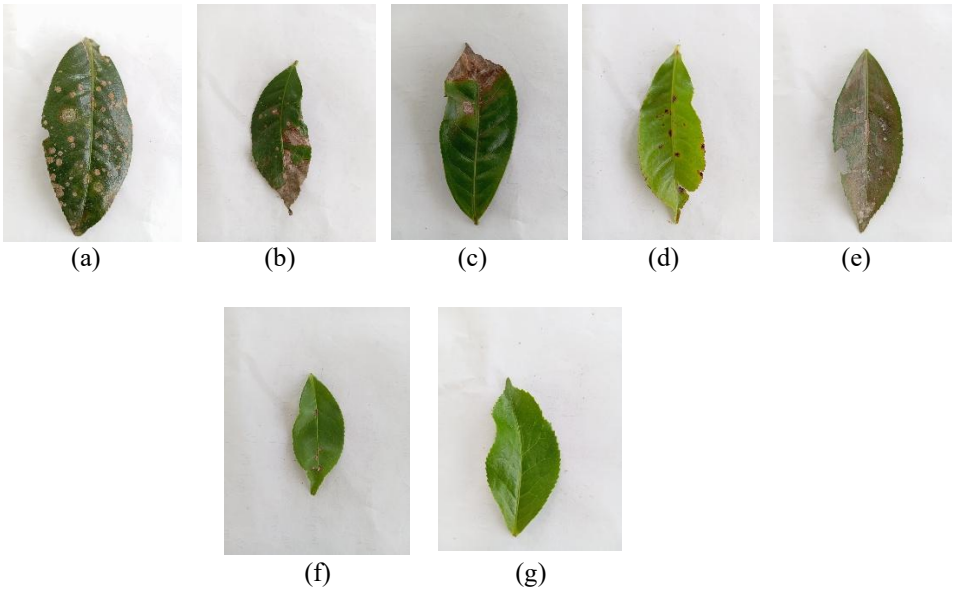
Landasan teori memuat konsep dan teori yang digunakan sebagai dasar dalam penelitian ini. Teori yang dibahas mencakup konsep-konsep yang berkaitan dengan objek penelitian, pendekatan anotasi, metode segmentasi, model yang digunakan, serta metode evaluasi yang mendukung pelaksanaan penelitian.

2.1.1 Tanaman Teh

Tanaman teh (*Camellia sinensis*) merupakan salah satu tanaman perkebunan yang memiliki nilai ekonomis dan banyak dibudidayakan di berbagai negara. Bagian tanaman yang terutama dimanfaatkan dalam produksi teh adalah pucuk yang terdiri atas daun muda dan tunas, yang dipanen sebagai bahan baku untuk menghasilkan berbagai jenis produk teh. Dengan demikian, kondisi kesehatan daun memiliki keterkaitan dengan bagian tanaman yang dimanfaatkan dalam proses produksi. Selain memiliki nilai ekonomi, teh juga mengandung berbagai senyawa bioaktif yang diketahui memiliki potensi manfaat bagi kesehatan (Bao dkk., 2022; Pamungkas & Supijatno, 2017). Industri teh juga berperan dalam perekonomian

negara-negara penghasil melalui kegiatan produksi, perdagangan, dan penyediaan lapangan kerja. Namun, produktivitas dan kualitas tanaman teh dapat dipengaruhi oleh berbagai faktor, termasuk kondisi lingkungan, pengelolaan tanaman, serta gangguan hama dan penyakit (Anjarsari dkk., 2021; Kumarihami & Song, 2018).

Hama dan penyakit merupakan salah satu gangguan yang dapat memengaruhi kondisi tanaman teh, terutama ketika serangan terjadi pada daun dan pucuk yang menjadi bagian penting dalam pemanenan. Beberapa penyakit foliar, seperti *Gray Blight* dan *Brown Blight*, dapat menimbulkan kerusakan pada jaringan daun dan berkontribusi terhadap penurunan hasil maupun kualitas tanaman teh (Pandey dkk., 2021). Serangan hama dan penyakit juga dapat menimbulkan karakteristik visual berupa perubahan warna, bercak, serta kerusakan jaringan pada permukaan daun (Alam dkk., 2025). Karakteristik visual tersebut menjadikan daun sebagai salah satu bagian tanaman yang dapat diamati untuk mengenali indikasi gangguan hama dan penyakit.



Gambar 2. 1 Jenis Penyakit Daun Teh: *(a) Algal leaf spot, (b) Brown blight, (c) Gray blight, (d) Helopeltis, (e) Red spider, (f) Green mirid bug, (g) Health* (Alam dkk., 2025)

Penyakit daun teh umumnya memiliki karakteristik visual yang beragam dan kompleks, sehingga dapat dibedakan berdasarkan perubahan warna, bentuk bercak, serta tekstur permukaan daun. Berdasarkan penelitian yang dilakukan oleh Alam, dkk. (2025) beberapa penyakit daun teh menunjukkan pola visual yang khas, seperti munculnya bercak coklat hingga keabu-abuan dengan bentuk tidak beraturan seperti pada penyakit *brown blight* dan *gray blight*. Selain itu, penyakit yang disebabkan oleh hama, seperti *red spider* dan *green mirid bug*, ditandai dengan bercak kecil berwarna gelap, perubahan warna daun menjadi kemerahan atau kecoklatan, serta kerusakan jaringan yang tidak merata pada permukaan daun (Alam dkk., 2025).

Variasi karakteristik visual tersebut menyebabkan tingkat kesulitan identifikasi dan penentuan batas lesi dapat berbeda pada setiap jenis penyakit. Beberapa lesi memiliki batas yang relatif jelas, sedangkan lesi lainnya dapat berukuran kecil, tersebar, memiliki bentuk tidak beraturan, atau menunjukkan perbedaan visual yang kurang tegas terhadap jaringan di sekitarnya. Kondisi tersebut menjadi tantangan dalam proses identifikasi dan segmentasi area penyakit pada citra daun teh. Oleh karena itu, pendekatan segmentasi dapat digunakan untuk merepresentasikan area lesi pada tingkat piksel sehingga bentuk dan luas area yang terindikasi penyakit dapat digambarkan secara lebih detail. Karakteristik masing-masing kelas penyakit dan gangguan hama daun teh yang digunakan dalam penelitian ini disajikan pada Tabel 2.1.

Tabel 2. 1 Karakteristik Penyakit Daun Teh (Alam dkk., 2025)

No	Nama Penyakit	Karakteristik
1	*Algal leaf spot*	a. Disebabkan oleh parasit alga *Cephaleuros* dan *Asterina* (Dipty dkk., 2025).

No	Nama Penyakit	Karakteristik
		b. Muncul lesi berwarna cokelat dengan bentuk tidak beraturan pada permukaan daun. c. Lesi dapat berkembang hingga 10 mm, dengan perubahan warna dari cokelat hingga abu-abu pada bagian tengah bercak dan tepi berwarna cokelat tua. d. Penyakit dapat muncul sebagai bercak cokelat berbentuk bulat atau tidak beraturan, yang pada tahap awal ditandai dengan penyusutan dan menguningnya daun.
2	*Brown blight*	a. Disebabkan oleh infeksi jamur foliar berat pada *Camellia sinensis.* b. Patogen utama berasal dari genus *Colletotrichum*, terutama *Collectotrichum camelliae*, spesies lain seperti *C. fructicola* dan *C. aenigma* juga dapat berperan. c. Ditandai dengan munculnya bercak kecil berwarna cokelat hingga kehitaman pada daun teh muda. d. Bercak awal bersifat basah, kemudian membesar, menggelap, dan saling menyatu, menyebabkan area daun mati ynag luas. e. Seiring waktu, terbentuk area jaringan daun mati besar yang menyebabkan daun mengering, layu, dan gugur sebelum waktunya. f. Jamur berkembang optimal pada kondisi hangat dan lembab, sehingga perkebunan teh di wilayah tropis dan subtropis lebih rentan terhadap penyakit ini.
3	*Gray blight*	a. Disebabkan oleh jamur *Pestalotiopsis theae*. b. Umumnya menyerang daun muda, dan daun yang terinfeksi cenderung gugur sebelum waktunya. c. Ditandai dengan munculnya lesi berwarna abu-abu kecokelatan pada permukaan daun, sering kali diawali sebagai bercak kecil yang kemudian membesar. d. Lesi memiliki margin (tepi) berwarna cokelat yang jelas. e. Seiring perkembangan penyakit, lesi saling menyatu membentuk area nekrotik luas yang menyebabkan gejala *leaf blight* dan nekrosis. f. Jamur berkembang optimal pada kondisi hangat dan lembab, spora berkecambah di permukaan

No	Nama Penyakit	Karakteristik
		daun, menembus jaringan, dan menyebabkan infeksi.
4	*Helopeltis*	a. Disebabkan oleh serangan serangga dari genus *Helopeltis,* terutama *Helopeltis theivora, Helopeltis antonii*, dan *Helopeltis bradyi*. b. Nimfa dan imago (dewasa) mengisap cairan dari daun, pucuk, dan tunas muda. c. Menimbulkan bercak berwarna cokelat hingga kehitaman pada permukaan daun akibat titik tusukan. d. Area yang terserang dapat berkembang menjadi bercak nekrotik. e. Daun dapat mengalami keriting, layu, dan gugur sebelum waktunya. f. Siklus hidup melibatkan peletakan telur pada bagian tanaman yang lunak, kemudian menetas menjadi nimfa dan berkembang menjadi serangga dewasa.
5	*Red spider*	a. Disebabkan oleh serangan tungau laba-laba merah pada daun teh. b. Tungau merusak sel pada permukaan atas daun, menyebabkan munculnya bercak kecil berwarna merah kecokelatan. c. Seiring peningkatan serangan, daun berubah warna menjadi kemerahan, mengering, rapuh, dan akhirnya gugur sebelum waktunya. d. Tungau berukuran sangat kecil, berwarna merah, dan sulit terlihat dengan mata telanjang umumnya berkoloni di bagian bawah daun serta membentuk jaring halus. e. Serangan lebih parah pada musim kering karena populasi tungau meningkat dengan cepat. f. Pada serangan berat, kerugian hasil dapat mencapai 40–50%, serta menurunkan kualitas rasa dan aroma daun teh.
6	*Green mirid bug*	a. Termasuk dalam famili *Miridae* (Serangga) dengan tipe mulut penusuk-pengisap untuk mengisap cairan tanaman. b. Menyerang daun, kuncup, dan pucuk teh dengan cara menusuk jaringan dan mengisap cairan. c. Menyebabkan perubahan warna, deformasi jaringan, serta nekrosis pada area yang terserang. d. Gejala visual berupa bercak cokelat hingga kehitaman pada daun yang dapat berujung pada gugur prematur.

No	Nama Penyakit	Karakteristik
		e. Serangan dapat menghambat pertumbuhan pucuk muda sehingga menurunkan produktivitas tanaman. f. Aktivitas hama meningkat pada bulan-bulan hangat dengan puncak infestasi pada musim tertentu. g. Dapat berpindah antar tanaman teh maupun gulma inang di sekitarnya. h. Bertelur pada tanaman teh, nimfa yang menetas langsung mulai makan dan, bersama individu dewasa, menyebabkan kerusakan signifikan dengan siklus reproduksi yang relatif cepat.
7	*Healthy*	a. Berwarna hijau cerah dengan permukaan halus dan seragam, tanpa bercak atau perubahan warna. b. Tekstur daun kokoh dan turgid, tidak menunjukkan gejala layu, keriting, maupun nekrosis. c. Pucuk dan tunas berkembang dengan baik, menunjukkan pertumbuhan yang konsisten. d. Tidak terdapat tanda serangan hama seperti bekas tusukan, bercak cokelat/kehitaman, atau eksudat cairan tanaman. e. Daun melekat kuat pada tanaman dan tidak mengalami gugur prematur.

2.1.2 Visi Komputer

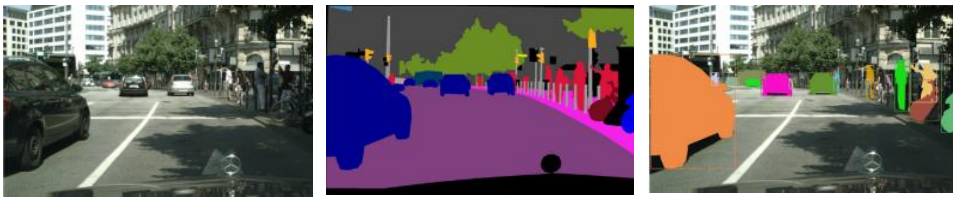
Visi Komputer (*Computer Vision*) merupakan cabang ilmu dari kecerdasan buatan yang memungkinkan komputer untuk menganalisis, merekam, memproses dan memahami makna informasi dalam gambar dan video digital (Upadhyay dkk., 2025). Tujuan utama dari visi komputer adalah mengekstraksi makna atau informasi penting dari gambar atau video digital berdasarkan karakteristik visual, seperti bentuk, pencahayaan, dan distribusi warna (Rasendriya, 2024).

Dalam bidang pertanian khususnya subsektor perkebunan, visi komputer telah banyak dimanfaatkan untuk mendukung otomatisasi dan pemantauan tanaman secara efisien. Teknologi ini memungkinkan pengamatan kondisi tanaman, deteksi

penyakit dan hama, serta analisis kualitas hasil panen berdasarkan informasi visual yang diperoleh dari gambar digital. Dibandingkan dengan metode konvensional yang bergantung pada pengamatan manual, pendekatan berbasis visi komputer dinilai lebih konsisten, cepat, dan mampu mengurangi ketergantungan terhadap tenaga ahli di lapangan yang memakan banyak biaya. Berbagai penelitian menunjukkan bahwa visi komputer berperan penting dalam meningkatkan efisiensi dan akurasi sistem pertanian modern, khususnya dalam proses deteksi penyakit tanaman yang memerlukan identifikasi gejala visual secara tepat dan berkelanjutan (Tian dkk., 2020; Upadhyay dkk., 2025).

2.1.3 Segmentasi Gambar

Segmentasi gambar merupakan salah satu bidang dalam visi komputer yang bertujuan untuk membagi sebuah gambar pada tingkat piksel sehingga dapat merepresentasikan objek secara efektif (Minaee dkk., 2020; Yu dkk., 2023). Hingga saat ini, segmentasi telah mengalami perkembangan yang cukup signifikan dan memainkan peran penting dalam berbagai bidang penelitian, seperti analisis gambar, persepsi robotik, diagnosa penyakit, monitoring video, *augmented reality*, dan *image compression* (Lei dkk., 2024).



(a) (b) (c)

(d)

Gambar 2. 2 Jenis Segmentasi: (a) Input gambar, (b) Segmentasi semantik, (c) Segmentasi instans, (d) Segmentasi panoptik (Gu dkk., 2022)

Tugas segmentasi secara umum diklasifikasikan ke dalam tiga sub-tugas yaitu segmentasi semantik, segmentasi instans dan segmentasi panoptik (Lei dkk., 2024). Segmentasi semantik bertugas dalam memberikan label pada setiap piksel berdasarkan kategori objek tertentu sehingga seluruh piksel dalam gambar memiliki kelas yang sama tanpa membedakan antar objek. Segmentasi instans merupakan pengembangan lebih lanjutan dari segmentasi semantik yang dimana dirancang untuk mengidentifikasi dan memisahkan setiap objek sebagai instans yang unik, meskipun berasal dari kelas yang sama. Selain itu, segmentasi panoptik mencakup segmentasi semantik dan segmentasi instans dengan memberikan label kelas sekaligus identitas instans pada setiap piksel dalam gambar (Gu dkk., 2022; Lei dkk., 2024; Minaee dkk., 2020).

Di bidang pertanian, segmentasi gambar telah banyak digunakan untuk pemantauan tanaman dan tanah, memprediksi waktu terbaik untuk menanam, memupuk hingga panen, memperkirakan hasil panen dan mendeteksi penyakit tanaman. Namun, segmentasi gambar pada gambar pertanian masih menghadapi berbagai tantangan, seperti kesulitan dalam mengenali tahapan perkembangan penyakit, inkonsistensi pelabelan data, serta perubahan morfologi tanaman akibat pengaruh lingkungan (Elbasi dkk., 2023; Lei dkk., 2024).

2.1.4 *Human-in-the-Loop*

Human-in-the-Loop (HITL) merupakan konsep yang menempatkan manusia sebagai bagian aktif dalam interaksi dengan sistem otomatis atau kecerdasan buatan. Istilah HITL telah digunakan dalam berbagai bidang dan memiliki pengertian yang berkembang sesuai dengan konteks penerapannya. Dalam konteks machine learning, keterlibatan manusia dapat berupa pemberian masukan, penilaian, maupun intervensi terhadap proses yang dijalankan oleh sistem. Pada HITL, penilaian manusia terintegrasi secara berkelanjutan dalam pelaksanaan dan pembagian tugas antara manusia dan sistem, sehingga manusia tidak hanya menerima hasil akhir tetapi juga berperan secara langsung dalam proses yang menghasilkan keluaran sistem (Anderson, 2022; Mosqueira dkk., 2023; Sherson dkk., 2023).

Selain HITL, terdapat beberapa konsep yang digunakan untuk menggambarkan perbedaan posisi manusia dalam interaksi dengan sistem otomatis, di antaranya *Human-on-the-Loop* (HOTL) dan *Human-out-of-the-Loop* (HOOTL). Pada HOTL, sistem menjalankan proses dengan tingkat otomatisasi yang relatif tinggi, sedangkan manusia terutama berperan dalam melakukan pengawasan dan dapat memberikan intervensi apabila diperlukan. Sementara itu, HOOTL menggambarkan kondisi ketika manusia tidak terlibat secara langsung dalam *loop* operasional suatu proses yang telah diotomatisasi. Meskipun demikian, batas antara HITL, HOTL, dan HOOTL tidak selalu bersifat mutlak karena definisi dan tingkat keterlibatan manusia dapat berbeda berdasarkan konteks penerapan sistem (Anderson, 2022; Sherson dkk., 2023).

Dalam penelitian ini, pendekatan HITL diterapkan karena anotator terlibat secara langsung dalam proses pembentukan anotasi dan tidak hanya mengawasi keluaran sistem. *Segment Anything Model* (SAM) digunakan sebagai alat bantu untuk menghasilkan kandidat *mask* berdasarkan *prompt* yang diberikan anotator. Selanjutnya, anotator memeriksa kesesuaian *mask*, memberikan interaksi atau koreksi apabila diperlukan, dan memvalidasi hasil akhir sebelum anotasi digunakan sebagai *ground truth*. Mekanisme tersebut menempatkan manusia sebagai bagian aktif dalam *loop* anotasi, sedangkan SAM berfungsi membantu proses segmentasi. Dengan demikian, HITL digunakan untuk memanfaatkan kemampuan otomatisasi SAM dengan tetap mempertahankan keterlibatan dan kontrol manusia terhadap hasil anotasi akhir.

2.1.5 Anotasi

Anotasi data merupakan proses pemberian label pada gambar untuk menyediakan informasi acuan (*ground truth*) yang digunakan dalam pelatihan dan evaluasi algoritma visi komputer. Anotasi berperan penting karena kualitas anotasi secara langsung memengaruhi kemampuan model dalam mempelajari representasi gambar dan menghasilkan prediksi yang akurat (Aljabri dkk., 2022).

(a) (b) (c)

Gambar 2. 3 Jenis Anotasi Gambar: (a) Anotasi tingkat gambar, (b) Anotasi bounding-box, (c) Anotasi tingkat piksel

Berdasarkan tingkat detailnya, anotasi gambar dapat dibedakan menjadi anotasi tingkat gambar (*image-level*), anotasi *bounding box*, dan anotasi tingkat piksel (*pixel-level annotation*). Untuk tugas segmentasi gambar, khususnya segmentasi instans, anotasi tingkat piksel merupakan bentuk anotasi yang paling relevan karena mampu merepresentasikan bentuk dan area objek secara presisi (Aljabri dkk., 2022). Anotasi jenis ini memungkinkan model mempelajari batas objek secara detail, yang tidak dapat dicapai oleh anotasi berbasis *bounding box*.

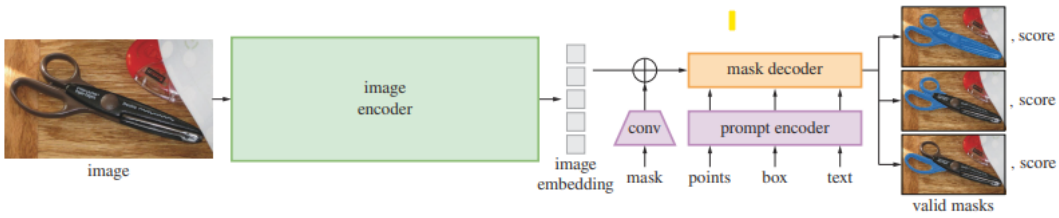
Meskipun penting, proses anotasi tingkat piksel memiliki tantangan yang signifikan, terutama dari segi waktu, biaya, dan konsistensi. Pada dataset berskala besar, anotasi manual menjadi proses yang mahal dan rentan terhadap subjektivitas, sehingga dapat memengaruhi kualitas *ground truth* yang dihasilkan (Pande, 2022).

2.1.6 *Segment Anything Model* (SAM)

Segment Anything Model (SAM) merupakan *foundation model* untuk tugas segmentasi yang dikembangkan dengan kemampuan *promptable segmentation* (Kirillov dkk., 2023). Sebelum berkembangnya SAM, beberapa metode *interactive segmentation* telah dikembangkan untuk membantu pengguna menghasilkan segmentasi melalui interaksi sederhana. *Deep Extreme Cut* (DEXTR) menghasilkan segmentasi objek berdasarkan empat titik ekstrem yang diberikan pengguna, sedangkan *Reviving Iterative Training with Mask Guidance* (RITM) menggunakan interaksi berupa klik positif dan negatif secara iteratif untuk menghasilkan dan memperbaiki hasil segmentasi. Pendekatan tersebut menunjukkan bahwa interaksi

pengguna dapat dimanfaatkan untuk membantu menentukan objek target dan menyempurnakan hasil segmentasi (Gool, 2016; Sofiiuk dkk., 2021).

SAM mengembangkan konsep segmentasi berbasis interaksi melalui kemampuan menerima berbagai jenis *prompt* untuk menentukan objek yang akan disegmentasi. SAM dapat menerima *prompt* berupa titik positif atau negatif, *bounding box*, *mask*, maupun informasi tekstual dalam formulasi modelnya, kemudian menghasilkan *mask* berdasarkan objek yang ditunjukkan oleh *prompt*. Model ini dirancang untuk memiliki kemampuan *zero-shot transfer*, sehingga dapat diterapkan pada berbagai tugas dan distribusi gambar baru tanpa harus melakukan pelatihan khusus untuk setiap tugas segmentasi yang dituju (Kirillov dkk., 2023).



Gambar 2. 4 Arsitektur *Segment Anything Model* (SAM) (Kirillov dkk., 2023)

Secara arsitektur, SAM terdiri dari tiga komponen utama, yaitu *image encoder*, *prompt encoder*, dan *mask decoder*. *Image encoder* berbasis *Vision Transformer* digunakan untuk menghasilkan representasi fitur dari gambar, sedangkan *prompt encoder* mengubah masukan pengguna menjadi representasi yang dapat diproses bersama fitur gambar. Selanjutnya, *mask decoder* menggabungkan representasi gambar dan *prompt* untuk menghasilkan prediksi *mask*. Arsitektur tersebut memungkinkan representasi gambar dihitung terlebih dahulu dan digunakan kembali ketika pengguna memberikan *prompt* yang berbeda, sehingga mendukung proses segmentasi secara interaktif. SAM dikembangkan menggunakan SA-1B,

yaitu dataset segmentasi berskala besar yang terdiri atas lebih dari 11 juta gambar dan sekitar 1,1 miliar *mask*, yang mendukung kemampuan generalisasi model pada berbagai distribusi gambar dan tugas segmentasi (Kirillov dkk., 2023).

Meskipun memiliki kemampuan generalisasi *zero-shot*, performa SAM dapat bervariasi ketika diterapkan pada domain dengan karakteristik yang berbeda dari data pelatihannya, terutama pada objek dengan batas yang sulit dibedakan atau karakteristik visual yang kompleks. Sejumlah penelitian pada domain khusus, termasuk citra medis dan pertanian, menunjukkan bahwa hasil segmentasi SAM masih dapat memerlukan interaksi atau penyesuaian lebih lanjut untuk memperoleh hasil yang sesuai dengan objek target (C. Zhang dkk., 2023; Y. Zhang dkk., 2024). Pada citra penyakit daun teh, SAM juga telah dimanfaatkan untuk mengekstraksi area yang terinfeksi sebelum dilakukan proses klasifikasi, yang menunjukkan potensinya dalam membantu segmentasi area penyakit pada domain pertanian (Balasundaram dkk., 2025).

Dalam penelitian ini, SAM digunakan sebagai alat bantu dalam proses anotasi interaktif dengan pendekatan *Human-in-the-Loop* (HITL). Anotator memberikan *prompt* untuk menentukan area target, kemudian SAM menghasilkan kandidat *mask* yang diperiksa dan disesuaikan apabila diperlukan sebelum divalidasi sebagai anotasi akhir. Penggunaan SAM pada tahap ini bertujuan untuk membantu pembentukan *mask* segmentasi, sementara keputusan mengenai kesesuaian batas objek dan validasi anotasi akhir tetap dilakukan oleh anotator.

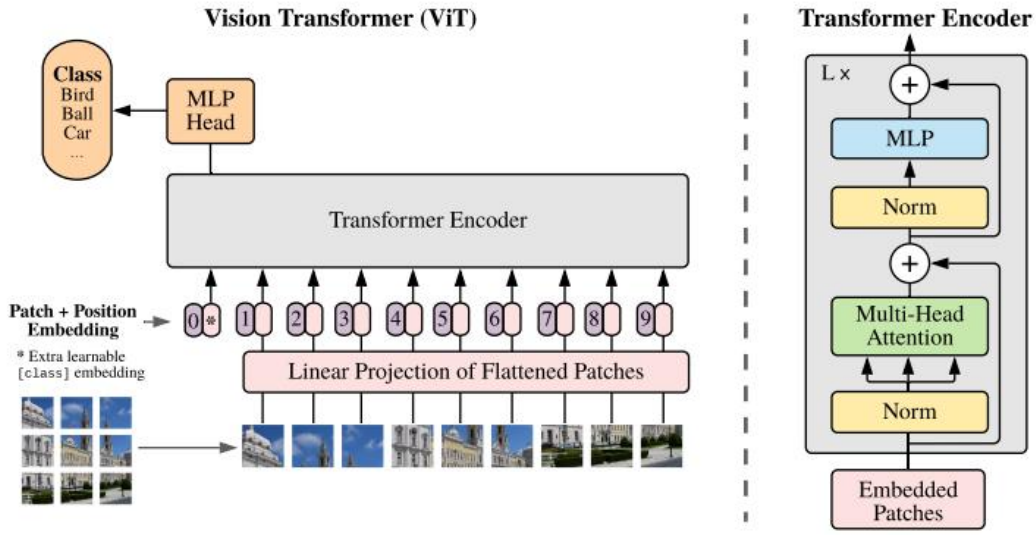
2.1.7 *X-AnyLabeling*

X-AnyLabeling merupakan perangkat lunak anotasi berbasis *open-source* yang mendukung proses pelabelan objek secara manual maupun semi-otomatis dengan integrasi model *Segment Anything Model* (SAM). *Tools* ini memungkinkan pengguna untuk menampilkan hasil *auto-segmentation* yang dihasilkan oleh SAM dalam bentuk *mask* polygon serta memberikan fleksibilitas kepada operator manusia untuk melakukan koreksi melalui penyesuaian titik-titik anotasi. Pendekatan ini bertujuan untuk meningkatkan presisi dan konsistensi *mask* segmentasi yang digunakan sebagai *ground truth* (CVHub520, 2023; Farook dkk., 2023).

2.1.8 *Vision Transformer* (ViT)

Transformer merupakan model *deep learning sequence-to-sequence* yang memproses data dalam bentuk urutan (*sequence*) dan menghasilkan urutan keluaran yang lain (Bi, 2021). Penelitian yang dilakukan oleh Vaswani dkk. (2017) melalui makalah berjudul *"Attention Is All You Need"* memperkenalkan arsitektur Transformer yang pada awalnya dikembangkan untuk pemrosesan bahasa alami (*Natural Language Processing* atau NLP). Perkembangan selanjutnya menghasilkan berbagai arsitektur berbasis *Transformer,* seperti *Bidirectional Encoder Representations from Transformers* (BERT) dan *Generative Pre-trained Transformer* (GPT) (Jamil & Piran, 2023; Vaswani, 2017). Kemampuan *Transformer* dalam memodelkan hubungan antarelemen melalui mekanisme *attention* kemudian mendorong penerapannya pada bidang visi komputer, termasuk

klasifikasi, deteksi objek, dan segmentasi citra (Bi, 2021; Jamil & Piran, 2023). Arsitektur umum *Vision Transformer* ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Arsitektur *Vision Transformer* (Zhai dkk., 2021)

Berbeda dengan arsitektur konvolusional yang memproses citra melalui operasi konvolusi, *Vision Transformer* (ViT) merepresentasikan citra sebagai sekumpulan token visual yang dapat diproses oleh *Transformer encoder*. Proses tersebut dimulai dengan membagi citra masukan menjadi sejumlah *patch* berukuran tetap. Untuk citra $X \in \mathbb{R}^{H \times W \times C}$ dengan ukuran *patch* $P \times P$, jumlah *patch* yang dihasilkan dinyatakan pada persamaan 2.1.

$$N = \frac{H}{P} \times \frac{W}{P} \tag{2. 1}$$

Setiap *patch* kemudian diratakan menjadi vektor berdimensi P^2C dan diproyeksikan secara linear ke ruang *embedding* berdimensi D (Zhai dkk., 2021). Proses tersebut dinyatakan pada persamaan 2.2.

$$z_i = x_p^i E + b \tag{2. 2}$$

dengan x_p^i merupakan *patch* ke-i yang telah diratakan, E merupakan matriks proyeksi yang dipelajari, b merupakan bias, dan z_i merupakan representasi embedding dari *patch* tersebut. Melalui proses ini, citra dua dimensi diubah menjadi urutan token dengan representasi pada persamaan 2.3.

$$Z = [z_1, z_2, \dots, z_N] \in \mathbb{R}^{N \times D} \quad (2.3)$$

Karena *Transformer* tidak secara langsung mempertahankan informasi mengenai posisi spasial setiap *patch*, informasi posisi ditambahkan melalui *positional embedding* (Vaswani, 2017; Zhai dkk., 2021). Penambahan tersebut memungkinkan model membedakan lokasi setiap token dan mempertahankan hubungan spasial antarbagian citra. Representasi awal yang diteruskan ke *Transformer encoder* dinyatakan pada persamaan 2.4.

$$Z_0 = Z + E_{pos} \quad (2.4)$$

dengan E_{pos} merupakan representasi posisi dari masing-masing token.

Selanjutnya, token diproses melalui serangkaian *Transformer encoder* yang terdiri atas mekanisme *Multi-Head Self-Attention* (MHSA) dan *Multi-Layer Perceptron* (MLP), disertai *layer normalization* dan *residual connection*. Mekanisme *self-attention* memungkinkan setiap token berinteraksi dengan token lainnya sehingga model dapat mempelajari hubungan antara berbagai bagian citra secara global (Vaswani, 2017; Zhai dkk., 2021). Pada mekanisme ini, representasi token diproyeksikan menjadi tiga komponen, yaitu *query* (Q), *key* (K), dan *value* (V) seperti yang disajikan pada persamaan 2.5.

$$Q = ZW_Q, K = ZW_K, V = ZW_V \quad (2.5)$$

dengan W_Q, W_K, dan W_V merupakan matriks bobot yang dipelajari selama proses pelatihan. Nilai *attention* kemudian dihitung berdasarkan tingkat kesesuaian antara *query* dan *key* menggunakan *scaled dot-product attention* seperti yang disajikan pada persamaan 2.6.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.\ 6)$$

dengan d_k merupakan dimensi *key* yang digunakan sebagai faktor penskalaan. Hasil perhitungan tersebut menentukan seberapa besar informasi dari token lain yang digunakan dalam memperbarui representasi suatu token. Pada MHSA, proses *attention* dilakukan secara paralel melalui beberapa *attention head*, sehingga model dapat mempelajari berbagai pola hubungan antar *patch* dari ruang representasi yang berbeda.

Hasil dari mekanisme *attention* selanjutnya diproses oleh MLP untuk melakukan transformasi nonlinier terhadap setiap representasi token (Zhai dkk., 2021). Secara umum, proses tersebut dirumuskan seperti pada persamaan 2.7.

$$\text{MLP}(x) = W_2\text{GELU}(W_1 x + b_1) + b_2 \quad (2.\ 7)$$

dengan W_1dan W_2 merupakan matriks bobot, b_1 dan b_2 merupakan bias, serta GELU merupakan fungsi aktivasi nonlinier. *Residual connection* dan *layer normalization* digunakan pada proses *attention* dan MLP untuk mempertahankan aliran informasi serta mendukung kestabilan proses pembelajaran. Secara umum, satu blok *Transformer encoder* dinyatakan pada persamaan 2.8 dan 2.9.

$$Z_l' = Z_{l-1} + \text{MHSA}(\text{LN}(Z_{l-1})) \quad (2.\ 8)$$

$$Z_l = Z_l' + \text{MLP}(\text{LN}(Z_l')) \quad (2.\ 9)$$

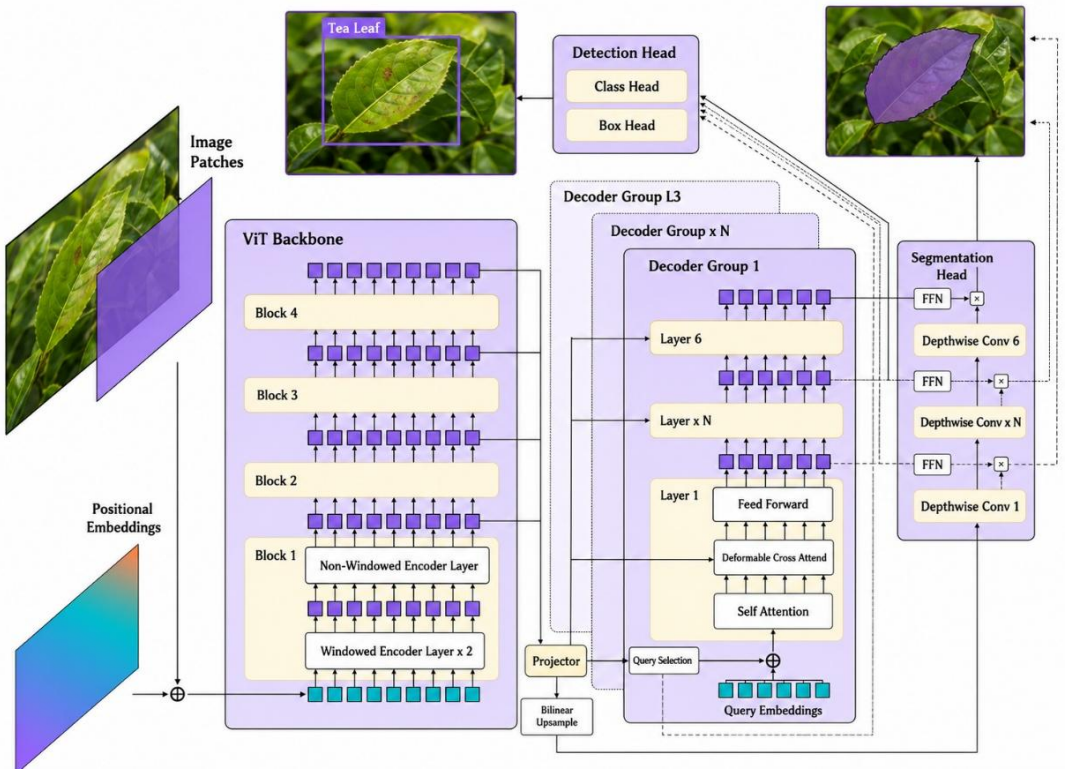
Proses tersebut dilakukan secara berulang melalui beberapa lapisan encoder sehingga representasi token secara bertahap memuat informasi visual dan hubungan kontekstual dari berbagai bagian citra. Dalam tugas segmentasi, kemampuan tersebut memungkinkan model memperoleh representasi fitur yang mempertimbangkan hubungan lokal maupun global antarwilayah citra, sehingga informasi mengenai bentuk, lokasi, dan karakteristik area objek dapat dipertahankan untuk proses prediksi berikutnya.

Pada penelitian ini, prinsip *Vision Transformer* menjadi dasar proses ekstraksi fitur pada *backbone* berbasis DINOv2 yang digunakan oleh RF-DETR (Oquab dkk., 2024). Representasi fitur yang dihasilkan dari beberapa lapisan *backbone* selanjutnya diproyeksikan dan digunakan oleh komponen RF-DETR untuk membentuk kandidat objek, memproses *object query* melalui *Transformer decoder*, serta menghasilkan prediksi kelas, *bounding box*, dan *mask* segmentasi. Mekanisme tersebut dibahas lebih lanjut pada bagian RF-DETR.

2.1.9 *Roboflow Detection Transformer* (RF-DETR)

2.1.8.1 Arsitektur RF-DETR

Roboflow Detection Transformer (RF-DETR) merupakan pengembangan lanjutan dari arsitektur *DEtection Transformer* (DETR) yang dirancang untuk memperoleh keseimbangan antara akurasi dan latensi melalui pendekatan *weight-sharing Neural Architecture Search* (NAS) berbasis *Transformer* (Robinson dkk., 2025). Secara umum, RF-DETR terdiri atas *backbone* berbasis *Vision Transformer* (ViT), pemrosesan fitur, *Transformer decoder*, dan *prediction head* yang bekerja secara *end-to-end* untuk menghasilkan prediksi objek. Arsitektur RF-DETR ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Arsitektur RF-DETR (Robinson dkk., 2025)

Proses RF-DETR diawali dengan ekstraksi fitur citra menggunakan *backbone* berbasis ViT dengan bobot *pre-trained* dari DINOv2. DINOv2 merupakan model

self-supervised yang dilatih menggunakan data berskala besar sehingga mampu menghasilkan representasi visual yang dapat digunakan pada berbagai tugas visi komputer (Oquab dkk., 2024). Backbone RF-DETR juga mengombinasikan mekanisme *windowed attention* dan *non-windowed attention* untuk memproses informasi visual secara efisien sekaligus mempertahankan kemampuan dalam menangkap konteks citra (Robinson dkk., 2025).

Berbeda dengan penggunaan satu representasi fitur akhir, RF-DETR memanfaatkan representasi fitur dari beberapa tingkat kedalaman *backbone* (Robinson dkk., 2025). Secara umum, kumpulan fitur tersebut dapat dinyatakan $\mathcal{F} = \{F^{(l_1)}, F^{(l_2)}, \dots, F^{(l_k)}\}$ dengan $F^{(l_k)}$ merupakan representasi fitur yang diperoleh dari lapisan *backbone* tertentu. Penggunaan fitur dari beberapa tingkat kedalaman memungkinkan model memanfaatkan representasi visual yang terbentuk pada tahapan berbeda dalam *backbone.*

Representasi fitur tersebut selanjutnya diproses melalui *feature projector* untuk menyelaraskan dan menggabungkan fitur ke dalam dimensi representasi yang digunakan oleh komponen *Transformer* RF-DETR (Robinson dkk., 2025). Proses tersebut secara umum dinyatakan pada persamaan 2.10.

$$F_{\text{proj}} = \phi\left(F^{(l_1)}, F^{(l_2)}, \dots, F^{(l_k)}\right) \tag{2. 10}$$

dengan ϕ merupakan fungsi proyeksi fitur dan F_{proj} merupakan representasi fitur hasil proyeksi. Tahap ini menjadi penghubung antara fitur visual yang dihasilkan *backbone* dan proses pembentukan representasi objek pada *Transformer decoder.*

Dari representasi fitur tersebut, RF-DETR membentuk kandidat objek yang digunakan untuk menginisialisasi *object query* (Carion dkk., 2020; Robinson dkk., 2025). Setiap *object query* merepresentasikan kandidat instans yang akan diproses dan diperbarui oleh *Transformer decoder*. Sekumpulan *object query* dinyatakan pada persamaan 2.11.

$$Q^{(0)} = [q_1, q_2, \dots, q_N] \tag{2. 11}$$

dengan N merupakan jumlah *object query* dan setiap q_i merupakan representasi fitur dari kandidat objek. Dengan demikian, *decoder* tidak memproses seluruh lokasi citra secara langsung sebagai keluaran objek, tetapi menggunakan sejumlah *query* untuk merepresentasikan kandidat instans yang akan diprediksi.

Selanjutnya, *object query* diproses secara bertahap melalui beberapa lapisan *Transformer decoder* (Carion dkk., 2020; Robinson dkk., 2025). Setiap lapisan *decoder* melibatkan mekanisme *self-attention*, *cross-attention*, dan *feed-forward network* (FFN). *Self-attention* memungkinkan terjadinya interaksi antar-*object query*, sehingga setiap *query* dapat mempertimbangkan informasi dari kandidat objek lainnya (Carion dkk., 2020; Vaswani, 2017). Sementara itu, *cross-attention* menghubungkan *object query* dengan representasi fitur citra sehingga informasi visual yang relevan dapat digunakan untuk memperbarui representasi setiap *query*. Alur umum pembaruan *query* pada lapisan *decoder* dapat dinyatakan sebagai berikut:

$$Q^{(l)} \rightarrow \text{SelfAttention} \rightarrow \text{CrossAttention} \rightarrow \text{FFN} \rightarrow Q^{(l+1)}$$

Pada proses *cross-attention*, RF-DETR menggunakan mekanisme *deformable attention* untuk mengambil informasi dari sejumlah lokasi relevan pada *feature map*

berdasarkan *reference point* setiap *query* (Zhu dkk., 2021). Secara umum, proses tersebut dirumuskan seperti pada persamaan 2.12.

$$\text{DeformAttn}(q_i, p_i, F) = \sum_{h=1}^{H_a} \sum_{R=1}^{K} A_{ihk} F(p_i + \Delta p_{ihk}) \tag{2. 12}$$

dengan q_i merupakan *object query*, p_i merupakan *reference point*, Δp_{ihk} merupakan *sampling offset*, A_{ihk} merupakan bobot *attention*, dan F merupakan representasi fitur citra. Melalui mekanisme ini, setiap *query* dapat mengambil informasi visual dari lokasi-lokasi yang relevan terhadap kandidat objek tanpa harus memberikan perhatian yang sama terhadap seluruh lokasi pada *feature map* (Zhu dkk., 2021).

Representasi *query* yang telah diperbarui kemudian diteruskan ke lapisan *decoder* berikutnya sehingga informasi mengenai karakteristik dan lokasi kandidat objek diperbaiki secara bertahap (Carion dkk., 2020; Robinson dkk., 2025). Secara umum, proses pembaruan tersebut dapat dinyatakan sebagai berikut.

$$Q^{(0)} \rightarrow Q^{(1)} \rightarrow \cdots \rightarrow Q^{(L)}$$

dengan L merupakan jumlah lapisan decoder dan $Q^{(L)}$ merupakan representasi query akhir. Pada RF-DETR, konfigurasi arsitektur decoder dirancang dengan pendekatan *weight-sharing Neural Architecture Search* (NAS) untuk memperoleh keseimbangan antara performa prediksi dan efisiensi komputasi (Robinson dkk., 2025).

Representasi *query* akhir selanjutnya digunakan oleh *prediction head* untuk menghasilkan prediksi kelas dan lokasi objek (Carion dkk., 2020; Robinson dkk.,

2025). Prediksi kelas diperoleh melalui proyeksi linear terhadap setiap representasi *query* seperti pada persamaan 2.13.

$$z_i = W_{\text{cls}} q_i + b_{\text{cls}} \quad (2.\ 13)$$

dengan q_i merupakan representasi *query*, W_{cls} dan b_{cls} merupakan parameter yang dipelajari, serta z_i merupakan logit kelas untuk kandidat objek ke-i. Sementara itu, prediksi *bounding box* diperoleh melalui jaringan MLP seperti yang disajikan pada persamaan 2.14 dan 2.15.

$$b_i = \text{MLP}_{\text{bbox}}(q_i) \quad (2.\ 14)$$

dengan:

$$b_i = (c_{x,i}, c_{y,i}, w_i, h_i) \quad (2.\ 15)$$

yang merepresentasikan koordinat pusat, lebar, dan tinggi *bounding box* (Carion dkk., 2020). Dengan demikian, setiap *object query* yang telah diproses oleh decoder menghasilkan representasi yang digunakan untuk menentukan kategori dan lokasi kandidat objek.

2.1.8.2 *Weight-sharing Neural Architecture Search* (NAS)

Weight-sharing Neural Architecture Search (NAS) merupakan pendekatan pencarian arsitektur yang memungkinkan berbagai konfigurasi atau subarsitektur dievaluasi secara efisien melalui mekanisme berbagi bobot (*weight sharing*) dalam satu supernet, sehingga setiap kandidat tidak perlu dilatih secara terpisah dari awal (Xie dkk., 2020). Pada RF-DETR, pendekatan *weight-sharing* NAS digunakan untuk mengeksplorasi berbagai konfigurasi arsitektur dengan ruang pencarian yang mencakup ukuran *patch* pada *backbone*, resolusi input, konfigurasi *window attention*, jumlah lapisan *decoder*, dan jumlah token *query*. Selama proses

pencarian dan pelatihan, konfigurasi subarsitektur disampel dari ruang pencarian dan menggunakan parameter yang dibagikan dalam supernet. Pendekatan ini meningkatkan efisiensi pencarian arsitektur sekaligus memberikan efek regularisasi melalui pelatihan berbagai konfigurasi model (Robinson dkk., 2025). Hasil proses pencarian tersebut digunakan untuk menentukan konfigurasi arsitektur yang memberikan keseimbangan antara akurasi dan efisiensi komputasi. Dengan demikian, NAS berperan pada tahap perancangan dan pemilihan konfigurasi arsitektur RF-DETR, sedangkan model yang telah ditentukan menggunakan konfigurasi arsitektur tetap pada proses pelatihan lanjutan dan inferensi.

2.1.8.3 RF-DETR-Seg

RF-DETR-Seg merupakan perluasan dari RF-DETR yang dirancang untuk melakukan segmentasi instans secara *end-to-end* dengan menambahkan *lightweight instance segmentation head*. Selain menghasilkan prediksi kelas dan bounding box, arsitektur ini memanfaatkan representasi *object query* dari *Transformer decoder* dan fitur spasial citra untuk menghasilkan *mask* pada setiap instans. Fitur spasial terlebih dahulu disesuaikan ke resolusi *mask* menggunakan interpolasi bilinear, kemudian diproses dan diproyeksikan menjadi representasi fitur piksel. Secara umum, proyeksi fitur spasial dan fitur *query* dinyatakan pada persamaan 2.16.

$$\tilde{F} = \phi_s(F)$$
$$\tilde{q}_i = \phi_q(q_i) \quad (2.16)$$

dengan F merupakan fitur spasial citra, ϕ_s merupakan fungsi transformasi dan proyeksi fitur spasial, q_i merupakan representasi *object query* ke-i, dan ϕ_q merupakan fungsi transformasi dan proyeksi *query*. Hasil proyeksi tersebut berada

pada ruang fitur dengan dimensi yang sama sehingga dapat digunakan untuk membentuk prediksi *mask*.

Mask untuk setiap *object query* kemudian diperoleh melalui operasi *dot-product* antara representasi *query* dan representasi fitur pada setiap lokasi spasial. Nilai *mask* logit untuk *query* ke-i pada lokasi (x, y) dirumuskan pada persamaan 2.17.

$$m_{i,x,y} = \sum_{c=1}^{d} \tilde{q}_{i,c} \, \tilde{F}_{c,x,y} + b \tag{2. 17}$$

dengan d merupakan dimensi fitur, $\tilde{q}_{i,c}$ merupakan elemen ke-c dari representasi *query* ke-i, $\tilde{F}_{c,x,y}$ merupakan fitur pada channel ke-c dan lokasi spasial (x, y), serta b merupakan bias. Nilai *dot-product* tersebut merepresentasikan kesesuaian antara karakteristik instans yang direpresentasikan oleh *object query* dan fitur visual pada setiap lokasi spasial. Dengan demikian, setiap *query* dapat menghasilkan *mask* yang berkaitan dengan instans yang sama dengan prediksi kelas dan *bounding box*-nya. Mekanisme ini memungkinkan RF-DETR-Seg menghasilkan keluaran berupa kelas, lokasi objek, dan *mask* instans dalam satu alur pemrosesan *end-to-end* (Robinson dkk., 2025).

RF-DETR-Seg juga menerapkan proses *pre-training* menggunakan dataset *Objects365* yang dilengkapi dengan *mask* instans yang dihasilkan menggunakan SAM2. Proses *pre-training* tersebut memberikan representasi awal mengenai karakteristik visual dan bentuk objek sebelum model diterapkan pada tugas segmentasi target. Dengan memanfaatkan representasi hasil *pre-training* tersebut, model dapat diadaptasi melalui proses *fine-tuning* untuk melakukan segmentasi pada dataset yang digunakan dalam penelitian (Robinson dkk., 2025).

2.1.10 Metrik Evaluasi

2.1.10.1 *Precision*

Precision merupakan metrik evaluasi yang digunakan untuk mengukur proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan model. Dalam konteks deteksi dan segmentasi, precision menunjukkan seberapa banyak prediksi objek atau area penyakit yang dinyatakan terdeteksi oleh model benar-benar sesuai dengan objek yang seharusnya terdeteksi. Nilai precision yang tinggi menunjukkan bahwa model menghasilkan sedikit *false positive*, sehingga prediksi yang diberikan cenderung lebih tepat (Liu dkk., 2025; Terven dkk., 2025). Secara matematis, precision dirumuskan pada Persamaan 2.1.

$$Precision = \frac{TP}{TP + FP} \tag{2. 18}$$

Keterangan:

Precision	:	Nilai presisi model.
TP (*True Positive*)	:	jumlah prediksi positif yang benar.
FP (*False Positive*)	:	jumlah prediksi positif yang salah.

2.1.10.2 *Recall*

Recall merupakan metrik evaluasi yang digunakan untuk mengukur proporsi objek positif yang berhasil terdeteksi dengan benar oleh model terhadap seluruh objek positif yang sebenarnya ada pada data. Dalam konteks deteksi dan segmentasi penyakit daun, recall menunjukkan kemampuan model dalam menemukan area penyakit yang memang termasuk ke dalam target anotasi. Nilai *recall* yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar objek atau area penyakit yang seharusnya teridentifikasi, sehingga jumlah *false negative* yang

dihasilkan relatif kecil (Liu dkk., 2025; Terven dkk., 2025). Secara matematis, recall dirumuskan pada Persamaan 2.2

$$Recall = \frac{TP}{TP + FN} \tag{2. 19}$$

Keterangan:

Recall	:	Nilai *recall* model.
TP (*True Positive*)	:	jumlah prediksi positif yang benar.
FN (*False Negatif*)	:	jumlah objek positif yang gagal terdeteksi oleh model.

2.1.10.3 *F1-Score*

F1-Score merupakan metrik evaluasi yang digunakan untuk mengukur keseimbangan antara precision dan recall melalui rata-rata harmonik dari kedua metrik tersebut. Dalam konteks deteksi dan segmentasi penyakit daun, *F1-Score* memberikan gambaran performa model secara lebih seimbang, karena tidak hanya mempertimbangkan ketepatan prediksi positif, tetapi juga kemampuan model dalam menemukan seluruh objek atau area penyakit yang seharusnya terdeteksi. Nilai *F1-Score* yang tinggi menunjukkan bahwa model mampu mencapai *precision* dan *recall* yang sama-sama baik, sehingga hasil deteksi dan segmentasi menjadi lebih konsisten (Schlosser dkk., 2024; Terven dkk., 2025). Secara matematis, F1-Score dirumuskan pada Persamaan 2.3

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{2. 20}$$

Keterangan:

F1-Score	:	Nilai *F1-Score*
Precision	:	Rasio prediksi positif yang benar terhadap seluruh prediksi positif.

Recall : Rasio objek positif yang berhasil terdeteksi terhadap seluruh objek positif sebenarnya.

2.1.10.4 *Intersection over Union* (IoU)

Intersection over Union (IoU) adalah metrik fundamental yang sering digunakan untuk mengevaluasi akurasi algoritma deteksi objek seperti segmentasi gambar. IoU menilai tumpang tindih antara objek yang diprediksi dan objek yang sebenarnya dengan menghitung rasio tumpang tindih terhadap luas gabungan antara anotasi yang diprediksi dan sebenarnya (Schlosser dkk., 2024). Berdasarkan persamaan 2.1.

$$IoU = \frac{|GT \cap P|}{|GT \cup P|} \tag{2. 21}$$

Keterangan:

IoU : Nilai *Intersection over Union.*

GT : *Ground Truth*, area objek sebenarnya berdasarkan anotasi.

P : *Prediction*, area objek hasil prediksi model.

∩ (*Intersection*) : Luas irisan antara *ground truth* dan prediksi.

∪ (*Union*) : Luas area gabungan antara *ground truth* dan prediksi

2.1.10.5 *Average Precision* (AP/mAP)

Average Precision (AP) mengukur kinerja model untuk setiap kategori secara individual berdasarkan hubungan antara *precision* dan *recall* berdasarkan persamaan 2.2. Selanjutnya, nilai AP dari seluruh kategori dirata-ratakan menjadi *mean Average Precision* (mAP), sehingga evaluasi kinerja model tidak didominasi oleh kategori dengan jumlah *instance* yang lebih banyak (Terven dkk., 2025). Berdasarkan persamaan 2.3.

$$AP = \int_{x=0}^{1} Precision\left(Recall^{-1}(x)\right) dx = \frac{1}{n} . \sum_{i=1}^{n} Precision(Recall_i) \quad (2.\ 22)$$

$$mAP = \frac{1}{n} . \sum_{i=1}^{n} AP_i \quad (2.\ 23)$$

Keterangan:

AP	:	Nilai rata-rata presisi.
mAP	:	Nilai rata rata presisi dari seluruh kelas.
dx	:	Elemen differensial yang menunjukan perhitungan luas area di bawah kurva precision-recall.
n	:	Jumlah titik pengukuran atau threshold recall.
x	:	variabel recall pada rentang 0 sampai 1.
Precision	:	Rasio prediksi benar terhadap seluruh prediksi positif.
Recall	:	Rasio prediksi benar terhadap seluruh *ground truth.*

2.1.10.6 *Dice Coefficient*

Dice Coefficient menghitung kesamaan dua himpunan nilai melalui dua kali irisan dibagi dengan jumlah kedua himpunan tersebut (Schlosser dkk., 2024). dalam tugas yang dimana kelas latar depan relatif jarang seperti deteksi lesi atau objek kecil lainnya, *Dice* dapat mengungguli loss klasifikasi piksel demi piksel secara langsung dengan mengoptimalkan tumpang tindih spasial antara wilayah yang diprediksi dan wilayah sebenarnya (Terven dkk., 2025). Berdasarkan persamaan 2.4.

$$DC = \frac{2 . |GT \cap P|}{|GT| + |P|} \quad (2.\ 24)$$

Keterangan:

DC	:	*Dice Coefficient*, Nilai kesamaan antara hasil prediksi dan *ground truth*.
GT	:	*Ground Truth*, area objek sebenarnya berdasarkan anotasi.
P	:	*Prediction*, area objek hasil prediksi model.
∩ (*Intersection*)	:	Luas irisan antara *ground truth* dan prediksi.

2.1.10.7 *Confusion Matrix*

Confusion matrix merupakan representasi tabular yang digunakan untuk membandingkan kelas aktual dengan kelas hasil prediksi model, sehingga dapat memberikan gambaran mengenai pola prediksi benar maupun kesalahan klasifikasi antar kelas. Pada matriks ini, setiap baris umumnya merepresentasikan kelas sebenarnya, sedangkan setiap kolom merepresentasikan kelas hasil prediksi model. Melalui *confusion matrix*, dapat diketahui kelas-kelas yang paling sering teridentifikasi dengan benar maupun kelas yang cenderung tertukar dengan kelas lain, sehingga analisis kesalahan model dapat dilakukan secara lebih terperinci (Schlosser dkk., 2024; Terven dkk., 2025). Dalam penelitian ini, *confusion matrix* digunakan sebagai analisis tambahan untuk melihat kecenderungan kesalahan prediksi antar kelas penyakit daun teh, khususnya pada kelas-kelas yang memiliki kemiripan karakteristik visual.

2.2 *State of the Art*

Tabel 2. 2 *State of The Art*

No	***State of The Art***	
1	Peneliti, Tahun	(Dai dkk., 2024)
	Judul	AISOA-Ssformer: An Effective Image Segmentation Method for Rice Leaf Disease Based on the *Transformer* Architecture
	Objek	Penyakit daun padi
	Model	AISOA-SSformer (pengembangan dari SegFormer berbasis *Transformer*)
	Hasil	Model AISOA-SSformer memperoleh performa terbaik pada dataset yang dibangun sendiri dengan nilai *MIoU* 83,1%, *Dice coefficient* 80,3%, dan *Recall* 76,5%, dengan ukuran model relatif ringan (14,71 juta parameter). Model masih menunjukkan penurunan performa pada kategori penyakit yang tidak termasuk dalam data pelatihan. Selain itu, performa dapat menurun pada kondisi ekstrem seperti pencahayaan drastis, atau daun saling tumpang tindih. Penulis merekomendasikan perluasan dataset agar mencakup lebih banyak variasi penyakit dan kondisi lingkungan.
2	Peneliti, Tahun	(Alfred dkk., 2025)
	Judul	Detectron2-enhanced *mask* R-CNN for precise instance segmentation of rice blast disease in Tanzania: Supporting timely intervention and data-driven severity assessment
	Objek	Penyakit daun padi
	Model	*Mask* R-CNN berbasis Detectron2 dengan backbone ResNet50 + Feature Pyramid Network (FPN)
	Hasil	Model mencapai performa tinggi dengan mAP 89,4%, AP@50 94,6%, dan AP75 90,5% pada dataset lapangan yang dikumpulkan di lima wilayah Tanzania. Model menunjukkan performa kuat pada objek kecil (APs 81,3%) dan objek besar (APl 86,1%), serta mampu melakukan segmentasi pixel-level untuk menghitung rasio keparahan penyakit (severity ratio) berdasarkan proporsi luas lesi terhadap luas daun terinfeksi. Performa pada objek berukuran sedang relatif lebih rendah (APm 50,8%), diduga karena distribusi data yang tidak seimbang pada kategori ukuran medium. Selain itu, meskipun ukuran model telah dikompresi, inference awal masih relatif lambat pada CPU-only setup (1,53 detik per frame sebelum optimasi). Model juga bergantung pada anotasi polygon pixel-level yang memerlukan proses

No	*State of The Art*	
		labeling cukup intensif dan berpotensi mahal dalam skala besar. Penulis merekomendasikan peningkatan jumlah data untuk kategori lesion berukuran sedang guna meningkatkan APm.
3	Penulis, Tahun	(Shi dkk., 2026)
	Judul	DTM-Unet: A deep learning framework incorporating DenseNet and *Transformer* for precise identification and quantitative segmentation of cucumber leaf disease
	Objek	Penyakit daun mentimun
	Model	DTM-Unet (Dense + *Transformer* + Multi-Residual Dilated Convolution (MRDC) + U-Net)
	Hasil	Model DTM-Unet mencapai performa terbaik pada dataset DM-B dengan nilai Pixel Accuracy (PA) 98,71%, mIoU 86,96%, Recall 91,27%, dan F1-score 87,26%. Model memiliki kompleksitas tinggi dengan jumlah parameter 89,02 juta, lebih besar dibandingkan baseline (misalnya U-Net 47,26M). Hal ini menyebabkan waktu inferensi relatif lebih lambat (±1,198 detik per gambar pada RTX 3090). Selain itu, performa model masih bergantung pada kualitas dan keberagaman dataset pelatihan. Penulis merekomendasikan perluasan dataset dengan variasi penyakit dan kondisi lingkungan yang lebih luas.
4	Penulis, Tahun	(Arora & Gautam, 2026)
	Judul	Plant leaves disease detection in complex background using automatic segmentation and improved multi-scale feature fusion network
	Objek	Penyakit daun apel
	Model	Multi-Scale Feature Fusion (MSFF)
	Hasil	Model MSFF mencapai akurasi 99,85%, precision 99,12%, dan F1-score 97,5%. Pada tahap segmentasi menggunakan L*a*b* color space, nilai IoU tertinggi mencapai 0,99 pada kategori penyakit kombinasi (Rust + Scab). Pendekatan segmentasi berbasis threshold L*a*b* masih sensitif terhadap variasi pencahayaan ekstrem dan karakteristik perangkat akuisisi gambar, meskipun telah dilakukan normalisasi warna. Model fusion memiliki kompleksitas arsitektur tinggi karena menggabungkan empat CNN + ViT, sehingga kurang ringan untuk deployment pada perangkat edge. Penulis merekomendasikan integrasi fitur tambahan seperti tekstur dan bentuk untuk meningkatkan deteksi penyakit multi-class dan pengujian pada domain lain.
5	Penulis, Tahun	(Soeb dkk., 2023)

No	*State of The Art*	
	Judul	Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)
	Objek	Penyakit daun teh dari 5 jenis penyakit dengan total 4000 gambar
	Model	YOLO-T (Modifikasi YOLOv7)
	Hasil	Model YOLO-T memperoleh performa tinggi dengan Detection Accuracy: 97,3%, Precision: 96,7%, Recall: 96,4%, mAP@0.5: 98,2%, F1-score: 0,965. Model mampu bekerja pada natural scene dengan background kompleks, serta mempertahankan performa tinggi dalam kondisi lapangan nyata. Kelemahan utama adalah waktu pelatihan yang cukup lama, yaitu sekitar 5 jam 23 menit, jauh lebih lama dibanding YOLOv5 (±1 jam). Selain itu, Model masih menggunakan bounding box sehingga tidak memberikan segmentasi detail area penyakit. Penulis merekomendasikan Perluasan dataset dengan variasi varietas daun, sudut pengambilan gambar, dan kondisi lingkungan berbeda.
6	Penulis, Tahun	(Xue dkk., 2023a)
	Judul	YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5
	Objek	Penyakit daun teh dari 2 kelas penyakit
	Model	YOLO-Tea (Modifikasi YOLOv5s
	Hasil	Model YOLO-Tea mencapai AP 0.5: 79,3%, APTLB: 73,7% (Tea leaf blight) dan APGMB: 82,6% (Green mirid bug). Model mengalami peningkatan performa dari YOLOv5s baseline sebesar 7,6%. Kekurangan dari penelitian ini adalah dataset yang relatif kecil dan masih berbasis bounding box sehingga tidak memberikan segmentasi detail area penyakit. Penulis merekomendasikan ekspansi dataset penyakit daun teh.
7	Penulis, Tahun	(Y. Li dkk., 2024)
	Judul	Tea leaf disease and insect identification based on improved MobileNetV3
	Objek	Klasifikasi 17 penyakit dan serangga daun teh
	Model	MobileNetV3-CA
	Hasil	Model MobileNetV3-C3 mencapai performa terbaik dengan Accuracy: 95,88%, Precision: 95,98%, Recall: 96,19%, dan F1-score: 96,01%. Penelitian ini memiliki kekurangan yaitu hanya berbasis klasifikasi tanpa deteksi lokasi penyakit.
8	Penulis, Tahun	(Balasundaram dkk., 2025)

No	*State of The Art*	
	Judul	Tea leaf disease detection using Segment Anything Model and deep convolutional neural networks
	Objek	Penyakit daun teh yang berjumlah 6 kelas
	Model	SAM + CNN
	Hasil	Model terbaik (CNN + MLP) mencapai Accuracy: 95,06%, precision, recall dan F1-score memperoleh rata-rata 0,95. Kekurangan dari penelitian ini adalah model SAM sangat sensitif terhadap perubahan pencahayaan dan background, model masih berbasis klasifikasi dan metrik evaluasi yang digunakan masih terbatas. Penulis merekomendasikan hyperparameter tuning pada SAM untuk meningkatkan kualitas *mask* dan penggunaan dataset yang lebih variatif.
9	Penulis, Tahun	(Leng dkk., 2024)
	Judul	An Improved YOLOv8-Based Method for Real-Time Detection of Harmful Tea Leaves in Complex Backgrounds
	Objek	Tiga kategori daun teh berbahaya
	Model	YOLO-DBD (Modifikasi YOLOv8s)
	Hasil	Model YOLO-DBD mencapai performa mAP0.5: 86,8%, mAP0.5-0.95: 78,1%. Model masih memiliki kekurangan dengan masih berfokus pada bounding box dan dataset yang relatif kecil. Rekomendasi dari penulis adalah eksplorasi dataset yang lebih besar.
10	Penulis, Tahun	(Song dkk., 2025)
	Judul	TLDDM: An Enhanced Tea Leaf Pest and Disease Detection Model Based on YOLOv8
	Objek	6 kelas penyakit & hama daun teh
	Model	TLDDM (Modifikasi YOLOv8)
	Hasil	TLDDM mencapai performa mAP0.5: 98%, precision: 98,34%, recall: 96,57%, F1-score: 0,98. Penelitian ini masih memiliki kekurangan yang masih berbasis bounding box dan performa hanya diuji pada dataset teh. Penulis merekomendasikan untuk menguji pada domain lain dan mengintegrasikan dengan model YOLO terbaru.
11	Penulis, Tahun	(Paul dkk., 2026)
	Judul	YOLOv9t-DyE: A lightweight detection framework with SAM-assisted segmentation for quantifying chilli leaf curl complex

No	*State of The Art*	
	Objek	Penyakit daun cabai
	Model	YOLOv9t-DyE (modifikasi dari YOLOv9t) + MobileSAM
	Hasil	Model YOLOv9t-DyE mencapai *precision* sebesar 0.769, mAP sebesar 0.813, dan mAP0.5-0.95 sebesar 0.45. Pipeline sangat bergantung pada proses deteksi YOLO, jika *bounding-box* tidak akurat, maka segmentasi dan estimasi severity ikut terdampak. Penulis merekomendasikan untuk ekspansi dataset di berbagai varietas dan kondisi lingkungan dan eksplorasi *zero-shot detection*.

Berdasarkan Tabel 2.1, penelitian terdahulu menunjukkan bahwa pendekatan segmentasi berbasis deep learning, baik yang mengintegrasikan CNN maupun *Transformer*, telah mencapai performa yang cukup tinggi pada berbagai jenis penyakit daun tanaman. Model seperti AISOA-SSFormer dan DTM-Unet mampu memperoleh nilai mIoU di atas 83%, sementara pendekatan *Mask* R-CNN dan beberapa varian YOLO menunjukkan capaian mAP yang tinggi pada tugas deteksi. Pada kasus daun teh, sebagian besar penelitian masih berfokus pada deteksi berbasis bounding box, seperti YOLO-T, YOLO-Tea, YOLO-DBD, dan TLDDM, dengan performa mAP yang kompetitif. Namun, pendekatan tersebut belum mampu memberikan representasi detail area penyakit pada tingkat piksel karena masih terbatas pada kotak pembatas.

Beberapa penelitian terbaru mulai memanfaatkan *Segment Anything Model* (SAM) untuk membantu proses segmentasi penyakit daun, baik sebagai alat segmentasi awal maupun dikombinasikan dengan model deteksi berbasis YOLO untuk menghitung tingkat keparahan penyakit. Namun, penggunaannya masih bersifat terpisah dari model utama dan belum diterapkan sebagai segmentasi *end-*

to-end dalam satu arsitektur terpadu berbasis *Transformer*. Meskipun banyak penelitian telah mencapai akurasi deteksi dan klasifikasi yang tinggi, masih terdapat keterbatasan dalam menghasilkan detail segmentasi area penyakit pada level piksel, ketergantungan pada proses anotasi manual yang cukup intensif, serta tantangan dalam menghadapi variasi kondisi pencahayaan dan latar belakang di lapangan. Oleh karena itu, diperlukan pendekatan yang mampu menggabungkan segmentasi berbasis *Transformer* secara *end-to-end* dengan strategi anotasi yang lebih presisi dan efisien, sehingga representasi area penyakit daun teh dapat diperoleh secara lebih detail, akurat, dan adaptif terhadap kondisi nyata di lapangan.

2.3 Matriks Penelitian

Tabel 2. 3 Matriks Penelitian

No	Penulis	Model				Anotasi		Jenis Tugas			Pixel-level		SAM
		CNN	YOLO	*Transformer*	*Hybrid*	Manual	*Assisted*	Klasifikasi	Objek	Segmentasi	Ya	Tidak	
1	(Dai dkk., 2024)	-	-	√	-	√	-	-	-	√	√	-	-
2	(Alfred dkk., 2025)	√	-	-	-	√	-	-	-	√	√	-	-

No	Penulis	Model				Anotasi		Jenis Tugas			Pixel-level		SAM
		CNN	YOLO	*Transformer*	*Hybrid*	Manual	*Assisted*	Klasifikasi	Objek	Segmentasi	Ya	Tidak	
3	(Shi dkk., 2026)	-	-	-	√	√	-	-	-	√	√	-	-
4	(Arora & Gautam, 2026)	-	-	-	√	-	√	√	-	-	√	-	-
5	(Soeb dkk., 2023)	-	√	-	-	√	-	-	√	-	-	√	-
6	(Xue dkk., 2023b)	-	√	-	-	√	-	-	√	-	-	√	-
7	(Y. Li dkk., 2024)	√	-	-	-	√	-	√	-	-	-	√	-
8	(Balasundaram dkk., 2025)	√	-	-	-	-	√	-	-	√	√	-	√
9	(Leng dkk., 2024)	-	√	-	-	√	-	-	√	-	-	√	-
10	(Song dkk., 2025)	-	√	-	-	√	-	-	√	-	-	√	-
11	(Paul dkk., 2026)	-	√	-	-	√	-	-	-	√	√	-	√
12	**Penelitian ini**	-	-	√	-	-	√	-	-	√	√	-	√