

**PEMETAAN DAN IDENTIFIKASI KESIAPAN PETIK TANAMAN TEH  
BERDASARKAN CITRA *DRONE* MENGGUNAKAN *MASK REGION-BASED  
CONVOLUTIONAL NEURAL NETWORK (MASK R-CNN)* DAN *GREEN LEAF INDEX  
(GLI)***

**SKRIPSI**

Diajukan untuk Memenuhi Sebagian dari  
Syarat Memperoleh Gelar Sarjana Komputer  
Program Studi Ilmu Komputer



Oleh:

Muhammad Ramadhan Maulana. A

1803771

**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PENDIDIKAN INDONESIA  
2023**

## **LEMBAR HAK CIPTA**

### **PEMETAAN DAN IDENTIFIKASI KESIAPAN PETIK TANAMAN TEH BERDASARKAN CITRA DRONE MENGGUNAKAN *MASK REGION-BASED CONVOLUTIONAL NEURAL NETWORK (MASK R-CNN)* DAN *GREEN LEAF INDEX (GLI)***

Oleh  
Muhammad Ramadhan Maulana. A  
NIM 1803771

Sebuah skripsi yang diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer pada Program Studi Ilmu Komputer di Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam

© Muhammad Ramadhan Maulana. A  
Universitas Pendidikan Indonesia  
Desember 2023

Hak cipta dilindungi Undang-Undang  
Skripsi ini tidak boleh diperbanyak seluruhnya atau sebagian, dengan dicetak ulang, difotokopi, atau cara lainnya tanpa izin dari penulis.

## **LEMBAR PENGESAHAN**

### **PEMETAAN DAN IDENTIFIKASI KESIAPAN PETIK TANAMAN TEH BERDASARKAN CITRA *DRONE* MENGGUNAKAN *MASK REGION-BASED CONVOLUTIONAL NEURAL NETWORK (MASK R-CNN)* DAN *GREEN LEAF INDEX (GLI)***

Oleh

Muhammad Ramadhan Maulana. A  
1803771

Disetujui dan disahkan oleh:

Pembimbing I,

**Eddy Prasetyo Nugroho, M.T.**  
NIP. 197505152008011014

Pembimbing II,

**Rosa Ariani Sukamto, M.T.**  
NIP. 198109182009122003

Mengetahui

Ketua Program Studi Pendidikan Ilmu Komputer

**Dr. Muhammad Nursalman, M.T.**  
NIP. 197909292006041002

## **PERNYATAAN**

Saya menyatakan bahwa skripsi yang berjudul “Pemetaan dan Identifikasi Kesiapan Petik Tanaman Teh Berdasarkan Citra *Drone* Menggunakan *Mask Region-Based Convolutional Neural Network (Mask R-CNN)* dan *Green Leaf Index (GLI)*” ini sepenuhnya hasil karya saya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, Desember 2023

Muhammad Ramadhan Maulana. A

1803771

## KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan kepada Allah SWT. Atas limpahan rahmat, ridha, dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul “Pemetaan dan Identifikasi Kesiapan Petik Tanaman Teh Berdasarkan Citra *Drone* Menggunakan *Mask Region-Based Convolutional Neural Network (Mask R-CNN)* dan *Green Leaf Index (GLI)*”.

Tujuan dari penyusunan skripsi ini untuk memenuhi dan melengkapi salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Ilmu Komputer Universitas Pendidikan Indonesia.

Penulisan menyadari masih terdapat kekurangan dalam penulisan skripsi. Serta jauh dari kesempurnaan baik dalam bahasan, penyusunan, maupun penulisannya dikarenakan oleh keterbatasan dimiliki oleh penulis. Maka dari itu, penulis mengharapkan kritik serta saran sehingga dapat menjadi masukan bagi penulis untuk lebih baik kedepannya.

Semoga skripsi ini dapat memberikan manfaat khususnya bagi penulis dan umumnya bagi semua pihak yang membacanya.

Bandung, Desember 2023

Penulis

## **UCAPAN TERIMA KASIH**

*Alhamdulillahi Rabbil 'Aalamiin*, segala puji bagi Allah SWT. Atas limpahan rahmat, ridha, dan karunia-Nya sehingga atas izin-Nya penulis dapat menyelesaikan skripsi yang berjudul “Pemetaan dan Identifikasi Kesiapan Petik Tanaman Teh Berdasarkan Citra Drone Menggunakan *Mask Region-Based Convolutional Neural Network (Mask R-CNN)* dan *Green Leaf Index (GLI)*”. Dalam penyusunan skripsi ini, penulis telah mendapatkan banyak bimbingan, dorongan, bantuan, serta doa dari berbagai pihak baik secara langsung ataupun tidak langsung. Pada kesempatan ini, penulis sekaligus peneliti mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Kedua Orang tua, saudara dan keluarga saya yang tanpa henti-hentinya selalu memberikan doa serta dukungan untuk kelancaran penelitian sehingga dapat memberikan motivasi peneliti dalam menyelesaikan skripsi.
2. Bapak Nursalman selaku Ketua Program Studi Ilmu Komputer Departemen Pendidikan Ilmu Komputer Universitas Pendidikan Indonesia yang telah membantu dan mendukung saya dalam penyelesaian skripsi ini.
3. Ibu Dr. Rani Megasari, M.Kom. selaku dosen yang sebelumnya menjabat sebagai Ketua Program Studi Ilmu Komputer Departemen Pendidikan Ilmu Komputer Universitas Pendidikan Indonesia yang senantiasa memberikan arahan serta dukungan kepada peneliti selama masa kuliah dan skripsi.
4. Bapak Eddy Prasetyo Nugroho, M.T. selaku dosen pembimbing I yang telah meluangkan waktu, ilmu, pengalaman serta memberikan bimbingan kepada penulis, sehingga skripsi ini dapat terselesaikan.
5. Ibu Rosa Ariani Sukamto, M.T. selaku dosen pembimbing II yang telah meluangkan waktu, memberikan saran serta arahan kepada penulis, sehingga skripsi ini dapat terselesaikan.
6. Bapak Yudi Wibisono selaku Dosen Pembimbing Akademik yang telah membimbing dan memberikan arahan kepada penulis selama masa kuliah.
7. Tammy, Zidan, Indra, Athiyyah, Safira, Indra, Riza dan teman-teman SMP penulis lainnya yang selalu memberikan masukan serta dukungan selama penyelesaian skripsi ini. Bahkan ada yang sampai menemanai penulis hingga larut malam ketika melakukan penyelesaian skeipsi ini, penulis ucapkan terima kasih banyak terhadap segala bentuk perhatian yang telah kalian berikan kepada penulis.

8. Sendy, Arafi, Ronald, Rivaldi, Dendi, Farhan dan teman teman SMK penulis lainnya yang selalu memberikan dukungan dan juga menjadi tempat untuk meluangkan keluh kesah selama perkuliahan sampai dengan proses penyelesaian skripsi.
9. Audrey, Fina, Rully, Andin, Eva, Afe, Alyya, Syachrul, Phonteuka dan teman teman satu jurusan penulis lainnya yang selama penyelesaian skripsi ini selalu membantu meningkatkan semangat penulis dalam penyelesaian skripsi dengan segala kegiatan bersama yang menyenangkan.
10. Nemo, Dori, Rona, Dimsum, Ano, Cevin selaku teman penulis yang penulis kenal secara daring, penulis sangat berterima kasih pada kalian yang telah senantiasa menemani malam-malam penulis dengan keseruan dan selalu memberikan dukungan serta masukan positif kepada penulis selama menjalani studi di masa pandemi hingga sekarang dimana terselesaiannya skripsi ini.
11. Terkhusus untuk seseorang yang tidak dapat penulis sebutkan namanya, sosok yang sangat berjasa dalam memberikan dukungan dan semangat yang luar biasa. Berkat bantuan dan dorongannya, penulis berhasil membangun kembali semangat dalam menyelesaikan skripsi ini setelah tertunda cukup lama. Bantuan dan dukungannya yang tanpa henti saat itu sungguh berarti, dan sekarang, akhirnya, skripsi ini berhasil terselesaikan. Meskipun saat ini penulis terpaksa berada di jalan yang berbeda, namun penulis benar-benar ingin mengucapkan terima kasih yang tulus atas segala bentuk kontribusi, dukungan, hingga attensi yang telah diberikan. Hal yang tidak akan pernah bisa penulis lupakan dan akan selalu menjadi pelajaran terbaik bagi penulis.
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam menyelesaikan skripsi ini.

**PEMETAAN DAN IDENTIFIKASI KESIAPAN PETIK TANAMAN TEH  
BERDASARKAN CITRA DRONE MENGGUNAKAN *MASK REGION-BASED*  
CONVOLUTIONAL NEURAL NETWORK (*MASK R-CNN*) DAN GREEN LEAF INDEX  
(GLI)**

**ABSTRAK**

Sebagai negara agraris, sektor pertanian memainkan peran krusial dalam perekonomian Indonesia. Teh, sebagai salah satu komoditi utama ekspor setelah minyak dan gas, menjadi penopang devisa negara. Manajemen perkebunan teh menghadapi tantangan *monitoring* yang signifikan, terutama dengan karakteristik lahan yang luas dan kompleks. Proses *monitoring* konvensional dengan melibatkan banyak tenaga pekerja dan waktu yang lama menjadi hambatan utama. Penelitian ini menjelaskan implementasi teknologi *drone* dalam pemetaan dan identifikasi kesiapan petik tanaman teh. Diawali dengan akuisisi data citra hasil tangkapan *drone* lalu diolah untuk dijadikan data untuk penggabungan citra dan *dataset* untuk model pemetaan dan klasifikasi kesiapan petik teh. Selanjutnya implementasi metode *image stitching* menghasilkan data citra area kebun teh yang luas dari citra tangkapan *drone*. Kemudian dilakukan pemetaan terhadap area tanaman teh berdasarkan citra hasil gabungan menggunakan metode *Mask Region-Based Convolutional Neural Network (Mask R-CNN)*. Hingga pada akhirnya klasifikasi dilakukan terhadap hasil pemetaan untuk mengidentifikasi tingkat kesiapan petik area tanaman teh tersebut berdasarkan nilai *Green Leaf Index (GLI)* dan juga *Artificial Neural Network*, model *sequential*. Model *Mask R-CNN* berhasil melakukan pemetaan area tanaman teh dengan rata-rata tingkat presisi sebesar 39%, serta identifikasi tingkat kehijauan teh menggunakan (GLI) serta model *sequential* mampu mengklasifikasikan kesiapan petik tanaman teh, dengan akurasi mencapai 82% dan nilai *loss* sebesar 0.4202. Hasil penelitian ini memberikan kontribusi dalam meningkatkan efisiensi *monitoring* perkebunan teh, memberikan dasar informasi yang lebih baik untuk pengambilan keputusan, dan secara potensial meningkatkan daya saing industri teh Indonesia.

Kata kunci: *citra drone, green leaf index, image stitching, mask region-based convolutional neural network*, pemantauan pemetaan teh, pemetaan area tanaman teh

**PEMETAAN DAN IDENTIFIKASI KESIAPAN PETIK TANAMAN TEH  
BERDASARKAN CITRA DRONE MENGGUNAKAN MASK REGION-BASED  
CONVOLUTIONAL NEURAL NETWORK (MASK R-CNN) DAN GREEN LEAF INDEX  
(GLI)**

**ABSTRACT**

*As an agrarian nation, the agricultural sector plays a crucial role in Indonesia's economy. Tea, as one of the primary export commodities following oil and gas, serves as a key contributor to the country's foreign exchange reserves. Tea plantation management faces significant monitoring challenges, particularly due to the vast and complex nature of the land. Conventional monitoring processes, involving extensive manual labor and time, pose a major obstacle. This research elucidates the implementation of drone technology for mapping and identifying the readiness for tea harvest. The process begins with acquiring image data captured by drones, which is then processed for image fusion and dataset creation for mapping and classifying tea harvest readiness models. Subsequently, the implementation of the image stitching method generates extensive image data of tea plantation areas from drone captures. The mapping of tea plant areas is then performed based on the fused images using the Mask Region-Based Convolutional Neural Network (Mask R-CNN) method. Ultimately, classification is conducted on the mapping results to identify the harvest readiness level of tea plant areas based on the Green Leaf Index (GLI) and an Artificial Neural Network, sequential model. The Mask R-CNN model successfully maps tea plant areas with an average precision of 39%, while the GLI and sequential model classify tea plant harvest readiness with an accuracy of 82% and a loss value of 0.4202. This research contributes to enhancing the efficiency of tea plantation monitoring, providing a better information foundation for decision-making, and potentially boosting the competitiveness of Indonesia's tea industry.*

*Keywords:* *drone imagery, green leaf index, image stitching, mask region-based convolutional neural network, tea picking monitoring, tea plant area mapping*

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>iii</b>
<b>PERNYATAAN.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>UCAPAN TERIMA KASIH .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>viii</b>
<b>ABSTRACT.....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	5
1.3. Tujuan Penelitian .....	5
1.4. Batasan Penelitian .....	5
1.5. Manfaat Penelitian .....	6
1.6. Sistematika Penulisan .....	7
<b>BAB II KAJIAN PUSTAKA.....</b>	<b>12</b>
2.1. Teh .....	12
2.2. Pesawat Nirawak/ <i>Unmanned Air Vehicle (UAV)/Drone</i> .....	14
2.3. <i>Computer Vision</i> .....	15
2.4. Kecerdasan Buatan/ <i>Artificial Intelligence</i> .....	17
2.4.1 <i>Machine Learning</i> .....	18
2.4.1 <i>Deep Learning</i> .....	18
2.4.3 <i>Artificial Neural Networks (ANN)</i> .....	19
2.4.3 <i>Convolutional Neural Networks (CNN)</i> .....	21
2.4.4 <i>Mask R-Convolutional Neural Networks (CNN)</i> .....	23
2.5. <i>Image Stitching</i> .....	32
2.5.2 <i>Scale-Invariant Feature Transform (SIFT)</i> .....	35
2.5.1 Oriented FAST and Rotated BRIEF (ORB) .....	35

2.5.3 Binary Robust Invariant Scalable Keypoints (BRISK) .....	36
2.5.4 <i>Accelerated-KAZE</i> (AKAZE) .....	37
2.6. Image Segmentation.....	38
2.7. <i>Vegetation Index</i> .....	39
2.7.1 <i>Visible Atmospheric Resistant Index</i> (VARI) .....	42
2.7.2 <i>Green Leaf Index</i> (GLI) .....	43
2.8. OpenCV .....	43
2.9. Detectron2.....	44
2.9. Penelitian Terdahulu .....	45
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>61</b>
3.1 Desain Penelitian .....	61
3.2 Alat dan Bahan Penelitian.....	65
3.2.1 Alat Penelitian.....	65
3.2.2 Bahan Penelitian .....	66
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN .....</b>	<b>67</b>
4.1 Perancangan Model.....	67
4.1.1 Perancangan Model Penggabungan Citra .....	69
4.1.2 Perancangan Model Pemetaan Tanaman Teh .....	70
4.1.3 Perancangan Model Identifikasi Kesiapan Petik Tanaman Teh .....	70
4.2 Identifikasi Kebutuhan Data .....	71
4.2.1 Data Citra Area Perkebunan Teh .....	72
4.2.2 <i>Dataset Objek Tanaman Teh</i> .....	73
4.2.3 <i>Dataset Vegetation Index</i> Tanaman Teh.....	75
4.3 Pengembangan Model.....	78
4.3.1 Pengembangan Model Penggabungan Citra .....	78
4.3.2 Pengembangan Model Pemetaan Tanaman Teh .....	91
4.3.3 Pengembangan Model Identifikasi Kesiapan Petik Tanaman Teh .....	96
4.4 Implementasi dan Integrasi Setiap Model.....	100
4.5 Analisis dan Evaluasi Hasil .....	110
4.5.1 Evaluasi Hasil Penggabungan Citra.....	110
4.5.2 Evaluasi Hasil Pemetaan Area Tanaman Teh.....	112

4.5.3 Evaluasi Hasil Identifikasi Kesiapan Petik Tanaman Teh.....	115
<b>BAB V PENUTUP .....</b>	<b>117</b>
5.1. Kesimpulan.....	117
5.2. Saran .....	118
<b>DAFTAR PUSTAKA.....</b>	<b>120</b>

## DAFTAR GAMBAR

Gambar 2.1 Perkebunan teh (Maulida dkk., 2022).....	12
Gambar 2.2 <i>Unmanned air vehicle (UAV)/drone</i> (DJI, 2023).....	14
Gambar 2.3 Perbandingan antara hasil tangkapan <i>drone</i> dan satelit (Satish, 2022).....	15
Gambar 2.4 Pohon topik penelitian terkait citra dan <i>vision</i> (Shahab dkk., 2007) .....	16
Gambar 2.5 Hubungan antara AI, ML, NN, dan DL (S. Li dkk., 2021).....	18
Gambar 2.6 Arsitektur dari <i>Artificial Neural Network</i> dengan <i>Sequential Model</i> . .....	20
Gambar 2.7 Gambaran arsitektur <i>convolutional neural networks</i> (CNN) (Minaee dkk., 2020). .	22
Gambar 2.8 Alur perkembangan R-CNN hingga <i>Mask R-CNN</i> .....	24
Gambar 2.9 Gambaran umum deteksi objek menggunakan R-CNN (Girshick dkk., 2013). ....	25
Gambar 2.10 Gambaran arsitektur <i>Fast R-CNN</i> (Ren dkk., 2015). .....	25
Gambar 2.11 Perbedaan antara arsitektur R-CNN dan Fast R-CNN (P. P. Sharma, 2021). ....	26
Gambar 2.12 Arsitektur <i>Faster R-CNN</i> (Ren dkk., 2015). .....	27
Gambar 2.13 Arsitektur <i>Mask R-CNN</i> (He dkk., 2017). .....	28
Gambar 2.14 Detail arsitektur <i>Mask R-CNN</i> untuk <i>image segmentation</i> (J. Lin dkk., 2022)....	28
Gambar 2.15 <i>Feature Pyramid Networks (FPN) backbone</i> (Subash dkk., 2020).....	29
Gambar 2.16 <i>Region Proposal Network (RPN)</i> (Subash dkk., 2020). .....	30
Gambar 2.17 Proses RoIAlign (Jiao dkk., 2019).....	30
Gambar 2.18 Struktur <i>Mask Head</i> (He dkk., 2017).....	32
Gambar 2.19 Tahapan proses <i>image stitching</i> (S. K. Sharma & Jain, 2020).....	33
Gambar 2.20 Contoh hasil <i>semantic</i> dan <i>instance segmentation</i> (Minaee dkk., 2020).....	39
Gambar 2.21 Tingkat RGB dan NIR terhadap kesehatan daun (Evangelides & Nobajas, 2020). 40	
Gambar 2.22 Perbandingan jumlah piksel yang dianggap hijau antara hasil GLI dan VARI.....	60
Gambar 3.1 Desain penelitian.....	61
Gambar 4.1 Arsitektur sistem pemetaan dan identifikasi kesiapan petik tanaman teh.....	67
Gambar 4.2 Diagram alir tahapan penelitian dari awal hingga akhir. .....	68
Gambar 4.3 Tahapan Proses <i>Image Stitching</i> . .....	69
Gambar 4.4 Arsitektur dari <i>Artificial Neural Network</i> dengan <i>Sequential Model</i> . .....	71
Gambar 4.5 Pratinjau hasil tangkapan <i>drone</i> terhadap area perkebunan teh.....	72
Gambar 4.6 Pratinjau hasil anotasi/ <i>labeling</i> objek tanaman teh pada citra perkebunan teh.....	73
Gambar 4.7 Struktur JSON untuk menyimpan label/anotasi dan <i>metadata</i> untuk <i>dataset</i> . .....	74

Gambar 4.8 Diagram alir perhitungan <i>vegetation index</i> dengan <i>green leaf index</i> (GLI) .....	75
Gambar 4.9 Gambaran hasil dari perhitungan <i>vegetation index</i> dengan GLI.....	76
Gambar 4.10 Diagram alir ekstraksi VI data area tanaman teh berdasarkan <i>dataset</i> hasil anotasi. .....	77
Gambar 4.11 Visualisasi ekstraksi VI pada sebuah area tanaman teh berdasarkan data anotasi..	77
Gambar 4.12 Pratinjau <i>dataset</i> VI per objek tanaman teh. ....	78
Gambar 4.13 Pratinjau kelompok citra Siap Petik 4.....	80
Gambar 4.14 Hasil resolusi citra terhadap proses <i>resize image</i> .....	80
Gambar 4.15 Hasil pendekripsi fitur pada citra menggunakan metode SIFT. ....	81
Gambar 4.16 <i>Output</i> dari pencocokan fitur dan perhitungan <i>confidence</i> . ....	82
Gambar 4.17 Visualisasi pencocokan fitur terhadap dua citra. ....	82
Gambar 4.18 Diagram <i>visualiasi</i> hubungan dari citra berdasarkan <i>confidence</i> .....	83
Gambar 4.19 Data <i>subset</i> berisikan <i>filename</i> citra yang relevan untuk penggabungan citra. ....	83
Gambar 4.20 Hasil dari transformasi setiap citra yang akan digabung pada proses <i>warping</i> . ....	84
Gambar 4.21 Hasil dari posisi ( <i>corners</i> ) dan ukuran ( <i>size</i> ) akhir dari setiap citra. ....	85
Gambar 4.22 Hasil dari <i>seam mask</i> .....	85
Gambar 4.23 Citra hasil proses <i>blending</i> .....	86
Gambar 4.24 Visualisasi garis transisi setiap citra pada citra akhir hasil penggabungan.....	86
Gambar 4.25 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 1’.....	87
Gambar 4.26 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’.....	87
Gambar 4.27 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’.....	88
Gambar 4.28 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 4’ .....	88
Gambar 4.29 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 5’ .....	89
Gambar 4.30 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 6’ .....	89
Gambar 4.31 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 1’.....	90
Gambar 4.32 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 2’.....	90
Gambar 4.33 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 3’.....	91
Gambar 4.34 Konfirmasi pendaftaran <i>dataset</i> dengan menampilkan citra beserta <i>label</i> .....	93
Gambar 4.35 Konfigurasi model Mask R-CNN pada <i>framework</i> Detectron2.....	94
Gambar 4.36 Kategori dan jumlah <i>instances</i> dari data <i>train</i> . .....	95
Gambar 4.37 Contoh hasil identifikasi area tanaman teh dengan model yang telah dilatih.....	96

Gambar 4.38 Penampilan <i>dataset</i> yang telah didaftarkan pada program. ....	97
Gambar 4.39 Hasil pembagian <i>dataset</i> menjadi data <i>train</i> dan <i>test</i> .....	98
Gambar 4.40 Gambaran data yang digunakan untuk melakukan prediksi. ....	99
Gambar 4.41 Contoh hasil prediksi kesiapan petik tanaman teh. ....	100
Gambar 4.42 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 1’. ....	101
Gambar 4.43 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 1’.....	101
Gambar 4.44 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’.....	102
Gambar 4.45 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 2’.....	102
Gambar 4.46 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’.....	103
Gambar 4.47 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 3’.....	103
Gambar 4.48 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 4’ . ....	104
Gambar 4.49 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 4’.....	104
Gambar 4.50 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 5’ . ....	105
Gambar 4.51 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 5’.....	105
Gambar 4.52 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 6’ . ....	106
Gambar 4.53Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 6’.....	106
Gambar 4.54 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 1’.....	107
Gambar 4.55 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Tidak Siap Petik 1’.	
.....	107
Gambar 4.56 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’ .....	108
Gambar 4.57 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 2’.....	108
Gambar 4.58 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’ . ....	109
Gambar 4.59 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 3’.....	109
Gambar 4.60 Kategori dan jumlah <i>instances</i> dari data <i>evaluation</i> . ....	112
Gambar 4.61 Hasil evaluasi <i>loss</i> dan <i>accuracy</i> dari model yang dilatih. ....	115
Gambar 4.62 Gambaran dari <i>confusion matrix</i> .....	116
Gambar 4.63 <i>Confusion matrix</i> dari hasil prediksi pada data <i>test</i> . ....	116

## **DAFTAR TABEL**

Tabel 2.1 Penelitian Terdahulu.....	45
Tabel 2.2 Daftar metode <i>vegetation index</i> yang sering digunakan pada penelitian.....	58
Tabel 4.1 Pengelompokan data citra untuk proses penggabungan gambar.....	79
Tabel 4.2 <i>Training time</i> dari pelatihan model.....	112
Tabel 4.3 Tingkat kepresisian model hasil pelatihan.....	113

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Sebagai negara agraris, salah satu sektor yang cukup penting dalam perekonomian di Indonesia adalah sektor pertanian (Subdirektorat Statistik Tanaman Perkebunan, 2021). Tentunya hal tersebut menjadikan sebagian besar dari populasi penduduk di Indonesia mempunyai mata pencaharian yang berpusat pada sektor pertanian (Maulida dkk., 2022). Dalam sektor tersebut, komoditi yang berperan penting dalam perekonomian Indonesia salah satunya adalah teh. Selain minyak dan gas, teh adalah barang ekspor yang signifikan untuk menghasilkan mata uang asing/devisa bagi negara. Hal tersebut dibuktikan dengan masuknya Indonesia kedalam peringkat besar negara pengekspor teh di dunia (Subdirektorat Statistik Tanaman Perkebunan, 2021).

Kualitas teh yang terjaga sesuai dengan standar dan juga kuantitas produksi yang stabil merupakan faktor penting pendukung daya saing teh di pasar ekspor (Chaprilia & Yuliawati, 2018; Sita & Rohdiana, 2021). Hal tersebut dapat dicapai salah satunya dengan adanya upaya penanganan intensif untuk komoditi teh mulai dari area kebun di mana teh ditanam dan dipanen hingga ke pabrik (Effendi dkk., 2010). Tentunya dalam manajemen penanganan teh yang baik dibutuhkan proses yang disebut sebagai pemantauan/*monitoring* (THEREZA dkk., 2020) guna mendapatkan informasi terkait kondisi lingkungan perkebunan (Faruqi, 2019). Informasi tersebut dapat menjadi dasar dari setiap keputusan yang diperlukan baik bersifat saran (Faruqi, 2019) maupun solusi untuk penanggulangan masalah yang terjadi (Manfreda dkk., 2018; THEREZA dkk., 2020).

Karakteristik dari lahan perkebunan teh yang relatif sangat luas menyebabkan sukaranya proses *monitoring* untuk dilakukan (Maulida dkk., 2022). Alasan tersebut membuat *monitoring* dengan sistem konvensional membutuhkan tenaga pekerja yang banyak dan juga memakan waktu yang lama (THEREZA dkk., 2020; Zhang dkk., 2021). Ketidakoptimalan proses *monitoring* lahan perkebunan teh dapat menimbulkan manajemen operasional menjadi tidak efisien sehingga baik pengambilan keputusan maupun respon terhadap gangguan di lapangan menjadi relatif lambat (THEREZA dkk., 2020). Pada akhirnya masalah tersebut akan berdampak secara langsung pada produktivitas dan kualitas teh yang dihasilkan (Maulida dkk., 2022; Rahardjo dkk., 2015).

Penerapan teknologi yang tepat dalam proses *monitoring* dapat menjadi aspek yang mempengaruhi solusi dari masalah yang terjadi di lapangan, terutama di lahan perkebunan teh yang luas (Rahardjo dkk., 2015; Wirabudi dkk., 2019). Salah satu teknologi yang dapat dimanfaatkan untuk optimasi proses *monitoring* adalah pesawat nirawak atau yang lebih dikenal dengan sebutan *drone* (Wirabudi dkk., 2019; Zhang dkk., 2021). Dalam beberapa tahun terakhir teknologi *Drone* menjadi alternatif yang populer untuk mengoptimalkan *environmental monitoring* didalam *agricultural system* seperti perkebunan dan bahkan hutan (Barbedo, 2019; Manfreda dkk., 2018; Siregar, 2023; Teixeira dkk., 2023).

*Drone* dapat menangkap citra digital dari udara yang tentunya sangat cocok dengan lahan perkebunan teh yang luas. Dibandingkan dengan teknologi yang serupa seperti satelit, *drone* memiliki banyak keunggulan seperti mampu menangkap gambar yang bebas awan dan gangguan atmosfer, resolusi tangkapan gambar tinggi, biaya operasional yang jauh lebih murah, efisien secara waktu, dan juga mampu menjangkau area yang tidak dapat dijangkau oleh satelit maupun oleh pesawat besar (Hernina & Putra, 2021).

Dengan segala kelebihan yang dapat diberikan oleh teknologi *drone*, tentu saja tidak membuat teknologi *drone* sempurna. Semakin tinggi terbang *drone*, maka semakin luas juga area yang menjadi cakupan dari tangkapan citra udara *drone* (Özgür Çelik dkk., 2020). Hal tersebut memang menguntungkan karena area perkebunan teh yang luas dapat dicakup dalam satu tangkapan citra udara. Akan tetapi itu mengakibatkan semakin rendah detail dari tangkapan citra udara yang dihasilkan (Özgür Çelik dkk., 2020; Seifert dkk., 2019). Tentunya dengan hal tersebut, pengaturan *drone* yang dibuat terbang lebih rendah menjadi solusi agar mendapatkan citra udara dengan detail yang lebih tinggi. Namun dengan pengaturan terbang yang rendah, maka cakupan dari tangkapan citra udara yang dihasilkan semakin kecil yang pada akhirnya menyebabkan *drone* harus lebih banyak menangkap citra udara untuk dapat mencakup keseluruhan area perkebunan teh (Özgür Çelik dkk., 2020).

Pemanfaatan citra udara tangkapan *drone* dapat dimanfaatkan sebagai sumber citra untuk pemetaan seluruh tanaman teh pada area perkebunan. Dengan melakukan analisis menyeluruh terhadap hasil dari proses pemetaan, tentunya dapat menghasilkan banyak informasi mengenai kondisi tanaman teh pada area perkebunan. Sebagai contoh, informasi tersebut bisa berupa identifikasi luasan area untuk pemetikan teh, kadar air maupun nutrisi, bahkan hingga informasi

penyakit dan hama pada teh (Barbedo, 2019; Cuaran & Leon, 2021; Neupane & Baysal-Gurel, 2021; Sishodia dkk., 2020).

Penggabungan citra tentunya dapat dilakukan untuk mengatasi keterbatasan area tangkapan citra dari *drone*, sehingga dengan menggabungkan potongan-potongan citra tersebut dapat menghasilkan gambaran utuh dari area perkebunan teh. Penggabungan citra dapat dilakukan menggunakan aplikasi untuk melakukan rekayasa citra seperti Adobe Photoshop maupun dengan pendekatan metode *image stitching*. Metode *image stitching* adalah teknik untuk menggabungkan (*stitch/menjahit*) beberapa gambar yang menghasilkan satu kesatuan utuh sebuah gambar (Z. Wang & Yang, 2020). Teknik ini dapat digunakan untuk menanggulangi masalah terkait dengan tangkapan citra udara *drone* terhadap area perkebunan teh yang di mana jumlah tangkapan citra udaranya banyak karena berfokus pada detail dan kualitas tangkapan citra udara (Pham dkk., 2021). Dengan begitu dapat diraih citra udara dari area perkebunan teh dengan detail dan resolusi yang tinggi secara menyeluruh.

Pemanfaatan citra udara tangkapan *drone* juga dapat dimanfaatkan sebagai sumber citra untuk pemetaan seluruh tanaman teh pada area perkebunan. Dengan melakukan analisis menyeluruh terhadap hasil dari proses pemetaan, tentunya dapat menghasilkan banyak informasi mengenai kondisi tanaman teh pada area perkebunan. Sebagai contoh, informasi tersebut bisa berupa identifikasi luasan area untuk pemetikan teh, kadar air maupun nutrisi, kesiapan petik dari teh yang biasanya berdasarkan warna daun teh, bahkan hingga informasi penyakit dan hama pada teh (Barbedo, 2019; Cuaran & Leon, 2021; Neupane & Baysal-Gurel, 2021; Sishodia dkk., 2020). Warna hijau pada daun tanaman teh menjadi indikator utama kesehatan dan kematangan tanaman. Perubahan warna daun mencerminkan proses fotosintesis dan akumulasi pigmen, termasuk klorofil. Klorofil adalah komponen utama dalam fotosintesis yang memberikan warna hijau pada daun. Oleh karena itu, memonitor tingkat kehijauan daun dapat memberikan petunjuk mengenai tahapan pertumbuhan dan kematangan tanaman teh.

Menentukan tingkat kehijauan atau mengukur tingkat kesehatan tanaman dapat dilakukan berdasarkan perhitungan *Vegetation Index* atau indeks vegetasi. Indeks vegetasi yang paling umum digunakan adalah *Normalized Difference Vegetation Index* (NDVI), yang membutuhkan sensor *near-infrared* (NIR) untuk mengukur reflektansi tanaman. Namun, penggunaan NDVI dengan sensor NIR memerlukan peralatan yang mahal dan kompleks, yang tidak selalu tersedia atau terjangkau bagi banyak petani teh. *Green Leaf Index* (GLI) merupakan salah satu alternatif yang

menarik karena tidak bergantung pada sensor *near-infrared*. GLI memanfaatkan informasi spektrum *visible* untuk mengukur tingkat kehijauan daun. Penggunaan GLI memberikan keleluasaan dalam penerapannya tanpa memerlukan peralatan khusus. Dalam konteks ini, penerapan citra *drone* untuk mendapatkan gambaran visual dan analisis tingkat kehijauan menggunakan GLI.

Perkebunan teh sering kali memiliki karakteristik unik, termasuk susunan lajur tanaman teh yang membentang secara luas. Keberhasilan manajemen perkebunan teh memerlukan pemahaman yang mendalam tentang kondisi pertumbuhan di setiap ruas lajur tanaman. Oleh karena itu, pencitraan dan pemetaan yang efisien menjadi krusial untuk memisahkan dan memahami variabilitas dalam setiap area perkebunan terutama tingkat vegetasi tanaman teh yang akan dipetik. Penerapan *image processing* terhadap citra kebun teh hasil tangkapan *drone* dapat menjadi solusi yang dapat digunakan dalam pemetaan tanaman teh demi meminimalisir hasil pemetaan dan analisis dari kegalatan (Svyrydov dkk., 2018). Pengolahan citra/*image processing* merupakan sebuah proses rekayasa gambar yang di mana diolah sedemikian rupa demi mendapatkan hasil interpretasi yang mudah dipahami baik itu oleh manusia maupun komputer (Putri, 2020). Dengan memisahkan entitas tanaman teh dengan objek sekitarnya (tanah, pohon, bebatuan, dsb) tentunya dapat memfokuskan informasi hanya kepada tanaman teh.

Pendekatan *image processing* yang dirasa cocok untuk hal tersebut adalah dengan memanfaatkan algoritma *deep learning* yakni *Mask Region-based Convolutional Neural Network* (Mask R-CNN). *Mask R-CNN* merupakan pengembangan dari *Region-Based Convolutional Neural Networks* (R-CNN) yang awalnya diusulkan oleh Ross Girshick pada tahun 2013 (Girshick dkk., 2013). *Mask R-CNN* dapat melakukan pendekripsi objek dan juga segmentasi terhadap objek dalam citra. Teknik ini dapat diimplementasikan pada citra hasil tangkapan *drone* demi membantu dalam mendekripsi dan memetakan area tanaman teh dan memisahkannya dengan objek atau tanaman sekitar sehingga dapat dilakukan analisa yang lebih terpusat pada tanaman teh.

Dengan begitu diharapkan penelitian ini akan menciptakan sistem pemetaan kesiapan petik tanaman teh pada perkebunan teh sehingga dapat membantu para petani untuk memantau area perkebunan teh secara lebih efektif dan efisien. Penggunaan *drone* dan sistem yang dikembangkan akan membantu meningkatkan kualitas dan produktivitas tanaman teh di Indonesia, sehingga memberikan dampak positif terhadap perekonomian negara serta mendorong pertumbuhan sektor perkebunan dalam industri teh.

## **1.2. Rumusan Masalah**

Pada pembahasan ini, terdapat rumusan masalah yang mencakup inti dari pertanyaan-pertanyaan yang didasarkan pada latar belakang yang telah disajikan sebelumnya. Berikut adalah rumusan masalah yang akan dibahas pada penelitian ini:

1. Bagaimana cara membangun data citra area kebun teh yang lebih luas berdasarkan hasil tangkapan *drone* dengan metode *image stitching*?
2. Bagaimana cara mengimplementasikan metode untuk pemetaan tanaman teh dari citra area perkebunan teh hasil tangkapan *drone* dengan metode *deep learning* yaitu *Mask Region-based Convolutional Neural Network*?
3. Bagaimana cara mengidentifikasi tingkat kehijauan teh menggunakan metode *vegetation index Green Leaf Index* terhadap citra area perkebunan teh hasil tangkapan *drone* untuk kesiapan petik dari tanaman teh?
4. Bagaimana cara menentukan klasifikasi kesiapan petik tanaman teh dari hasil pengolahan citra area perkebunan teh menggunakan *Artificial Neural Network* dengan *Sequential Model*?

## **1.3. Tujuan Penelitian**

Dengan merujuk pada rumusan masalah yang telah disebutkan sebelumnya, adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun data citra area kebun teh yang lebih luas berdasarkan hasil tangkapan *drone* dengan metode *image stitching*.
2. Mengimplementasikan metode untuk pemetaan tanaman teh dari citra area perkebunan teh hasil tangkapan *drone* dengan metode *deep learning* yaitu *Mask Region-based Convolutional Neural Network*.
3. Mengidentifikasi tingkat kehijauan teh menggunakan metode *vegetation index Green Leaf Index* terhadap citra area perkebunan teh hasil tangkapan *drone* untuk kesiapan petik dari tanaman teh.
4. Menentukan klasifikasi kesiapa petik tanaman teh dari hasil pengolahan citra area perkebunan teh menggunakan *Artificial Neural Network* dengan *Sequential Model*.

## **1.4. Batasan Penelitian**

Penelitian ini dibatasi dengan hanya mencangkup ruang lingkup membuat pemetaan dan identifikasi kesiapan petik tanaman teh berdasarkan citra *drone* menggunakan metode

penggabungan citra, segmentasi citra dan perhitungan *vegetation index*. Berikut adalah batasan masalah yang dapat diterapkan pada penelitian ini:

1. Lingkup Perkebunan Teh: Penelitian ini terbatas pada pemetaan dan identifikasi kesiapan petik tanaman teh pada perkebunan teh dengan menggunakan citra *drone*. Penelitian tidak mempertimbangkan jenis tanaman lainnya atau konteks perkebunan yang berbeda.
2. *Dataset*: *Dataset* dibuat dari hasil pengumpulan data citra *drone* yang diambil secara langsung di Pusat Penelitian Teh dan Kina, Gambung, Ciwidey, Kabupaten Bandung yang kemudian data tersebut akan diolah untuk pemetaan dan identifikasi kesiapan petik tanaman teh.
3. Perangkat Lunak: Penelitian ini berfokus pada penggunaan metode penggabungan citra/*image stitching* dengan OpenCV, pemetaan tanaman teh dengan algoritma *Mask R-CNN* menggunakan Detectron2 dan perhitungan *vegetation index* dengan *Green Leaf Index* untuk mengidentifikasi kesiapan petik dari tanaman teh. Selain itu, pengembangan model dilakukan pada platform Google Colab untuk mendapatkan *resource* yang mumpuni. Penggunaan perangkat lunak lain atau implementasi pada platform tertentu tidak termasuk dalam lingkup penelitian ini.
4. Lingkup Pemetaan: Penelitian ini terbatas pada pemetaan dan identifikasi kesiapan petik tanaman teh, yang dihasilkan dari pemetaan area tanaman teh terhadap peta yang merupakan gabungan citra hasil dari tangkapan udara oleh *drone*.

## 1.5. Manfaat Penelitian

Adapun berbagai manfaat yang dapat diperoleh dari penelitian ini yang dituliskan dalam uraian sebagai berikut:

1. Kontribusi pada Bidang Pemetaan Tanaman Teh: Penelitian ini memberikan kontribusi dalam pengembangan teknologi pemetaan dan identifikasi kesiapan petik tanaman teh menggunakan citra *drone*. Hasil penelitian ini dapat membantu meningkatkan efisiensi proses pemetaan tanaman teh, yang pada pemanfaatannya dapat mendukung pengelolaan dan produktivitas yang lebih baik di perkebunan teh.
2. Kontribusi pada Pelatihan Model dengan Data Objek Baru: Penelitian ini juga memberikan kontribusi dalam pembangunan model dengan penambahan objek yang dilatih yakni tamanan teh kedalam *pre-trained model* yang dilatih berdasarkan *dataset* Microsoft COCO (*Common*

*Objects in Context*) yang berisi 91 objek (T.-Y. Lin dkk., 2015) namun belum ada objek untuk tanaman teh di dalamnya.

3. Pengembangan Teknologi Pertanian: Penelitian ini dapat memberikan landasan dan panduan bagi pengembangan teknologi serupa dalam bidang pertanian lainnya. Metode dan kerangka kerja yang diusulkan dapat diterapkan dan dimodifikasi sesuai dengan kebutuhan spesifik pada berbagai jenis tanaman atau sektor pertanian lainnya.
4. Peningkatan Pemahaman tentang Metode dan Model: Penelitian ini memberikan pemahaman yang lebih baik tentang perbandingan dan penerapan metode penggabungan citra/*image stitching*, metode pemetaan tanaman teh/*image segmentation* dengan *Mask R-CNN* dan metode identifikasi *vegetation index* dengan GLI dalam konteks pemetaan dan identifikasi tanaman teh.
5. Penelitian Lebih Lanjut: Hasil penelitian ini dapat menjadi dasar untuk penelitian lebih lanjut dalam bidang pemetaan dan identifikasi kesiapan petik tanaman menggunakan *citra drone*. Peneliti lain dapat mengembangkan atau memperluas penelitian ini dengan mempertimbangkan variabel tambahan, memperluas dan meningkatkan kualitas *dataset* maupun metode atau model yang digunakan, atau bahkan menggabungkan teknik lain untuk meningkatkan kinerja sistem.

Melalui manfaat-manfaat ini, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan teknologi pemetaan dan identifikasi kesiapan petik tanaman teh menggunakan *citra drone* serta memberikan dampak positif bagi industri perkebunan teh dan sektor pertanian secara umum.

## 1.6. Sistematika Penulisan

Sistematika penulisan berfungsi sebagai panduan bagi penulis agar penulisan menjadi lebih terstruktur dan sistematis guna mencapai tujuan akhir dari penelitian. Terdiri dari lima bab, berikut adalah uraian tentang sistematika penulisan:

### BAB I PENDAHULUAN

Bab ini membahas permasalahan penelitian, alasan pemilihan topik, dan elemen-elemen seperti latar belakang masalah, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, serta sistematika penulisan. Fokusnya adalah pada pentingnya kualitas teh untuk ekspor

Indonesia sebagai sumber devisa. Dengan kebun teh yang luas dan pemantauan manual yang memakan waktu, diperlukan teknologi *drone* untuk efektivitas pemantauan. Penerapan *drone* dengan dukungan pengolahan citra, termasuk *image stitching*, memungkinkan pemetaan atau segmentasi objek teh dengan metode *Mask R-CNN* untuk meningkatkan akurasi. Penggunaan *Green Leaf Index (GLI)* dalam perhitungan *vegetation index* dari tangkapan *drone* memberikan informasi relevan terkait kesiapan petik tanaman teh. Teknologi *drone* dan pengolahan citra dengan algoritma canggih dapat menghasilkan data informatif dan dapat diandalkan untuk menentukan kesiapan petik tanaman teh.

## BAB II KAJIAN PUSTAKA

Bab ini membahas teori pendamping dan dukungan untuk penelitian, terkait dengan teh dan teknologi yang akan diterapkan dalam penelitian ini, seperti *drone*, pengolahan citra, *deep learning*, *image stitching*, image segmentation dengan implementasi *Mask R-CNN*, dan *vegetation index* dengan metode GLI. Disajikan juga tabel berisi penelitian terdahulu sebagai dasar acuan, menunjukkan kebaruan dalam penelitian ini.

## BAB III METODOLOGI PENELITIAN

Bab ini membahas metodologi penelitian yang mencakup langkah-langkah mulai dari desain penelitian hingga implementasi dan evaluasi. Metode yang diterapkan adalah *Research and Development (R&D)* fokus pada pengembangan produk atau inovasi dengan langkah-langkah dari penelitian sebelumnya. Langkah-langkah penelitian terdiri dari perumusan masalah, studi literatur, dan identifikasi dan akuisisi kebutuhan data. Kemudian perancangan model, termasuk rancangan penggabungan citra, pemetaan area tanaman teh, dan identifikasi kesiapan petik teh. Selanjutnya pengembangan model rancangan, seperti pengembangan model penggabungan citra, pemetaan tanaman teh, dan identifikasi kesiapan petik teh. Hingga akhirnya ada tahapan *Implementation* (Penerapan) dan *Evaluation* (Evaluasi) melibatkan analisis dan evaluasi hasil, penarikan kesimpulan, dan dokumentasi. Selain itu, bab ini menjelaskan alat dan bahan penelitian yang digunakan, termasuk perangkat keras, perangkat lunak, serta bahan penelitian sebagai pendukung proses penelitian. Rincian alat dan bahan menjadi bagian penting untuk memahami konteks penelitian yang dilakukan.

## BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil dari penelitian yang telah dilakukan. Hasil penelitian didapatkan setelah melalui beberapa proses yang diterapkan. Dimulai dari pengidentifikasi dan akuisisi data dimana ditentukan data yang dibutuhkan adalah data untuk penggabungan citra berupa data citra area perkebunan teh hasil tangkapan *drone* yang telah dilakukan pemilihan seraca visual daerah *overlay* yang dirasa dapat digabungkan satu sama lain. Kemudian juga *dataset* area tanaman teh untuk proses pelatihan model pemetaan area tanaman teh yang dimana *dataset* berupa JSON yang berisikan data informasi dari citra dan koordinat *polygon* yang merepresentasikan area tanaman teh yang ada dalam citra tersebut. *Dataset* tersebut didapatkan melalui proses anotasi/*labelling* secara manual terhadap area tanaman teh yang ada dalam setiap citra. Selain itu *dataset* untuk pelatihan model identifikasi kesiapan petik tanaman teh disiapkan berdasarkan hasil perhitungan *vegetation index* dengan metode *green leaf index* (GLI) yang dibantu dengan data hasil anotasi sebelumnya sehingga menghasilkan data yang terfokus pada sebuah area tanaman teh. *Dataset* tersebut berisikan data-data yang merepresentasikan informasi VI pada area tanaman teh tersebut. Tahapan selanjutnya adalah tahapan perancangan dan pengembangan dari setiap model yang akan digunakan. Dimulai dari model penggabungan citra yang dimana berisikan tujuh tahapan proses dimulai dari *pre-processing*, *finding features*, *matching features*, *outlier removal*, *homography estimation*, *warp images* dan pada akhirnya *blending*. Hasil dari pengembangan model penggabungan citra dapat menggabungkan beberapa citra area perkebunan teh menjadi citra area perkebunan teh yang lebih luas. Selanjutnya model pemetaan area tanaman teh yang menggunakan metode *Mask Region-based Convolutional Neural Network* (*Mask R-CNN*) untuk melakukan pendeksi serta segmentasi objek yang dimana disini akan dilatih untuk mendeksi objek tanaman teh. *Mask R-CNN* sendiri terdiri dari beberapa komponen seperti *Backbone Network*, *Region Proposal Network* (RPN), RoIAlign, dan *Mask head*. *Backbone Network* yang digunakan pada penelitian ini adalah ResNet 50/101, FPN yang merupakan merupakan jaringan saraf konvolusional yang telah dilatih sebelumnya. Kemudian dilakukan pelatihan model berdasarkan *dataset* yang telah disiapkan sebelumnya dan menghasilkan model yang dapat mendeksi dan melakukan pemetaan terhadap area tanaman teh. Adapun *Average Precision* (AP) terbaik didapatkan dari pelatihan dengan *backbone* ResNet-101, dengan iterasi sebanyak 1500 iterasi yang dimana mencapai AP sekitar 39%. Model terakhir adalah model untuk mengidentifikasi kesiapan petik tanaman teh yang dibangun menggunakan *Artificial Neural*

*Network* (ANN) dengan *Sequential Model* yang dimana dibuat satu *input layer*, satu *hidden layer* dan *satu output layer* dalam arsitekturnya. Model tersebut dilatih berdasarkan *dataset* yang berisikan VI hasil perhitungan dengan GLI dan menghasilkan akurasi sebesar 82%, dan nilai *loss* sebesar 0.4202 berdasarkan pelatihan dengan 100 *epochs*. Pada akhirnya dilakukan integrasi terhadap ketiga model tersebut yang mana diawali dengan penggabungan citra dengan *image stitching method* untuk mendapatkan citra area perkebunan teh yang lebih luas. Kemudian data itu menjadi data untuk dilakukan pemetaan terhadap area tanaman teh yang ada didalamnya menggunakan model hasil *training* dengan *Mask R-CNN* dan dihasilkan area tanaman teh yang terdeteksi dalam citra tersebut. Kemudian berdasarkan hasil pendekripsi, dilakukan kalkulasi VI dengan GLI yang kemudian data tersebut dilakukan klasifikasi untuk menentukan apakah area tersebut siap petik atau tidak. Pada akhirnya dihasilkan citra area perkebunan teh beserta dengan informasi area tanaman teh mana yang siap petik dan yang tidak siap petik. Evaluasi dilakukan terhadap hasil citra area perkebunan teh yang siap petik dan tidak siap petik, dan diperlihatkan bahwa model dapat menggabungkan citra dengan baik, kemudian pemetaan hasil model terlihat beberapa yang cukup optimal, namun ada juga yang tidak, dipengaruhi oleh kualitas citra yang kurang baik. Untuk model identifikasi meskipun memiliki akurasi yang cukup tinggi, namun ternyata setelah dilihat kebanyakan mengidentifikasi tanaman menjadi siap petik, hal tersebut bisa terjadi salah satunya karena *dataset* yang disiapkan lebih banyak memiliki data siap petik dibanding tidak siap petik.

## BAB V KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang diperolah peneliti setelah melaksanakan alur penelitian. Pada bab ini juga diberikan saran untuk penelitian kedepannya. Kesimpulan yang diperoleh adalah peneliti mengimplementasikan metode *image stitching* untuk proses penggabungan citra dan menghasilkan citra area perkebunan teh yang lebih luas. Selain itu juga peneliti mengimplementasikan metode *Mask R-CNN* untuk pemetaan area tanaman teh yang adalah citra area perkebunan teh hasil tangkapan *drone* dan berhasil melatih model yang dapat melakukan pemetaan terhadap area tanaman teh. Kemudian peneliti mengimplementasikan metode VI dengan GLI untuk mengidentifikasi tingkat kesiapan petik area tanaman teh dengan yang merupakan hasil klasifikasi dari model *sequential* hasil pelatihan dengan akurasi yang cukup baik. Saran peneliti untuk penelitian selanjutnya yaitu dengan penambahan kuantitas dan keragaman *dataset* untuk

membangun model pemetaan dan identifikasi kesiapan petik tanaman teh yang lebih baik. Selain itu juga bisa dibangun *Grapichal User Interface* (GUI) untuk mempermudah penggunaan dari model-model yang telah dibuat.

## BAB II

### KAJIAN PUSTAKA

#### 2.1. Teh

Teh merupakan salah satu komoditas pertanian yang memiliki peran penting dalam perekonomian Indonesia. Tanaman teh telah lama dikembangkan dan dikonsumsi di banyak negara, termasuk Indonesia. Teh diperoleh dari daun *Camellia sinensis* yang mengandung senyawa bioaktif seperti polifenol, katekin, dan kafein, yang memberikan manfaat kesehatan bagi konsumen (Chadijah dkk., 2021).

Sejarah perkebunan teh di Indonesia dimulai ketika seorang yang memiliki kebangsaan Jerman bernama Andreas Cleyer mengimpor bibit teh dari Jepang untuk ditanam sebagai tanaman hias di Batavia pada tahun 1684, dan saat itu lah tanaman teh pertama kali masuk ke Indonesia. Kemudian Isidorus Loudeijk Levian, seorang ahli teh, melakukan uji coba penanaman secara ekstensif di Banyuwangi dan Purwakarta dan pada akhirnya teh dijadikan sebagai tanaman yang wajib ditanam oleh para petani pada masa pemerintahan Gubernur Van Den Bosh pada tahun 1828 melalui politik tanam paksa (*cultuurstelsel*). Sejak saat itu, teh telah menjadi komoditas dan elemen yang tidak terpisahkan dari budaya Indonesia (Hardoyo, 2013).



Gambar 2.1 Perkebunan teh (Maulida dkk., 2022).

Saat ini, Indonesia menjadi salah satu produsen teh terbesar di dunia dengan perkebunan teh yang tersebar di berbagai wilayah seperti Jawa Barat, Jawa Timur, dan Sumatera. Varietas teh yang umum di Indonesia antara lain adalah teh hijau, teh hitam, dan teh oolong, dengan karakteristik rasa dan aroma yang berbeda (Subdirektorat Statistik Tanaman Perkebunan, 2021).

Proses produksi teh meliputi beberapa tahap, seperti perkebunan, pengumpulan daun teh, pengolahan (pengeringan dan fermentasi), penggulungan, dan pengepakan (Puspasari, 2010). Setiap tahap tersebut memiliki peran penting dalam menghasilkan kualitas teh yang baik. Faktor-faktor seperti iklim, jenis tanah, ketinggian lokasi perkebunan, dan teknik budidaya juga mempengaruhi kualitas dan karakteristik teh yang dihasilkan (Effendi dkk., 2010).

Pentingnya industri teh bagi perekonomian Indonesia tidak dapat dipungkiri. Ekspor teh memberikan kontribusi yang signifikan dalam penerimaan devisa negara (Subdirektorat Statistik Tanaman Perkebunan, 2021). Selain itu, perkebunan teh juga menjadi sumber mata pencaharian bagi banyak petani dan pekerja di sektor pertanian (Maulida dkk., 2022). Oleh karena itu, menjaga kualitas dan produktivitas perkebunan teh menjadi hal yang sangat penting untuk mendukung pertumbuhan ekonomi dan kesejahteraan masyarakat.

Pemantauan/*Monitoring* Perkebunan Teh merupakan aspek penting dalam budidaya teh yang membutuhkan perhatian intensif. Saat ini, pemantauan tanaman teh di Indonesia, masih dilakukan secara manual di mana mandor sebagai pengawas setiap blok kebun akan langsung pergi ke lapangan untuk mengamati atau melakukan survei kondisi tanaman teh (Wirabudi dkk., 2019; Zhang dkk., 2021). Survei lapangan tersebut sangatlah penting karena merupakan saluran pertama dari informasi terkait dengan kondisi area perkebunan teh. Survei lapangan ini mencakup pemantauan distribusi tanaman, pertumbuhan, serta faktor-faktor penting lainnya seperti tinggi pohon dan indeks luas teh (Zhang dkk., 2021). Metode survei ini penting untuk mendapatkan informasi langsung tentang perkebunan teh dan membantu dalam mengembangkan strategi pengelolaan yang lebih baik.

Namun, metode survei lapangan tersebut memiliki keterbatasan. Prosesnya memakan waktu dan tenaga yang cukup banyak, terutama jika perkebunan teh memiliki luas lahan yang besar. Survei lapangan juga sulit diterapkan di daerah berbukit atau pegunungan yang luas, yang di mana merupakan tempat perkebunan teh sering kali berada (Maulida dkk., 2022). Sehingga implementasinya masih terbatas untuk menciptakan pengelolaan yang efisien dan akurat. Selain itu, masalah yang sering muncul adalah respons terhadap gangguan di lapangan yang relatif lambat dan proses estimasi dalam hal operasional juga terkendala (THEREZA dkk., 2020).

## **2.2. Pesawat Nirawak/*Unmanned Air Vehicle (UAV)/Drone***

*Unmanned air vehicle (UAV)* yang dalam bahasa Indonesia disebut dengan pesawat nirawak merupakan teknologi pesawat udara tanpa adanya awak yang biasanya digunakan untuk mendapatkan citra dari udara (Hernina & Putra, 2021). Teknologi yang lebih sering dikenal dengan sebutan *drone* ini biasanya dikendalikan secara *remote* dari jarak jauh baik itu oleh seseorang atau bahkan oleh komputer (Hernina & Putra, 2021).



Gambar 2.2 *Unmanned air vehicle (UAV)/drone* (DJI, 2023).

Pengaplikasian *drone* dapat dipertimbangkan dalam hal *environmental monitoring*/pemantauan lingkungan (Manfreda dkk., 2018), salah satunya adalah untuk pemantauan lingkungan perkebunan (Barbedo, 2019). Menurut (Hernina & Putra, 2021), dibandingkan dengan satelit, penggunaan *drone* ini memiliki banyak kelebihan, antara lain:

1. Bebas dari gangguan awan dan atmosfer, penggunaan *drone* untuk pemetaan di tempat yang ketinggian terbangnya rendah, menghasilkan foto yang bebas dari awan dan bebas dari gangguan atmosfer. Ketinggian terbang yang dapat diatur dan relatif rendah menghasilkan citra udara yang bebas dari gangguan awan dan juga atmosfer. Sehingga citra udara yang didapatkan dan objek yang dipantau di dalamnya menjadi lebih jelas.
2. Resolusi yang tinggi, citra udara berkualitas tinggi dihasilkan oleh ketinggian terbang *drone* yang rendah, antara 100 hingga 150 meter. Resolusi pada *drone* tentunya dapat diubah, begitu pula dengan ketinggian terbangnya dapat diubah. Resolusi yang diberikan oleh *drone* sama atau bahkan lebih tinggi daripada resolusi satelit tinggi jika satelit resolusi menengah dapat menghasilkan resolusi 15 hingga 30 meter. Contoh perbedaan hasil tangkapan gambar antara *drone* dan satelit dapat dilihat pada Gambar 2.3.



Gambar 2.3 Perbandingan antara hasil tangkapan *drone* dan satelit (Satish, 2022).

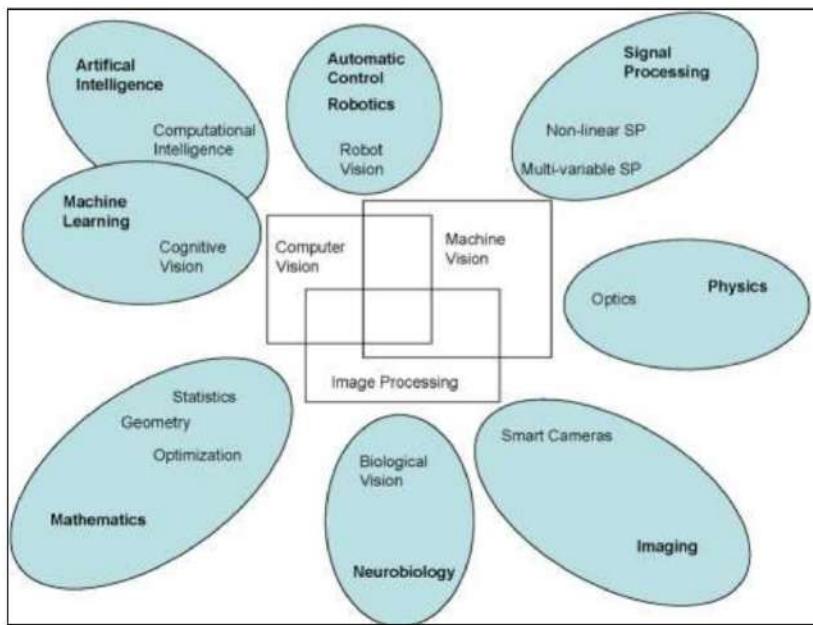
3. Relatif lebih murah, penggunaan *drone* untuk melakukan pemetaan lebih murah daripada menggunakan satelit yang memiliki resolusi tinggi. Tentunya untuk didapatkan resolusi yang sama, satelit beresolusi tinggi jauh lebih mahal. Selain itu, biaya operasional yang lebih murah dan minim risiko karena tanpa awak dibandingkan dengan fotografi udara tradisional.
4. Efisien, dari segi waktu, penggunaan *drone* dapat menjadi lebih efektif karena data yang kita perlukan sesuai dengan wilayah yang kita butuhkan. Drone dapat dibawa langsung ke lokasi di mana dibutuhkannya data dan digunakan untuk memetakan area penelitian yang diperlukan.
5. Mampu menjangkau lokasi/area yang tidak dapat dijangkau oleh satelit dan pesawat besar.

### 2.3. *Computer Vision*

*Computer vision* adalah sebuah bidang studi yang menggabungkan pengolahan citra/*image processing*, pengenalan pola/*pattern recognition*, *machine learning* bahkan *computer graphics* (Wiley & Lucas, 2018). Tujuan utamanya adalah memberikan kemampuan kepada komputer untuk secara otomatis melihat mengidentifikasi dan memahami dunia visual seperti manusia (Feng dkk., 2019). Pengembangan bidang ini dilakukan dengan mengadaptasi kemampuan sistem penglihatan manusia dalam mengambil dan memahami informasi berdasarkan data visual (Wiley & Lucas, 2018; Xu dkk., 2021). Dalam *computer vision*, pengolahan citra menjadi kunci utama dalam mendapatkan informasi dan pengetahuan dari citra atau video yang diperoleh.

Berbagai teknik dan metode digunakan dalam *computer vision* untuk mengenali, menganalisis, dan memahami informasi visual yang terkandung dalam citra atau video (Xu dkk., 2021). Beberapa teknik yang umum digunakan meliputi segmentasi citra/*image segmentation*, deteksi objek/*object detection*, pengenalan pola/*pattern recognition*, pelacakan objek, dan rekonstruksi 3D (O' Mahony dkk., 2019; Xu dkk., 2021). Proses-proses untuk memperoleh

pemahaman yang mendetail tentang data visual menggunakan komputer dikategorikan menjadi tiga tahapan yang berurutan, yaitu mengilustrasikan ekstraksi fitur-fitur dasar dalam data visual (misalnya, informasi warna, tekstur, dan kedalaman), pengolahan/pemrosesan, dan pembuatan informasi semantik (Xu dkk., 2021).



Gambar 2.4 Pohon topik penelitian terkait citra dan *vision* (Shahab dkk., 2007).

Dapat dilihat pada Gambar 2.4 yang merupakan gambaran dari topik penelitian yang berkaitan dengan citra dan *vision*. Pada Gambar 2.4 digambarkan bahwa topik penelitian *computer vision* memiliki singgungan dengan topik lain seperti *image processing* dan *machine vision*. Sehingga dapat dikatakan bahwa ketiga hal tersebut saling berkaitan erat satu sama lain dalam proses penelitian demi menghasilkan solusi yang terbaik.

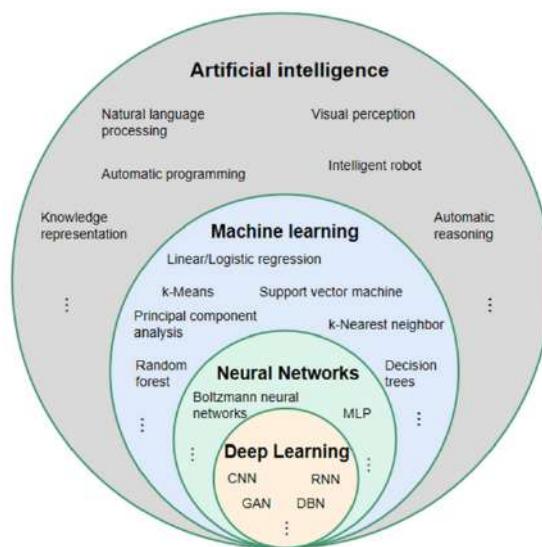
Pengolahan citra atau *image processing* merupakan sebuah proses memanipulasi suatu sebagian atau keseluruhan piksel dalam sebuah citra demi tercapainya tujuan tertentu (Putri, 2020). Proses manipulasi dan interpretasi citra ini dilakukan dengan cara memanfaatkan komputer atau bisa disebut dengan *computational transformation* (Pratama, 2018; Wiley & Lucas, 2018). Transformasi yang dilakukan bisa berupa penajaman, pengaturan kontras, penonjolan fitur, kompresi maupun koreksi terhadap ketidakjelasan atau *blur* (Putri, 2020; Wiley & Lucas, 2018). Tujuan dari pengolahan citra adalah untuk melakukan perbaikan citra atau melakukan penyesuaian citra sehingga citra dapat dianalisa dan menghasilkan informasi yang dibutuhkan (Putri, 2020; Wiley & Lucas, 2018).

Menurut (Pratama, 2018), citra berdasarkan jenisnya bisa dikelompokkan menjadi dua jenis, yakni citra kontinu dan citra diskrit atau citra digital. Citra kontinu dihasilkan dari sistem optik dengan sinyal analog, manusia dan kamera analog adalah contoh sistem optik yang menerima sinyal analog secara terus-menerus, sedangkan citra diskrit atau digital merupakan citra kontinu yang dikonversikan melalui proses digitalisasi sehingga menjadi gambar diskrit (gambar digital). Karena matriks digunakan untuk merepresentasikan gambar digital, maka, Pengolahan citra atau *image processing* pada dasarnya melibatkan manipulasi elemen matriks dalam bentuk piksel.

#### 2.4. Kecerdasan Buatan/Artificial Intelligence

Kecerdasan buatan atau *artificial intelligence* (AI) termasuk kedalam revolusi keempat dalam sejarah umat manusia (Ting dkk., 2019). Dalam konteks ini, kecerdasan buatan (AI) dapat didefinisikan sebagai ilmu dan rekayasa *intelligent machines* atau mesin cerdas dengan fokus khusus pada *intelligent computer programs* atau program komputer yang cerdas (Woschank dkk., 2020). Bidang ilmu komputer ini didedikasikan untuk menciptakan sistem yang dapat melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia atau dapat dikatakan bahwa kecerdasan buatan adalah hasil dari penggabungan antara kecerdasan manusia kedalam mesin.

AI mencakup berbagai teknik dan metode, termasuk *machine learning* (ML), dan *neural networks* (NN), *deep learning* (DL) yang digunakan untuk mengembangkan model dan algoritma yang dapat belajar dan beradaptasi dari data (Jakhar & Kaur, 2020). Gambaran dari hubungan antara ketiga komponen tersebut diperlihatkan pada Gambar 2.5 yang juga menampilkan metode dari masing-masing komponen yang disebutkan sebelumnya.



Gambar 2.5 Hubungan antara AI, ML, NN, dan DL (S. Li dkk., 2021).

#### **2.4.1 Machine Learning**

*Machine learning* (ML) adalah cabang bidang studi dari ilmu komputer yang merupakan bagian dari kecerdasan buatan (AI). *Machine learning* melibatkan pendekatan-pendekatan yang membuat mesin memiliki kemungkinan untuk belajar dari data tanpa harus deprogram secara eksplisit (Jakhar & Kaur, 2020). *Machine learning* dapat mendeteksi pola-pola bermakna secara otomatis dalam suatu dataset (Woschank dkk., 2020). Berdasarkan Gambar 2.5 *machine learning* memiliki beberapa metode dalam implementasinya seperti *Linear/Logistic regression*, *k-Means*, *k-Nearest Neighbor*, *Random Forest*, *Decision Tree* dan lain sebagainya (S. Li dkk., 2021).

Tujuan dari *machine learning* adalah melatih mesin berdasarkan data dan algoritma yang diberikan. Dengan menggunakan data dan informasi yang diproses, mesin-mesin tersebut belajar bagaimana membuat keputusan. *Machine learning* bersifat dinamis, yang berarti memiliki kemampuan untuk memodifikasi dirinya sendiri ketika terpapar oleh lebih banyak data (Jakhar & Kaur, 2020).

Aspek "learning" atau "pembelajaran" dalam *machine learning* mengacu pada upaya algoritma *machine learning* untuk meminimalisir kesalahan dan memaksimalkan probabilitas prediksinya menjadi benar sehingga meningkatkan efisiensi algoritma dan memastikan kemampuan belajar dan adaptasi berdasarkan analisis data (Woschank dkk., 2020). Sehingga secara singkat, dapat dikatakan bahwa *machine learning* merupakan teknik untuk mewujudkan sebuah kecerdasan buatan (AI) (Jakhar & Kaur, 2020).

#### **2.4.1 Deep Learning**

*Deep learning* (DL) merupakan teknik *machine learning* yang sangat populer dalam beberapa tahun terakhir. Menurut (Ting dkk., 2019), *deep learning* adalah kelas teknik *machine learning* terbaru yang telah menarik minat global secara besar-besaran. Secara formal, *deep learning* adalah subset atau bagian dari kecerdasan buatan (*artificial intelligence*) yang didefinisikan sebagai "model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk belajar representasi data dengan tingkat abstraksi yang berbeda" (Nagendran dkk., 2020). Definisi ini menunjukkan bahwa *deep learning* menggunakan arsitektur jaringan saraf tiruan (*artificial neural network/ANN*) dengan banyak lapisan untuk mempelajari dan menggambarkan data dengan tingkat abstraksi yang lebih tinggi. Sehingga istilah "*deep*" dalam *deep learning* merujuk pada jumlah lapisan (*layers*)

dalam ANN. Lebih banyak lapisan dapat memberikan model yang lebih kompleks dan mampu menjelajahi struktur data yang lebih rumit (Jakhar & Kaur, 2020) sehingga lebih mampu untuk ekstraksi fitur dan transformasi bertujuan pengawasan maupun tanpa pengawasan serta analisis pola dan klasifikasi data (Woschank dkk., 2020).

Dalam konteks biologi, DL mencoba meniru arsitektur jaringan saraf biologis di otak manusia. Seperti disebutkan oleh Jakhar & Kaur (2020), ketika otak menerima informasi baru, ia mencoba membandingkannya dengan informasi yang telah diketahui sebelumnya untuk mencoba memahaminya. Otak membaca informasi melalui pelabelan dan mengkategorikan item ke dalam berbagai kategori, dan DL menggunakan konsep yang sama. DL menggunakan metode pembelajaran representasi dengan tingkat abstraksi yang berbeda untuk memproses data masukan tanpa perlu melakukan rekayasa fitur secara manual. Hal ini memungkinkan DL untuk secara otomatis mengenali struktur-struktur rumit pada data dengan dimensi tinggi melalui proyeksi ke *manifold* dimensionalitas rendah (Ting dkk., 2019). Salah satu jenis algoritma *deep learning* adalah *convolutional neural network* (CNN), sering digunakan dalam pemrosesan gambar atau *image processing* (Nagendran dkk., 2020). Struktur dari CNN ini merupakan salah satu jenis jaringan saraf tiruan dalam *deep neural network*.

Secara sederhana, DL sangat efektif dalam mengolah data tak terstruktur dan memiliki akurasi yang jauh lebih tinggi dibandingkan dibandingkan teknik konvensional seperti *machine learning* tradisional. Dalam banyak domain seperti pemrosesan bahasa alami, pengenalan citra komputer, dan pengenalan suara (Ting dkk., 2019). Namun, DL juga membutuhkan data latih dengan volume besar serta perangkat keras dan perangkat lunak yang mahal (Jakhar & Kaur, 2020).

#### **2.4.3 Artificial Neural Networks (ANN)**

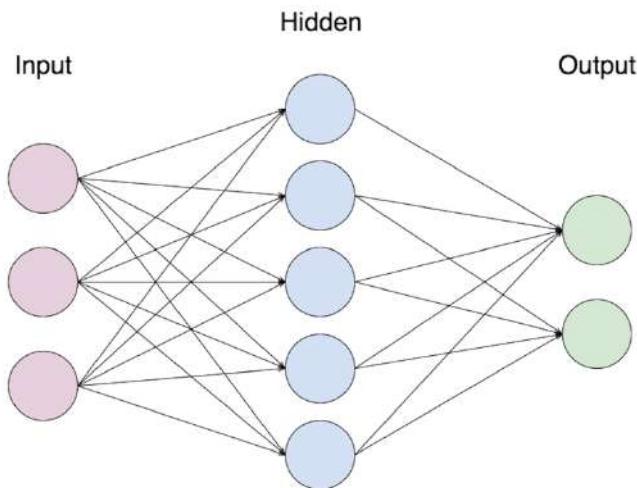
*Artificial Neural Networks* (ANN) merupakan salah satu pendekatan dalam bidang kecerdasan buatan yang terinspirasi dari struktur dan fungsi jaringan saraf biologis manusia. ANN terdiri dari serangkaian unit pemrosesan yang disebut neuron atau node, yang saling terhubung melalui koneksi bobot. Model ini mampu memetakan pola kompleks dan melakukan tugas-tugas seperti pengenalan pola, klasifikasi, regresi, dan prediksi (Zou dkk., 2008). Berikut karakteristik utama dari *Artificial Neural Network* (ANN):

1. Arsitektur neuron, ANN terdiri dari lapisan-lapisan neuron yang terhubung, yaitu lapisan input, lapisan tersembunyi, dan lapisan output. Neuron dalam lapisan *input* menerima sinyal

masukan, lapisan tersembunyi melakukan pemrosesan, dan lapisan *output* menghasilkan hasil prediksi.

2. Koneksi bobot, setiap koneksi antar neuron memiliki bobot yang menentukan kekuatan dan arah pengaruhnya. Bobot diubah selama proses pelatihan untuk meminimalkan kesalahan prediksi.
3. Fungsi aktivasi, neuron menggunakan fungsi aktivasi untuk mengonversi input menjadi *output*. Fungsi ini memperkenalkan non-linearitas, memungkinkan jaringan untuk memodelkan hubungan yang kompleks.
4. Pelatihan dan pembelajaran, ANN dilatih menggunakan algoritma pembelajaran yang dapat disesuaikan, seperti *backpropagation*, yang memperbarui bobot berdasarkan selisih antara prediksi dan target.

Model *Sequential* adalah struktur ANN yang mengatur lapisan-lapisan secara berurutan, di mana informasi mengalir sejajar dari lapisan input ke lapisan output (Manaswi, 2018). Model *Sequential* dapat diimplementasikan menggunakan Keras TensorFlow yang merupakan *library* untuk *machine learning*. *Sequential model* didasarkan pada konsep urutan lapisan ini adalah penggunaan paling umum dari Keras dan bagian paling mudah dari Keras (Manaswi, 2018). Gambaran dari arsitektur *sequential model* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Arsitektur dari *Artificial Neural Network* dengan *Sequential Model*.

Model *sequential* dapat dianggap sebagai tumpukan linear dari lapisan-lapisan. Singkatnya, model *sequential* merupakan model di mana dengan mudah dapat menambahkan lapisan, dan

setiap lapisan dapat memiliki konvolusi, *max pooling*, aktivasi, *dropout*, dan normalisasi *batch* (Manaswi, 2018). Adapun karakteristik dari model ini melibatkan:

1. *Sequential Layering*, lapisan-lapisan diatur secara berurutan, menciptakan jalur aliran informasi dari input ke output.
2. Fleksibilitas Struktural, model Sequential memungkinkan penambahan dan pengaturan lapisan dengan mudah, memfasilitasi eksperimen dengan berbagai arsitektur jaringan.
3. Kemudahan Implementasi, kepatuhan terhadap model *sequential* membuatnya lebih mudah diimplementasikan dan dikelola, baik untuk pemula maupun pengguna berpengalaman.

#### **2.4.3 Convolutional Neural Networks (CNN)**

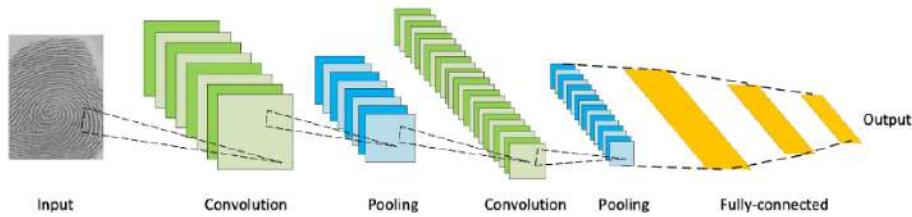
Menurut (Gollapudi, 2019) *Convolutional Neural Networks* (CNN) merupakan variasi dari jaringan saraf biasa yang memiliki keunggulan dalam menyelesaikan tugas *computer vision* ketika terdapat variasi yang luas dalam gambar. CNN didesain secara struktural mengikuti susunan hierarkis neuron di dalam otak manusia (Nagendran dkk., 2020).

Salah satu kelebihan CNN adalah kemampuannya untuk memodelkan informasi pada bagian-bagian kecil dan menggabungkannya menggunakan jaringan mendalam, hal ini dikemukakan oleh (Gollapudi, 2019). Proses ini terjadi melalui beberapa lapisan di mana lapisan pertama melakukan *edge detection/deteksi tepi*, mencari batas dan membuat *template*. Lapisan berikutnya menggunakan *template* tersebut sebagai dasar, mengambil bentuk yang lebih sederhana dari gambar, dan membentuk *template-template* baru yang mencakup skala objek, posisi, atau pencahayaan yang berbeda-beda. Lapisan terakhir mencocokkan gambar input dengan semua *template* tersebut, dan *output* akhir merupakan jumlah tertimbang dari semua keluaran tersebut. Hal ini membantu penanganan variasi kompleks pada gambar dengan tingkat akurasi yang lebih tinggi.

Gollapudi (2019) menjelaskan bahwa CNN memiliki tiga jenis lapisan yaitu:

1. *Convolution layer*
2. *Pooling layer*
3. *Fully connected layer*

Adapun gambaran dan alur dari masing-masing lapisan dapat dilihat pada gambar 2.7.



Gambar 2.7 Gambaran arsitektur *convolutional neural networks* (CNN) (Minaee dkk., 2020).

Pada Gambar 2.7 digambarkan arsitektur untuk CNN yang di mana terdapat beberapa lapisan terdiri dari tiga lapisan yang disebutkan sebelumnya yakni *convolution layer*, *pooling layer* dan *fully connected layer*. Lapisan-lapisan tersebut disesuaikan dengan kebutuhan dari pembangunan CNN. Lapisan pertama untuk CNN adalah lapisan untuk melakukan proses konvolusi gambar atau data secara umum menggunakan *filters* atau *kernels*. *Filters* adalah unit kecil diterapkan pada gambar atau data melalui *sliding window*. Kedalaman *filter* ini sama dengan kedalaman data *input*, jadi jika *input* adalah gambar RGB maka nilai kedalamannya adalah tiga (*Red*, *Green*, *Blue*), sehingga *filter* yang diterapkan juga memiliki kedalaman tiga. Operasi konvolusi melibatkan pengambilan hasil perkalian antar elemen *filter* dalam gambar dan kemudian menjumlahkan nilai-nilai tersebut untuk setiap *sliding action*.

Selanjutnya adalah *pooling layers* yang melibatkan *downsampling* fitur sehingga perlu mempelajari lebih sedikit parameter selama pelatihan. Biasanya ada dua *hyperparameter* yang diperkenalkan dengan lapisan penyatuhan, yakni:

1. Dimensi luas spasial (*the dimensions of the spatial extent*), dengan kata lain nilai N yang mana kita dapat mengambil representasi fitur  $N \times N$  dan memetakannya ke satu nilai.
2. *Stride* atau langkah yang merupakan jumlah fitur yang dilewati oleh *sliding window* sepanjang lebar dan tinggi yang mirip dengan konvolusi.

Secara umum *pooling layers* menggunakan *filter*  $2 \times 2$  *max* dengan nilai *stride* 2, *filter* ini adalah *filter* yang tidak tumpang tindih (*non-overlapping filter*). *Filter max* mengembalikan nilai maksimum di antara fitur-fitur di wilayah tersebut. Ada pula *Filter rata-rata (Average)* yang mengembalikan rata-rata fitur di wilayah tersebut juga, tetapi secara praktik *max pooling* bekerja lebih baik. Karena penggabungan diterapkan melalui setiap lapisan dalam volume 3D, kedalaman peta fitur setelah penggabungan tidak akan berubah. Melakukan *pooling* mengurangi

kemungkinan *overfitting* karena jumlah parameter yang lebih sedikit, karena dapat mengurangi jumlah fitur yang sangat signifikan dalam jumlah parameter yakni 25% dari jumlah aslinya.

Keluaran dari lapisan *convolutional* merepresentasikan fitur tingkat tinggi dalam data. Meskipun *output* tersebut dapat diratakan (*flattened*) dan dihubungkan ke lapisan *output*, menambahkan *fully connected layer* biasanya merupakan cara yang murah untuk mempelajari kombinasi nonlinear dari fitur-fitur ini. Pada dasarnya, lapisan *convolutional* menyediakan ruang fitur berdimensi rendah yang bermakna dan agak invariant dan *fully connected layer* mempelajari fungsi nonlinear yang mungkin ada di ruang tersebut. Karena *output* dari *pooling layer* adalah peta fitur 3D (volume fitur 3D). Namun, masukan untuk *fully connected feed-forward neural network* yang sederhana adalah vektor fitur satu dimensi. Maka cara mengubah *output* dari *pooling layer* menjadi *input* untuk lapisan *fully connected layer* adalah dengan mengubah keluaran lebar dan tinggi volume 3D ini menjadi satu dimensi, hal ini dilakukan dengan meratakan *layer* 3D ini menjadi vektor 1D yang dimana lebar x tinggi x volume.

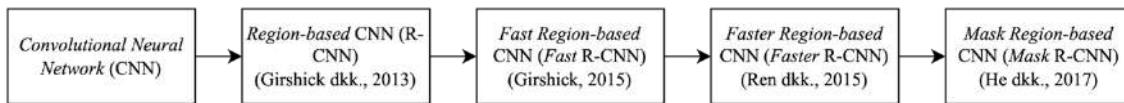
Sedangkan (Nagendran dkk., 2020) menjelaskan perbedaan utama antara *deep learning* dan jenis *machine learning* lainnya adalah bahwa CNN mengembangkan representasi-representasi yang diperlukan untuk pengenalan pola tanpa memerlukan masukan manusia dalam struktur data dan desain ekstraktor fitur. Algoritma pada CNN belajar sendiri fitur-fitur penting dari gambar untuk klasifikasi.

Dengan demikian, algoritma memiliki kebebasan untuk menemukan fitur-fitur klasifikasi yang mungkin tidak tampak bagi manusia (terutama ketika *dataset* cukup besar) dan dengan demikian meningkatkan performa klasifikasi gambar. (Nagendran dkk., 2020) juga menjelaskan bahwa CNN menggunakan data gambar mentah yang telah dilabeli oleh manusia melalui proses pembelajaran supervisi. Setiap citra dimasukkan ke lapisan *input* algoritma sebagai piksel-piksel mentah dan kemudian diproses secara berurutan melalui lapisan-lapisan CNN. *Output* akhir merupakan kemungkinan klasifikasi citra tersebut termasuk dalam kelompok tertentu.

#### **2.4.4 Mask R-Convolutional Neural Networks (CNN)**

Mask R-CNN merupakan salah satu *framework* untuk *object detection* terkini yang sangat canggih (He dkk., 2017). *Framework* ini merupakan pengembangan dari detektor jaringan saraf berbasis wilayah yang awalnya diusulkan oleh Ross Girshick pada tahun 2015 (Girshick, 2015).

Untuk lebih jelasnya terkait alur dari perkembangan awal R-CNN hingga ke *Mask R-CNN* dapat dilihat pada Gambar 2.8.

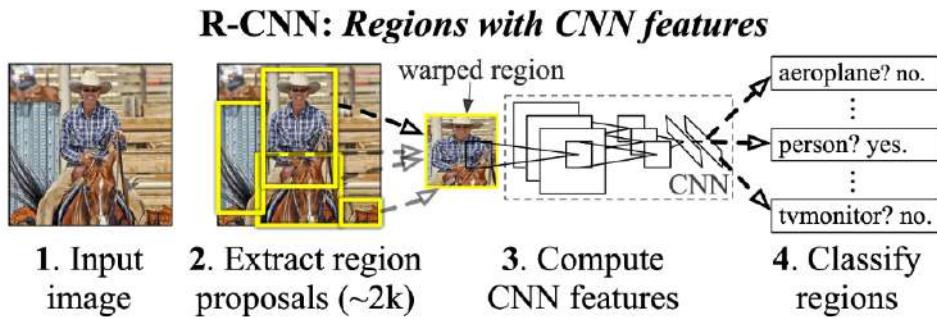


Gambar 2.8 Alur perkembangan R-CNN hingga *Mask R-CNN*.

Sejarah dikembangkannya R-CNN berangkat dari masalah dari implementasi CNN dengan *fully connected layers* yang tidak dapat menangani frekuensi kemunculan dan multi objek. Salah satu cara untuk menangani masalah dari CNN ini adalah dengan menggunakan *sliding window brute force search* untuk memilih suatu wilayah dan menerapkan model CNN pada wilayah tersebut. Akan tetapi pendekatan solusi tersebut juga memunculkan masalah karena objek yang sama dapat diwakili dalam gambar dengan ukuran yang berbeda dan rasio aspek yang berbeda, sehingga harus memilih wilayah/*regions* yang sangat banyak dan jika menerapkan *deep learning* (CNN) pada semua wilayah tersebut, maka secara komputasi akan sangat mahal.

Untuk mengatasi masalah dalam memilih wilayah/*regions* yang sangat banyak, (Girshick dkk., 2013) mengusulkan sebuah metode di mana kita menggunakan *selective search*/pencarian selektif untuk mengekstrak hanya 2000 wilayah dari gambar yang disebut dengan *region proposals*/proposal wilayah. Gambaran dari sistem deteksi objek dengan R-CNN dapat dilihat pada Gambar 2.9. Oleh karena itu, sekarang, alih-alih mencoba mengklasifikasikan sejumlah besar wilayah, klasifikasi bisa difokuskan pada 2000 wilayah tersebut. 2000 proposal wilayah ini dihasilkan dengan menggunakan algoritma pencarian selektif dengan proses sebagai berikut:

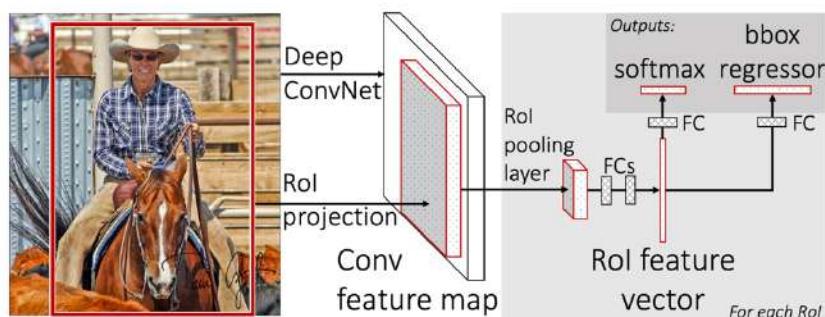
1. *Generating initial sub-segmentation*, menghasilkan banyak wilayah kandidat.
2. Menggunakan algoritma *greedy* untuk menggabungkan wilayah-wilayah yang serupa secara rekursif menjadi wilayah yang lebih besar.
3. Menggunakan wilayah yang dihasilkan untuk menghasilkan *final candidate region proposals*.



Gambar 2.9 Gambaran umum deteksi objek menggunakan R-CNN (Girshick dkk., 2013).

Berdasarkan Gambar 2.9, (Girshick dkk., 2013) menuliskan bahwa sistem yang mereka buat memiliki empat proses utama yakni (1) menerima masukan berupa gambar yang kemudian (2) dilakukan ekstraksi terhadap 2000 proposal wilayah dari bawah ke atas. Setelah itu dilanjutkan dengan (3) menghitung fitur untuk setiap proposal menggunakan *convolutional neural network* (CNN) yang besar, dan pada akhirnya (4) mengklasifikasikan setiap wilayah menggunakan *class-specific linear SVMs*.

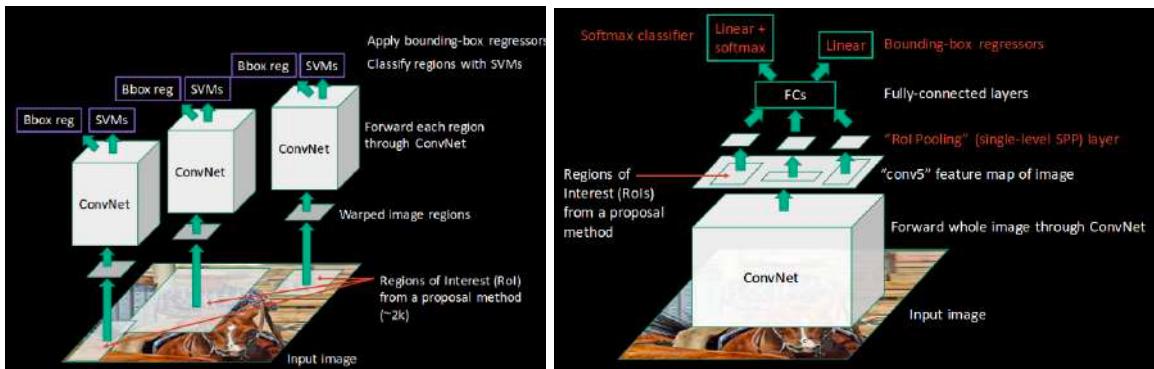
Dalam perkembangannya, R-CNN memiliki beberapa masalah terkait implementasinya yakni yang pertama adalah, masih membutuhkan waktu yang sangat lama untuk melatih jaringan karena harus mengklasifikasikan 2000 proposal wilayah per gambar. Kemudian tidak dapat diimplementasikan secara *real time* karena membutuhkan waktu cukup lama untuk setiap gambar uji. Sehingga pada 2015 (Ren dkk., 2015) dapat mengatasi beberapa kelemahan R-CNN untuk membangun algoritma pendekripsi objek yang lebih cepat dan disebut Fast R-CNN. Gambaran arsitektur Fast R-CNN dapat dilihat pada gambar 2.10 dibawah ini:



Gambar 2.10 Gambaran arsitektur *Fast R-CNN* (Ren dkk., 2015).

Pendekatannya mirip dengan algoritma R-CNN. Namun, alih-alih memasukkan proposal wilayah ke CNN, (Ren dkk., 2015) memasukkan gambar *input* ke CNN untuk menghasilkan peta

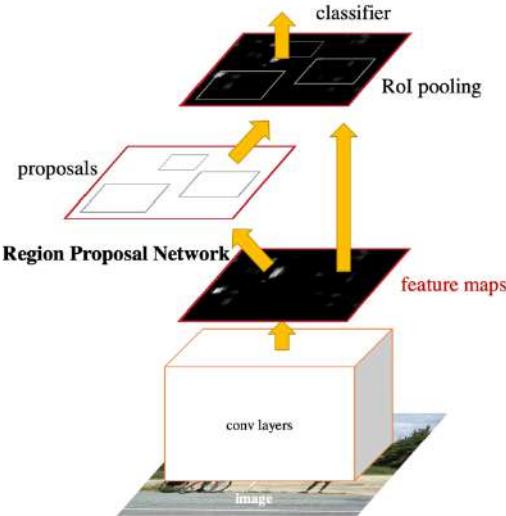
fitur konvolusi/*convolutional feature map*. Dari peta fitur konvolusional, proses selanjutnya adalah mengidentifikasi wilayah proposal dan melakukan *warping* wilayah tersebut menjadi kotak-kotak dan dengan menggunakan *region of interest (RoI) pooling layer*, dibentuklah menjadi ukuran yang tetap sehingga dapat dimasukkan ke dalam *fully connected layer*. Dari RoI *feature vector*, kemudian digunakanlah *softmax layer* untuk memprediksi kelas dari wilayah yang diusulkan dan juga *also offset values* untuk *bounding box*. Gambar 2.11 menggambarkan lebih jelas perbedaan dari arsitektur R-CNN dan Fast R-CNN.



Gambar 2.11 Perbedaan antara arsitektur R-CNN dan Fast R-CNN (P. P. Sharma, 2021).

Alasan mengapa *Fast R-CNN* itu “*fast*” atau lebih cepat daripada R-CNN adalah karena pada *Fast R-CNN* tidak perlu memasukkan 2000 *region proposals* ke *convolutional neural network* setiap saat. Sebaliknya, operasi konvolusi dilakukan hanya sekali per gambar dan peta fitur dihasilkan oleh operasi tersebut seperti yang dapat dilihat pada Gambar 2.10. Akan tetapi pada Gambar 2.11 diperlihatkan bahwa R-CNN dan *Fast R-CNN* sama-sama menggunakan *selective search* untuk menentukan *region proposals*. Algoritma pencarian selektif adalah algoritma yang tetap (*fixed algorithm*), tidak ada pembelajaran yang terjadi pada tahap tersebut yang dapat menyebabkan dihasilkannya proposal wilayah kandidat yang buruk. Pencarian selektif juga merupakan proses yang lambat dan memakan waktu yang mempengaruhi kinerja jaringan.

Oleh karena itu, (Ren dkk., 2015) pada penelitiannya membuat algoritma pendekripsi objek yang menghilangkan algoritma pencarian selektif dan membiarkan jaringan mempelajari proposal wilayah yang dinamakan *Faster R-CNN*. Dapat dilihat pada Gambar 2.12 yang menunjukkan gambaran dari arsitektur *Faster R-CNN*.



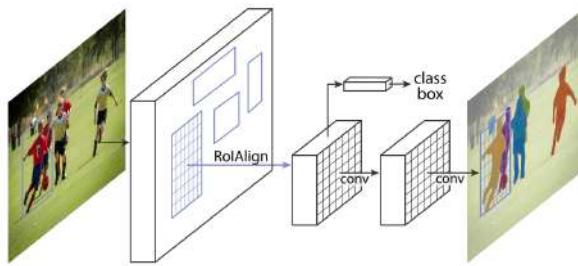
Gambar 2.12 Arsitektur *Faster R-CNN* (Ren dkk., 2015).

Sama seperti *Fast R-CNN*, proses diawali dengan mengekstrak peta fitur/*convolutional feature map* dari gambar *input* menggunakan *convolutional network*. Proses selanjutnya alih-alih menggunakan algoritma pencarian selektif pada peta fitur untuk mengidentifikasi proposal wilayah seperti yang digunakan *Fast R-CNN*, *Faster R-CNN* menggunakan jaringan terpisah digunakan untuk memprediksi proposal wilayah/*region proposals* yang disebut *Region Proposal Network* (RPN). Prediksi *region proposals* kemudian dibentuk ulang menggunakan *RoI pooling layer* yang kemudian digunakan untuk mengklasifikasikan gambar di dalam wilayah yang diusulkan dan memprediksi nilai *offset* untuk *bounding boxes*.

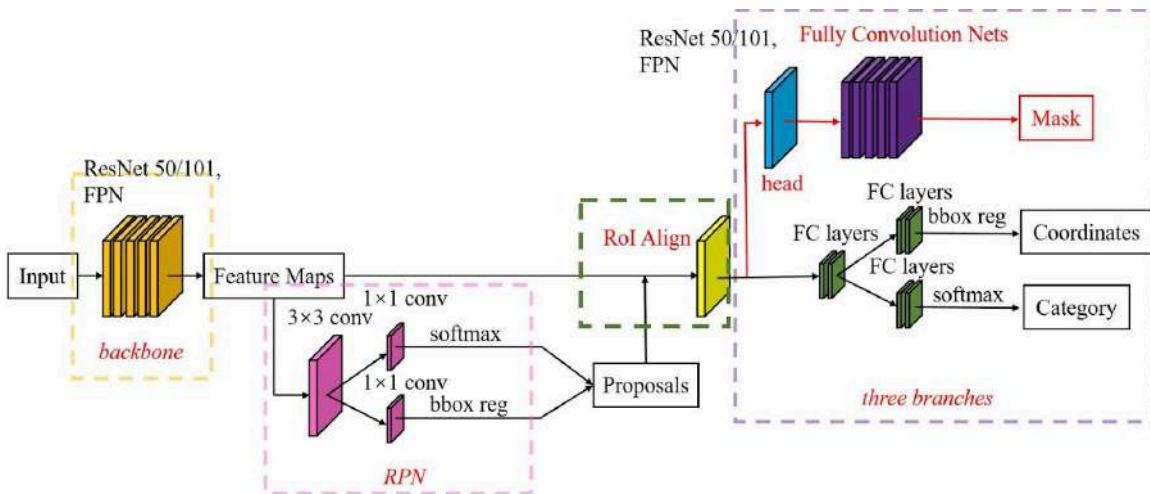
Sekarang kembali lagi pada *Mask R-CNN* yang di mana merupakan pengembangan dari *Faster R-CNN* yang dapat melakukan *object detection* (*bounding box*). Pada *Mask R-CNN*, model dapat juga melakukan yang dinamakan *segmentation*. *Mask R-CNN* merupakan pendekatan terhadap *instance segmentation* yang di mana adalah kombinasi dari dua turunan masalah, yakni:

1. *Object detection*
2. *Semantic segmentation*

*Mask R-CNN* menggabungkan *object detection* dan *semantic segmentation* menjadi satu prediksi untuk seluruh gambar (He dkk., 2017). Hal ini membuatnya sangat baik dalam melakukan *instance segmentation*. Arsitektur dari *Mask R-CNN* dapat dilihat pada Gambar 2.13 dan untuk lebih detailnya dapat diperhatikan pada Gambar 2.14.

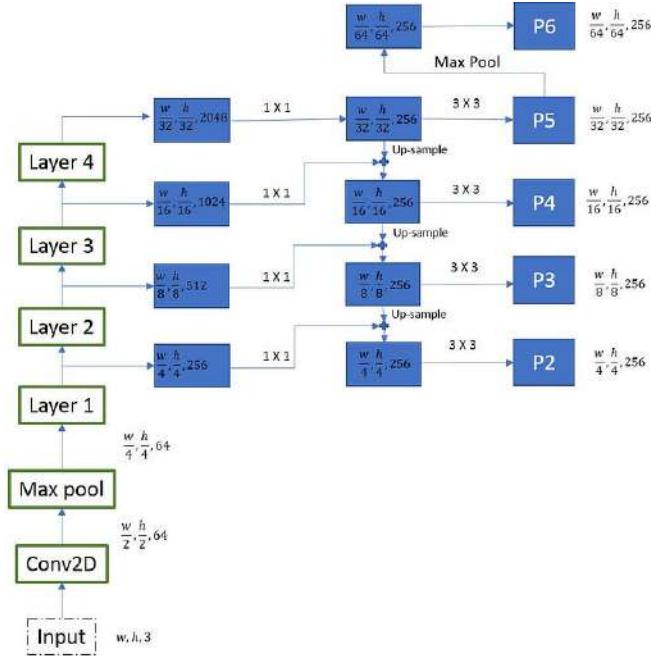


Gambar 2.13 Arsitektur *Mask R-CNN* (He dkk., 2017).



Gambar 2.14 Detail arsitektur *Mask R-CNN* untuk *image segmentation* (J. Lin dkk., 2022).

Berdasarkan Gambar 2.14, arsitektur *Mask R-CNN* terbagi menjadi empat bagian yakni *Backbone Network*, *Region Proposal Network* (RPN), RoIAlign, dan *Mask head*. *Backbone Network* yang digunakan pada penelitian ini adalah ResNet 50/101, FPN yang merupakan merupakan jaringan saraf konvolusional yang telah dilatih sebelumnya. Lebih jelasnya dapat dilihat pada Gambar 2.15 untuk *backbone* dari FPN (He dkk., 2017; Jiao dkk., 2019).

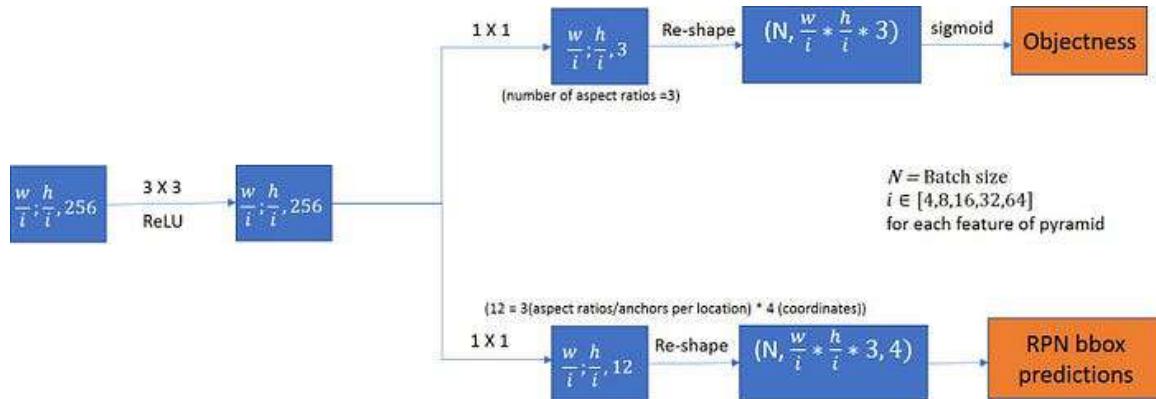


Gambar 2.15 *Feature Pyramid Networks (FPN) backbone* (Subash dkk., 2020).

FPN dirancang untuk mengatasi tantangan menangani objek dengan berbagai ukuran dan skala dalam suatu citra. Arsitektur FPN menciptakan piramida fitur multi-skala dengan menggabungkan fitur-fitur dari berbagai tingkat jaringan utama. Piramida ini mencakup fitur dengan resolusi spasial yang berbeda, mulai dari fitur beresolusi tinggi dengan informasi semantik kaya hingga fitur beresolusi rendah dengan detail spasial yang lebih tepat. Representasi multi-skala ini memungkinkan model untuk menangkap informasi kontekstual dan mendeteksi objek secara akurat pada skala yang berbeda dalam citra. FPN dalam Mask R-CNN terdiri dari langkah-langkah berikut:

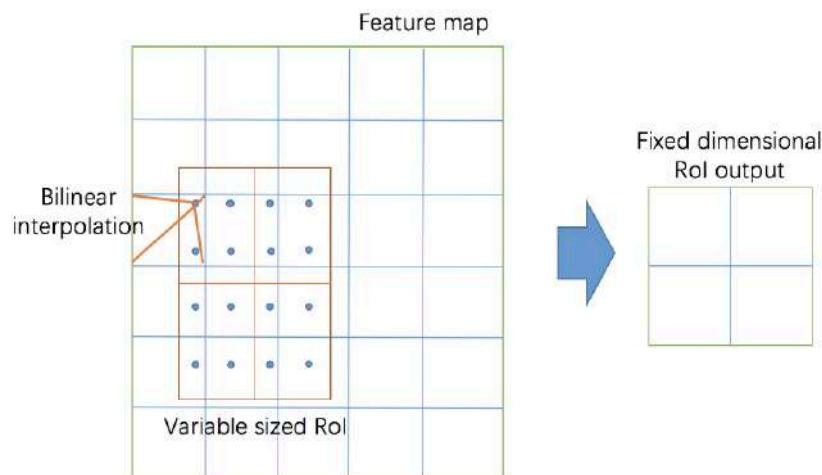
1. *Feature Extraction*: Jaringan utama mengekstrak fitur-fitur tingkat tinggi dari citra masukan.
2. *Feature Fusion*: FPN membuat koneksi antara tingkatan berbeda dari jaringan utama untuk membuat jalur dari atas ke bawah. Jalur ini menggabungkan informasi semantik tingkat tinggi dengan peta fitur tingkat rendah, memungkinkan model untuk menggunakan kembali fitur pada skala yang berbeda.
3. *Feature Pyramid*: Proses *fusion* menghasilkan piramida fitur multi-skala, di mana setiap tingkat piramida sesuai dengan resolusi fitur yang berbeda. Tingkat teratas piramida berisi fitur dengan resolusi tertinggi, sementara tingkat terbawah berisi fitur dengan resolusi terendah.

Selanjutnya adalah *Region Proposal Network* (RPN) yang bertanggung jawab untuk menghasilkan proposal wilayah atau kotak pembatas (*bounding box*) kandidat yang mungkin berisi objek dalam citra. Ini beroperasi pada peta fitur yang dihasilkan oleh jaringan utama dan mengusulkan wilayah potensial yang menarik. *Anchor boxes* yang memiliki *objectness score* tinggi kemudian digunakan sebagai proposal wilayah (*region proposals*) yang kemudian akan diproses lebih lanjut oleh model untuk deteksi objek. Berikut gambaran dari arsitektur RPN pada Gambar 2.16.



Gambar 2.16 *Region Proposal Network* (RPN) (Subash dkk., 2020).

Berikutnya adalah RoIAlign yang dimana setelah RPN menghasilkan proposal wilayah, lapisan ROIAlign (Region of Interest Align) diperkenalkan. Langkah ini membantu mengatasi masalah ketidakselarasan dalam ROI pooling. ROIAlign ini adalah "*pixel-to-pixel alignment*" yang memastikan hasil prediksi mask sesuai dengan objek dalam gambar tanpa adanya kuantisasi atau ketidakselarasan. Berikut gambaran dari proses ROIAlign dapat dilihat pada Gambar 2.17.



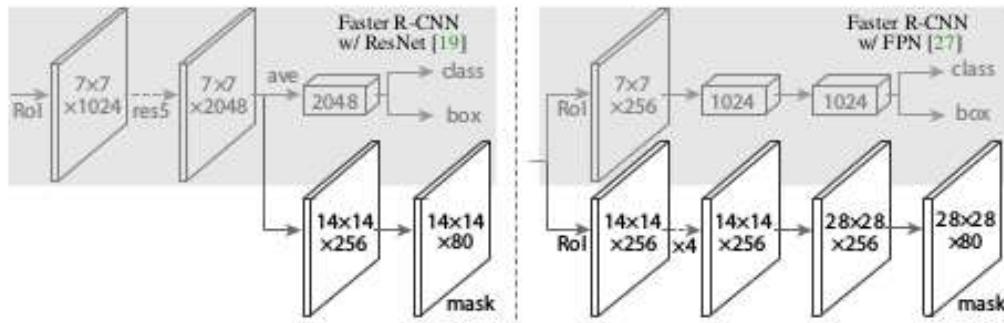
Gambar 2.17 Proses RoIAlign (Jiao dkk., 2019).

ROIAlign memainkan peran penting dalam mengekstrak fitur secara akurat dari peta fitur masukan untuk setiap proposal wilayah, memastikan segmentasi piksel yang tepat dalam tugas *instance segmentation*. Proses ROIAlign melibatkan langkah-langkah berikut:

1. *Input Feature Map*: Proses dimulai dengan *input* peta fitur, yang biasanya diperoleh dari jaringan utama. Peta fitur ini berisi informasi semantik tingkat tinggi tentang seluruh citra.
2. *Region Proposals*: *Region Proposal Network* (RPN) menghasilkan proposal wilayah (*candidate bounding boxes*) yang mungkin berisi *object of interest* dalam citra.
3. Pembagian Menjadi *Grid*: Setiap proposal wilayah dibagi menjadi sejumlah *bin* spasial yang sama atau *grid* berukuran tetap. Grid ini digunakan untuk mengekstrak fitur dari *input feature map* yang sesuai dengan *region of interest*.
4. Interpolasi Bilinear: Berbeda dengan *ROI pooling*, yang mengkuantisasi koordinat spasial *grid* ke *integer* terdekat, ROIAlign menggunakan interpolasi bilinear untuk menghitung kontribusi *pooling* untuk setiap *grid*. Interpolasi ini memastikan penyejajaran yang lebih tepat dari fitur dalam ROI.
5. *Output Features*: Fitur yang diperoleh dari *input feature map*, diselaraskan dengan setiap *grid* dalam peta fitur *output*, digunakan sebagai fitur representatif untuk setiap proposal wilayah. Fitur yang diselaraskan ini menangkap informasi spasial yang halus, yang penting untuk segmentasi yang akurat.

Penggunaan ROIAlign secara signifikan meningkatkan akurasi prediksi *mask* dalam *Mask R-CNN* (He dkk., 2017). Ini menghasilkan peningkatan sekitar 3 poin dalam *average precision* (AP) dan 5 poin dalam AP75 (AP pada ambang IoU 0.75) dibandingkan dengan RoIPool. ROIAlign juga tidak sensitif terhadap pilihan *max* atau *average pooling*, memberikan peningkatan kinerja yang konsisten.

Terakhir adalah *Mask Head*, *Mask Head* adalah cabang tambahan dalam Mask R-CNN, dapat dilihat pada Gambar 2.18. Bagian ini bertanggung jawab untuk menghasilkan *mask* segmentasi untuk setiap proposal wilayah. *Head* ini menggunakan fitur yang diselaraskan yang diperoleh melalui ROIAlign untuk memprediksi *binary mask* untuk setiap objek, menentukan batas piksel demi piksel dari *instance*/objek.



Gambar 2.18 Struktur *Mask Head* (He dkk., 2017).

Melalui kombinasi *object detection* dan *semantic segmentation* serta metode dua-langkah inovatif yang digunakan, *Mask R-CNN* telah membuktikan diri sebagai salah satu pendekatan paling efektif untuk mendeteksi objek dengan melakukan segmentasi lebih lanjut pada *computer vision*.

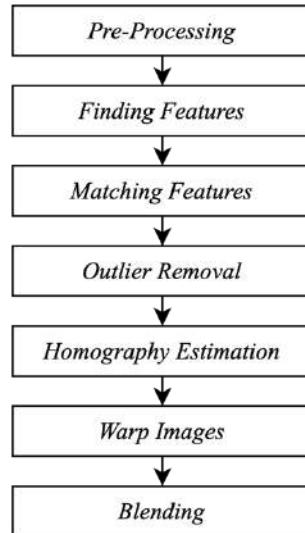
## 2.5. *Image Stitching*

Pada tahun-tahun terakhir ini, teknik *image stitching* telah berkembang pesat dan menjadi cabang penting dalam pengolahan citra digital dengan berbagai aplikasinya (Z. Wang & Yang, 2020). *Image stitching* memiliki peran vital dalam pemrosesan citra digital, membuatnya menjadi domain populer dalam pemetaan fotografi, visi komputer, pengolahan citra, dan grafika komputer (Z. Wang & Yang, 2020).

*Image stitching* atau juga dikenal sebagai penjahitan gambar adalah proses menggabungkan beberapa gambar fotografi yang memiliki bidang pandangan yang tumpang tindih (*overlap*) untuk menciptakan sebuah gambar tunggal yang mencakup bidang pandangan yang lebih luas atau panorama tersegmentasi. Tujuan dari *image stitching* adalah untuk menggabungkan daerah-daerah *overlap* tadi pada gambar-gambar tersebut secara mulus sehingga membentuk sebuah gambar akhir yang koheren dan beresolusi tinggi (Oktaviano dkk., 2021; Z. Wang & Yang, 2020).

Proses ini umumnya dilakukan menggunakan perangkat lunak komputer meskipun beberapa kamera digital juga sudah dilengkapi dengan kemampuan *image stitching* bawaan. Namun tidak semua kamera dapat menghasilkan gambar hasil jahit bernilai resolusi tinggi karena tergantung pada kemampuan masing-masing kamera (Oktaviano dkk., 2021).

Sebagian besar pendekatan *image stitching* membutuhkan *overlap* yang hampir persis antara gambar-gambar tersebut serta pengaturan eksposur yang identik untuk hasil optimal (Oktaviano dkk., 2021). Berikut tahapan proses *image stitching* ditampilkan pada Gambar 2.19.



Gambar 2.19 Tahapan proses *image stitching* (S. K. Sharma & Jain, 2020).

Dapat dilihat pada Gambar 2.14 bagaimana proses *image stitching* atau penggabungan gambar secara umum dilakukan. Berdasarkan Gambar 2.14, proses tersebut memiliki tujuh tahapan utama yakni *pre-processing*, *finding features*, *mathcing features*, *outlier removal*, *homography estimation*, *warp images* hingga *blending*. Berikut adalah penjelasan lebih detail terkait dengan tahapan proses *image stitching* berdasarkan Gambar 2.14, yakni:

### 1. *Pre-Processing*

Sebelum memproses gambar, langkah awal adalah mengubah resolusi gambar (reduksi) menjadi resolusi sedang (dan kemudian menjadi resolusi rendah). Hal ini bertujuan untuk mengurangi beban komputasi dan mempercepat proses. Tahapan ini juga sering disebut dengan tahap *pre-processing*, selain melakukan perubahan resolusi gambar bisa juga dilakukan perubahan warna ke dalam skala abu (*grayscale*) atau bahkan proses lainnya.

### 2. *Finding Features*

Selanjutnya adalah tahapan mendeteksi fitur-fitur pada gambar yang dimasukkan. Fitur-fitur ini digunakan untuk mencari kecocokan antara gambar-gambar. Dalam proses inilah algoritma seperti ORB, SIFT, BRISK dan AKAZE digunakan.

### 3. *Matching Features*

Setelah fitur-fitur dideteksi, proses selanjutnya adalah pencocokan fitur-fitur tersebut secara berpasangan. Tahapan ini membantu dalam menentukan bagaimana nanti gambar-gambar tersebut akan disusun.

#### 4. *Outlier Removal*

Setelah mencocokkan fitur-fitur, subset gambar-gambar yang relevan dibuat. Ini adalah langkah penting dalam memilih gambar mana yang akan digunakan (*inliers*) dan mengecualikan gambar yang tidak relevan (*outliers*) atau bisa dikatakan tidak memiliki fitur yang sama dengan gambar lain.

#### 5. *Find Homography*

Pada tahapan ini dilakukan penyesuaian pada kamera atau bisa dikatakan sudut pandang pada gambar-gambar berdasarkan *homography* sehingga meningkatkan keakuratan ketika proses penggabungan gambar. Proses ini termasuk estimasi parameter kamera, penyesuaian kamera dan juga koreksi distorsi kamera.

#### 6. *Warp Images*

Tahapan berikutnya adalah *warp images* atau transformasi gambar untuk menggabungkan gambar-gambar sesuai dengan fitur-fitur dan penyesuaian yang telah dilakukan. Ini mencakup seperti pergeseran gambar, rotasi gambar, distorsi gambar dan lainnya.

Selain itu pada bagian ini juga dilakukan *seam masking*. Seam atau dalam bahasa Indonesia adalah jahitan disini adalah menentukan garis transisi gambar-gambar dengan sedikit gangguan/*interference* untuk proses penjahitan atau *stitching*. Mask ini diperoleh dari gambar yang sudah melewati proses *warping*.

#### 7. *Blending*

Setelah semua tahap sebelumnya diselesaikan, gambar-gambar akhirnya digabungkan menjadi satu gambar panorama utuh. Proses ini melibatkan penggunaan masker jahitan (*seam masks*) dan *exposure error compensation* untuk menghasilkan gambar panorama yang mulus.

*Image stitching* memiliki berbagai aplikasi di bidang *remote sensing*, *aerospace*, *virtual reality*, *medical imaging*, dan lain-lain (Z. Wang & Yang, 2020). Beberapa contoh penerapan image stitching antara lain *mosaicking* dokumen, stabilisasi gambar pada kamera video *camcorder*, pembuatan fotomosaik beresolusi tinggi pada peta digital dan citra satelit. Selain itu juga digunakan dalam bidang pencitraan medis seperti super-resolusi multi-gambar *imaging* serta *video*

*stitching* untuk menyatukan beberapa klip video menjadi satu kesatuan. Terakhir terdapat pula aplikasi *object insertion* yang memungkinkan penambahan objek ke dalam sebuah komposisi gambar yang sudah ada. Sehingga secara keseluruhan dapat dikatakan bahwa *image stitching* adalah teknik yang sangat berharga untuk menciptakan gambar panorama atau menggabungkan beberapa gambar guna menangkap bidang pandangan yang lebih luas.

### **2.5.2 Scale-Invariant Feature Transform (SIFT)**

*Scale-Invariant Feature Transform* (SIFT) adalah sebuah algoritma deteksi dan ekstraksi fitur yang digunakan dalam bidang *computer vision* dan *image processing*. Algoritma ini pertama kali diperkenalkan oleh David G. Lowe sebagai metode untuk mengekstraksi fitur yang khas dan invariant dari gambar sehingga dapat digunakan untuk melakukan tugas seperti pencocokan dan pengenalan gambar (Lowe, 2004).

Algoritma SIFT dapat didekomposisi menjadi empat tahapan utama: *feature detection*, *feature description*, *feature matching*, dan *outlier rejection*. Algoritma ini bekerja dengan cara pertama-tama mendeteksi titik-titik kunci (*keypoints*) dalam sebuah gambar menggunakan filter Difference of Gaussian (DoG) (Lowe, 2004) yang merupakan aproksimasi dari *Laplacian-of-Gaussian* (LoG). Fitur-fitur di deteksi dengan mencari maksima lokal menggunakan DoG pada berbagai skala gambar subjeknya (Tareen & Saleem, 2018).

*Keypoints* ini kemudian digunakan untuk menghasilkan deskriptor yang mencerminkan penampilan lokal dan orientasi gambar pada titik tersebut. *Keypoints* tersebut diubah menjadi lebih optimal menggunakan algoritma deteksi ekstrema ruang-skala. Deskriptor kemudian dihitung untuk setiap *keypoint* berdasarkan arah gradien citra lokal. Metode deskripsi mengambil sebuah area sekitar 16x16 piksel di sekitar setiap fitur yang telah terdeteksi kemudian membaginya menjadi sub-blok sehingga menghasilkan total nilai bin sebanyak 128 buah. Terakhir, deskriptor-deskriptor tersebut dicocokkan antar gambar menggunakan algoritma *nearest-neighbor* dan *outlier rejection* menggunakan algoritma RANSAC (Lowe, 2004; Tareen & Saleem, 2018).

Salah satu kelebihan utama algoritma SIFT adalah ketangguhannya dalam menghadapi perubahan rotasi gambar, skala, serta variasi affine terbatas namun sayangnya biaya komputasionalnya cukup tinggi. Fitur-fitur SIFT dapat diaplikasikan pada berbagai tugas seperti pengenalan objek, pencarian citra, serta navigasi robot (Lowe, 2004; Tareen & Saleem, 2018).

### **2.5.1 Oriented FAST and Rotated BRIEF (ORB)**

Pada tahun 2011, (Rublee dkk., 2011) memperkenalkan *Oriented FAST and Rotated BRIEF* (ORB) sebagai alternatif cepat dan efisien untuk deteksi dan deskripsi fitur lokal yang menggunakan metode SIFT. ORB sendiri merupakan gabungan dari metode deteksi *modified FAST* (*Features from Accelerated Segment Test*) dan deskripsi *direction-normalized BRIEF* (*Binary Robust Independent Elementary Features*) (Tareen & Saleem, 2018).

Metode deteksi fitur pada ORB dilakukan dengan menggunakan algoritma FAST yang telah dimodifikasi. Algoritma ini akan mendeteksi sudut-sudut penting dalam setiap lapisan piramida skala gambar, kemudian kualitas sudut-sudut tersebut dinilai menggunakan skor Harris Corner untuk menyeleksi sudut-sudut berkualitas terbaik. Algoritma FAST sendiri tidak menghitung orientasi dan bersifat variatif terhadap rotasi. Oleh karena itu, dilakukan perhitungan pusat berbobot intensitas dari *patch* dengan sudut yang terletak di tengahnya untuk mendapatkan orientasinya. Selain itu, momen-momen juga dihitung untuk meningkatkan invariansi rotasi. (Setiawan dkk., 2020; Tareen & Saleem, 2018).

Deskripsi fitur pada ORB menggunakan versi modifikasi dari deskriptor BRIEF. Deskriptor ini memiliki ketidakstabilan saat terjadi rotasi, oleh karena itu digunakanlah versi modifikasi dari deskriptor BRIEF agar lebih stabil terhadap rotasi yang dihasilkan dari orientasi *patch* untuk mengkomputasikan deskriptor BRIEF. Fitur-fitur ORB bersifat invarian terhadap skala, rotasi, dan perubahan affine yang terbatas (Setiawan dkk., 2020; Tareen & Saleem, 2018). ORB telah diterapkan dalam berbagai bidang, seperti pengenalan objek, penyatuan gambar (*image stitching*), dan rekonstruksi 3D (Dai & Wu, 2023; Rublee dkk., 2011).

### 2.5.3 Binary Robust Invariant Scalable Keypoints (BRISK)

*Binary Robust Invariant Scalable Keypoints* (BRISK) adalah sebuah metode untuk mengekstraksi fitur yang khas dan invarian dari gambar yang dapat digunakan untuk melakukan tugas seperti pencocokan (*matching*) dan pengenalan (*recognition*) gambar. Metode ini diperkenalkan oleh S. Leutenegger et al. pada tahun 2011 (Leutenegger dkk., 2011). BRISK didesain agar efisien secara komputasional namun tetap memiliki performa yang baik dalam hal pencocokan dan kehandalan terhadap perubahan sudut pandang (Leutenegger dkk., 2011).

Metode BRISK didasarkan pada detektor kunci FAST, yaitu algoritma deteksi sudut yang efisien secara komputasional dan banyak digunakan dalam *computer vision*. Algoritma ini menggunakan piramida skala-ruang untuk mendeteksi titik-titik kunci (*keypoints*) dengan berbagai

skala dan orientasi, sehingga membuatnya tangguh terhadap perubahan skala dan rotasi (Leutenegger dkk., 2011).

Deskripsi BRISK didasarkan pada identifikasi arah karakteristik setiap fitur guna mencapai invariansi rotasi. BRISK menggunakan deskriptor biner yang *invariant* terhadap rotasi serta tahan terhadap *noise*, sehingga cocok digunakan pada aplikasi real-time dan kondisi yang menantang. Deskriptor tersebut dihitung dengan menggunakan pola perbandingan intensitas antara pasangan piksel dalam lingkungan lokal *keypoint* (Leutenegger dkk., 2011).

Untuk mengatasi invariansi terhadap pencahayaan, hasil uji kecerahan sederhana juga digabungkan dan deskriptor dibangun sebagai rangkaian biner. Fitur-fitur BRISK *invariant* terhadap skala, rotasi, serta perubahan *affine* yang terbatas (Tareen & Saleem, 2018). BRISK telah menemukan aplikasinya dalam berbagai bidang seperti pengenalan objek, penyusunan citra, dan rekonstruksi 3D (Leutenegger dkk., 2011).

#### **2.5.4 Accelerated-KAZE (AKAZE)**

*Accelerated-KAZE* (AKAZE) adalah sebuah algoritma yang diperkenalkan oleh P. F. Alcantarilla pada tahun 2013 sebagai alternatif yang lebih cepat dan efisien dibandingkan dengan SIFT dan SURF (Alcantarilla dkk., 2013). AKAZE merupakan perluasan dari metode KAZE, yang juga menggunakan filtrasi difusi non-linier seperti KAZE tetapi menggunakan skala-ruang non-linier yang dikonstruksi dengan menggunakan kerangka kerja *Fast Explicit Diffusion* (FED) yang efisien secara komputasional (Tareen & Saleem, 2018).

Detektor AKAZE didasarkan pada determinan Matriks Hessian (Tareen & Saleem, 2018). Deskriptor AKAZE didasarkan pada algoritma Modified Local Difference Binary (MLDB) yang juga sangat efisien (Tareen & Saleem, 2018). Deteksi dan deskripsi fitur baru yang dirancang agar lebih cepat dan efisien daripada SIFT dan SURF namun tetap memiliki performa baik dalam hal pencocokan serta kehandalan terhadap perubahan sudut pandang (Tareen & Saleem, 2018). Kualitas invariansi rotasi ditingkatkan dengan menggunakan filter Scharr. Maksima dari respons detektor pada lokasi spasial diambil sebagai titik-titik fitur. Fitur-fitur AKAZE invariant terhadap skala, rotasi, affine yang terbatas dan memiliki kekhasan yang lebih jelas pada berbagai skala karena pen(Tareen & Saleem, 2018)-linier (Tareen & Saleem, 2018).

AKAZE khususnya cocok untuk *real-time application* dan bekerja dengan baik dalam kondisi tantangan seperti pencahayaan rendah, *motion blur*, dan *occlusion*. AKAZE telah diterapkan dalam

berbagai bidang seperti *object detection*, *image stitching*, dan rekonstruksi 3D (Tareen & Saleem, 2018).

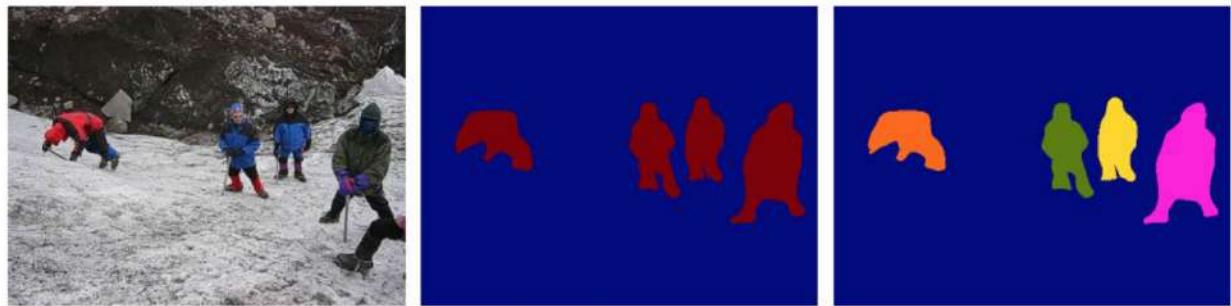
## 2.6. Image Segmentation

Menurut (Kaur & Kaur, 2014) algoritma *image segmentation* mengkategorikan piksel-piksel pada gambar satu per satu. Tujuannya adalah untuk mendapatkan transformasi dari gambar input di mana piksel latar belakang diberi label dengan cara tertentu dan objek-objek yang menarik diberi label lainnya, sehingga dapat diketahui batas setiap objek secara detail. Hasil keluaran dari proses ini juga berupa matriks (gambar *mask*) dengan nilai tunggal per piksel, yang berisi kategori yang telah ditugaskan kepada setiap piksel (Yagüe dkk., 2022).

Teknik *image segmentation* sangat berguna dalam aplikasi seperti kompresi gambar atau *object detection*, karena untuk jenis aplikasi tersebut tidak efisien jika seluruh gambar diproses. Oleh karena itu, *image segmentation* digunakan untuk memisahkan bagian-bagian pada gambar agar dapat diproses lebih lanjut (Kaur & Kaur, 2014).

Selain itu, *image segmentation* juga merujuk sebagai salah satu proses paling penting dalam *image processing*. Segmentasi bertujuan utama untuk menyederhanakan representasi suatu citra agar mudah dan dapat dianalisis dengan baik (Kaur & Kaur, 2014). *Image segmentation* merupakan langkah pertama yang penting dalam proses analisis citra. Hal ini karena segmentasi memainkan peran vital dalam tahap analisis citra untuk mengekstrak data dari gambar yang diberikan (Gurusamy dkk., 2014). Pemrosesan citra digital memiliki banyak aplikasi terbaru di bidang penginderaan jauh, kedokteran, fotografi, produksi film dan video, serta pemantauan keamanan (Gurusamy dkk., 2014).

Terdapat berbagai algoritma *image segmentation* yang telah dikembangkan dalam literatur. Mulai dari metode-metode awal seperti *thresholding*, *histogram-based bundling*, *region growing*, *k-means clustering*, hingga algoritma-algoritma yang lebih canggih seperti kontur aktif (*active contours*), potongan grafik (*graph cuts*), kondisional dan *Markov random fields*, serta metode berbasis kelimpahan (*sparsity based methods*) (Minaee dkk., 2020). Namun demikian, beberapa tahun terakhir ini model *deep learning* (DL) telah menghasilkan generasi baru model segmentasi gambar dengan peningkatan kinerja yang luar biasa dan sering kali mencapai tingkat akurasi tertinggi pada *benchmark*-populer sehingga menyebabkan perubahan paradigma di bidang ini (Kaur & Kaur, 2014).



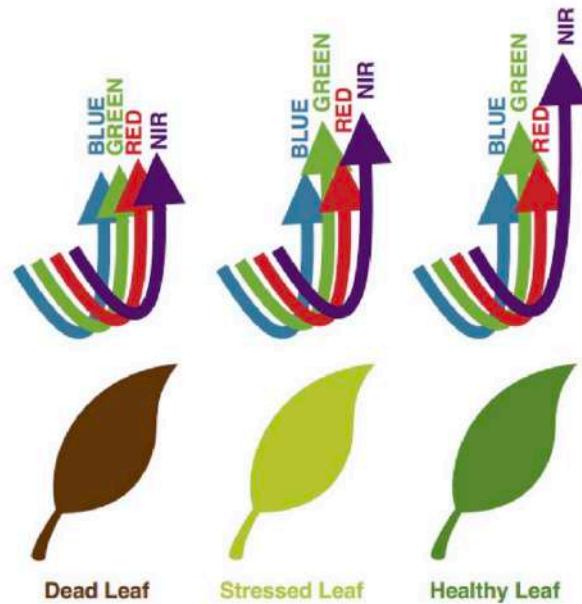
Gambar 2.20 Contoh hasil *semantic* dan *instance segmentation* (Minaee dkk., 2020).

*Image segmentation* dapat diformulasikan sebagai masalah klasifikasi piksel-piksel dengan label semantik (*semantic segmentation*) atau partisi objek individu (*instance segmentation*) (Minaee dkk., 2020). *Semantic segmentation* melakukan penandaan pada tingkat piksel dengan sekumpulan kategori objek (misalnya manusia, mobil, pohon, langit) untuk semua piksel gambar. Oleh karena itu, *semantic segmentation* umumnya lebih sulit daripada klasifikasi gambar yang memprediksi satu label untuk seluruh gambar. Sementara itu, *instance segmentation* meluaskan cakupan *semantic segmentation* dengan mendeteksi dan menggambarkan setiap objek yang menarik dalam gambar secara terpisah (misalnya partisi individu pada orang-orang (Minaee dkk., 2020).

## 2.7. Vegetation Index

*Vegetation index* (VI) atau indeks vegetasi merupakan representasi gabungan atau proses penyaringan dari beberapa set data spektral untuk menciptakan satu nilai tunggal untuk setiap titik dalam sebuah gambar demi menyoroti karakteristik vegetasi tertentu (Costa dkk., 2020; Kazemi & Ghanbari Parmehr, 2023). *Vegetation index* dihitung dengan menggunakan operator dan transformasi matematika pada reflektansi spektral di berbagai rentang spektral yang berbeda, termasuk cahaya terlihat (*visible bands*), inframerah dekat (*near-infrared/NIR band*), tengah (*middle band*), dan termal (*thermal band*) (Kazemi & Ghanbari Parmehr, 2023) dan digunakan untuk mengembangkan peta warna skala atau *scale color map* (Costa dkk., 2020). Dengan demikian, dengan mengukur pantulan cahaya setelah berinteraksi dengan tanaman, kita dapat menentukan karakteristik produk berdasarkan nilai pantulan untuk panjang gelombang cahaya yang berbeda (Kazemi & Ghanbari Parmehr, 2023). Gambaran dari tingkat nilai pantulan

gelombang cahaya terpengaruh berdasarkan tingkat kesehatan dauh dapat dilihat pada Gambar 2.21.



Gambar 2.21 Tingkat RGB dan NIR terhadap kesehatan daun (Evangelides & Nobajas, 2020).

Karakteristik spektral tanaman pada panjang gelombang yang berbeda dipengaruhi oleh klorofil, nitrogen, kandungan air tanaman, dan faktor morfologis (Kazemi & Ghanbari Parmehr, 2023). Ketika klorofil cukup dalam tanaman, pita *near-infrared* (NIR) adalah yang paling reflektif saat tanaman mengalami stres. Jika jumlah klorofil berkurang, perilaku spektral tanaman berubah dan menyebabkan penyerapan yang lebih sedikit pada pita biru dan merah. Oleh karena itu, pita merah dan biru juga tercermin bersama dengan pita hijau dan memberikan warna kuning atau merah pada vegetasi (Kazemi & Ghanbari Parmehr, 2023)

Dengan menggabungkan pita spektral, indeks vegetasi mampu efektif mewakili perubahan parameter fisiologis dan biokimia serta struktur tanaman (Yuan dkk., 2023). Sebagian besar *vegetation index* menggabungkan informasi yang terdapat dalam dua pita spektral, yaitu merah dan inframerah dekat (NIR) (Corresp dkk., 2018). Sejumlah indeks vegetasi telah dikembangkan dari data yang dikumpulkan menggunakan UAV/*Drone* (atau platform pemantauan jarak jauh lainnya) untuk fenotipe tanaman, termasuk indeks untuk menilai status kesehatan tanaman, status air tanaman, stres air, deteksi penyakit dan hama tanaman (Costa dkk., 2020; Yuan dkk., 2023).

Salah satu indeks vegetasi yang paling umum digunakan dan dipelajari adalah *Normalized Difference Vegetation Index* (NDVI). NDVI adalah salah satu teknik umum yang menggunakan *visible bands* dan NIR untuk menganalisis citra pemantauan jarak jauh dan menilai vegetasi hijau hidup (Corresp dkk., 2018). Awalnya, NDVI dikembangkan untuk menghasilkan korelasi yang baik antara nilai NDVI dan data vegetasi padang rumput (misalnya, biomassa kering dan hijau). NDVI telah menjadi yang paling dipelajari dan digunakan secara luas terutama dalam pengukuran vegetasi dan pertanian karena kemampuannya yang luar biasa untuk mendeteksi vegetasi (Costa dkk., 2020; Kazemi & Ghanbari Parmehr, 2023). Keberhasilan NDVI dalam pemantauan vegetasi disebabkan oleh ketersediaan saluran NIR dan saluran merah, karena reflektansi vegetasi yang hijau dan muda meningkat dalam saluran NIR dan menurun dalam saluran merah. Oleh karena itu, diperlukan sensor pemantauan multispektral dengan saluran NIR untuk aplikasi pertanian (Kazemi & Ghanbari Parmehr, 2023). Berikut formula yang digunakan untuk NDVI:

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Namun, sensor multispektral merupakan alat yang relatif mahal dan memerlukan lebih banyak pemrosesan dan kalibrasi yang memakan waktu. Alternatif dari sensor multispektral adalah pemanfaatan kamera RGB konvensional dengan resolusi tinggi yang lebih terjangkau. Berangkat dari situ, beberapa indeks tanaman mulai dikembangkan berapa tahun terakhir dengan hanya menggunakan saluran RGB konvensional saja untuk menghasilkan peta yang mirip dengan NDVI dan meminimalkan biaya pengumpulan data seperti *Visible Atmospheric Resistant Index* (VARI) dan *Green Leaf Index* (GLI) (Costa dkk., 2020; Eng dkk., 2019; Kazemi & Ghanbari Parmehr, 2023).

Beberapa penelitian telah dilakukan untuk menggunakan gambar UAV RGB daripada gambar multispektral dalam berbagai aplikasi pertanian presisi, termasuk identifikasi hama, identifikasi tanaman, dan estimasi hasil. Indeks vegetasi dengan RGB yang berasal dari gambar UAV menunjukkan kemampuan potensial dalam memprediksi pertumbuhan tanaman dan dapat digunakan sebagai indeks vegetasi jika gambar multispektral tidak tersedia (Kazemi & Ghanbari Parmehr, 2023; Saberioon dkk., 2014; Sánchez-Sastre dkk., 2020).

*Visible Atmospheric Resistant Index* (VARI) dan Indeks Daun Hijau (GLI) telah digunakan untuk melakukan pemantauan vegetasi hanya dengan menggunakan data pita terlihat (RGB) (Eng

dkk., 2019). Kedua rumus ini digunakan karena pendekatan mereka dalam pemantauan vegetasi dan penggunaan data RGB saja (Eng dkk., 2019).

### **2.7.1 *Visible Atmospheric Resistant Index (VARI)***

*Visible Atmospheric Resistant Index (VARI)* dikembangkan untuk estimasi regional kondisi tanaman. VARI lebih sensitif terhadap fraksi vegetasi karena memperkenalkan pantulan biru (Corresp dkk., 2018). VARI digunakan untuk memperkirakan fraksi vegetasi dengan sensitivitas minimal terhadap efek atmosfer (Gitelson dkk., 2002). Berikut formula yang digunakan untuk VARI:

$$VARI = \frac{(Green - Red)}{Green + Red + Blue}$$

Penambahan data pita biru (*blue-band*) dalam rumus VARI adalah untuk meminimalkan efek atmosfer, ini dapat mengestimasi fraksi vegetasi dengan kesalahan kurang dari sepuluh persen (Gitelson dkk., 2002). Selain itu, VARI juga dapat mendeteksi perubahan akumulasi biomassa dan sensitif terhadap jumlah klorofil dalam daun. VARI juga dapat digunakan untuk menghitung indeks potensi kebakaran dengan menggunakan data dari spektrometer pemantauan resolusi sedang, untuk memantau LAI (*Leaf Area Index*), dan menggunakan indeks pohon yang terlihat dalam pita warna umum (RGB) (Eng dkk., 2019).

### **2.7.2 Green Leaf Index (GLI)**

Karena GLI peka terhadap daun hijau, itu lebih baik digunakan untuk menentukan klorofil daun padi. Dalam percobaan lain, sepuluh rumus indeks vegetasi digunakan untuk mendeteksi penyakit *pine wilt* pada pohon. Mereka menemukan bahwa GLI dapat mendeteksi pohon yang terinfeksi lebih awal daripada rumus vegetasi lainnya. Namun, mereka juga menemukan bahwa GLI memiliki pemodelan yang lebih baik daripada yang lain (Eng dkk., 2019). Berikut rumus yang digunakan untuk perhitungan *vegetation index* menggunakan GLI:

$$GLI = \frac{2Green - Red - Blue}{2Green + Red + Blue}$$

## **2.8. OpenCV**

OpenCV yang merupakan singkatan dari *Open Source Computer Vision* adalah sebuah *open source library* untuk proses analisis gambar dan video (Culjak dkk., 2012). *Library* ini pertama kali diperkenalkan lebih dari dua dekade yang lalu tepatnya pada tahun 1999 oleh Intel (Culjak dkk., 2012). Awalnya, OpenCV dikembangkan di Laboratorium Penelitian Intel dalam upaya untuk memajukan pendekatan dalam membangun aplikasi yang intensif menggunakan CPU (Gollapudi, 2019). Tujuan utama dari pengembangan OpenCV adalah untuk membuat infrastruktur *computer vision* tersedia secara *universal*. *Library* ini terdiri dari berbagai *tools*, *libraries*, and *modules* yang mendukung implementasi aplikasi *computer vision*. Salah satu kelebihan utamanya adalah dukungan yang kuat dari komunitas pengembang.

OpenCV telah menjadi salah satu *toolkit* yang paling banyak digunakan dengan adanya kemampuan untuk mengimplementasikan *real-world use cases* dalam penggunaan komersial. Selain itu, OpenCV juga mendukung bahasa pemrograman C/C++, Python, dan Java serta dapat digunakan untuk membangun aplikasi *computer vision* pada sistem operasi *desktop* maupun *mobile* seperti Windows, Linux, macOS, Android, dan iOS. Dengan fleksibilitasnya tersebut maka tidak heran jika OpenCV banyak digunakan dalam berbagai bidang *image stitching* hingga *Augmented Reality* (AR). Makin banyak pula pengguna OpenCV yang mengembangkan aplikasi di berbagai platform. Hal ini menunjukkan betapa pentingnya keberadaan OpenCV dalam dunia teknologi saat ini.

## 2.9. Detectron2

Detectron2 adalah *library* generasi terbaru dari Facebook AI Research (FAIR) yang menyediakan algoritma deteksi dan segmentasi terkini. *Library* ini merupakan pengembangan dari Detectron dan *Mask RCNN-benchmark*. Ia mendukung sejumlah proyek penelitian *computer vision* serta aplikasi produksi di Facebook (Wu dkk., 2019). *Library* ini memiliki keunggulan dalam modularitas, kecepatan, dan dokumentasinya yang lengkap. Selain itu, tersedia juga berbagai arsitektur dan *pre-trained model*. Dengan memanfaatkan *transfer learning*, yaitu mentransfer pengetahuan jaringan saraf dari satu domain ke domain lainnya, kita dapat menghemat waktu pelatihan dan biaya dengan menggunakan *pre-trained models* tersebut. Selain itu, detectron2 juga memungkinkan integrasi dengan model-model baru berkat sifat modularitas dan fleksibilitasnya (Merz dkk., 2023).

Detectron2 mampu mengimplementasikan berbagai *computer vision framework* salah satunya adalah *Mask R-CNN*. Selain itu, *library* ini juga menyediakan fungsi-fungsi bantu untuk membuat tugas-tugas seperti visualisasi dan pelatihan menjadi lebih mudah (Ploeger & Dasovic, 2021). Model *Mask R-CNN* yang disertakan dalam Detectron2 telah dilatih menggunakan dataset COCO (*Common Objects in Context*) yang mencakup kurang lebih 91 kategori objek, mulai dari manusia hingga kendaraan dan hewan. Proses pelatihan pada dataset COCO memungkinkan model ini memiliki kemampuan untuk mengidentifikasi dan menentukan batas objek dengan akurasi tinggi dalam berbagai konteks visual. Dengan menggunakan *pre-trained model Mask R-CNN* dari Detectron2, pengguna dapat dengan mudah melakukan deteksi objek dan segmentasi instansinya dalam berbagai jenis gambar dengan hasil yang dapat diandalkan. Dengan kemampuan implementasinya yang luas serta fitur-fitur unggulannya tersebut menjadikan Detectron2 sebagai salah satu *library* paling populer dalam bidang deteksi dan segmentasi pada visi komputer.

OpenCV telah menjadi salah satu *toolkit* yang paling banyak digunakan dengan adanya kemampuan untuk mengimplementasikan *real-world use cases* dalam penggunaan komersial. Selain itu, OpenCV juga mendukung bahasa pemrograman C/C++, Python, dan Java serta dapat digunakan untuk membangun aplikasi *computer vision* pada sistem operasi *desktop* maupun *mobile* seperti Windows, Linux, macOS, Android, dan iOS. Dengan fleksibilitasnya tersebut maka tidak heran jika OpenCV banyak digunakan dalam berbagai bidang *image stitching* hingga *Augmented Reality* (AR). Makin banyak pula pengguna OpenCV yang mengembangkan aplikasi

di berbagai platform. Hal ini menunjukkan betapa pentingnya keberadaan OpenCV dalam dunia teknologi saat ini.

## 2.9. Penelitian Terdahulu

Dalam penyusunan penelitian ini, penulis merujuk kepada beberapa penelitian sebelumnya yang telah dilakukan sebelumnya. Rincian mengenai penelitian-penelitian sebelumnya dapat ditemukan pada Tabel 2.1 di bawah ini:

Tabel 2.1 Penelitian Terdahulu.

No	Judul Penelitian	Peneliti	Hasil Penelitian
1	<i>Using UAV image data to monitor the effects of different nitrogen application rates on tea quality</i>	Danni Luo, Yuan Gao, Yu Wang, Yujie Shi, Sizhou Chen, Zhaotang Ding, Kai Fan (2022)	Hasil penelitian menunjukkan bahwa model berbasis data citra UAV ( <i>Unmanned Aerial Vehicle</i> ) dan algoritma <i>machine learning</i> dapat secara efektif mendeteksi komponen bio-kimia utama dari tanaman teh. Dengan memanfaatkan teknologi ini, para peneliti dapat mengidentifikasi dan memonitor secara akurat tingkat aplikasi nitrogen yang berbeda dan dampaknya pada kualitas teh. (Luo dkk., 2022).
2	<i>Using Unmanned Aerial Vehicle-Based Multispectral Image Data to Monitor the Growth of Intercropping Crops in Tea Plantation</i>	Yujie Shi, Yuan Gao, Yu Wang, Danni Luo, Sizhou Chen, Zhaotang Ding and Kai Fan (2022)	Hasil penelitian membuktikan bahwa penggunaan data citra multispektral UAV yang dikombinasikan dengan metode <i>machine learning</i> dapat secara efektif memantau status pertumbuhan tanaman tumpangsari di kebun teh. Dengan menerapkan teknologi ini, para peneliti berhasil memberikan panduan praktis yang berharga bagi kebun teh

No	Judul Penelitian	Peneliti	Hasil Penelitian
			dengan pendekatan yang sesuai dengan kondisi lokasi tertentu. Dengan demikian, teknologi ini memungkinkan pengelolaan kebun teh secara lebih terarah dan optimal, sehingga dapat meningkatkan manfaat ekologis dan lingkungan (Shi dkk., 2022).
3	<i>Design Autonomous Drone Control For Monitoring Tea Plantation Using Dynamic Programming and Kruskal Algorithm</i>	Andri Agustav Wirabudi, Rendy Munadi, Angga Rusdinar, Angga Rusdinar, Dong Ho Lee (2019)	Hasil penelitian menunjukkan bahwa ketinggian yang tepat untuk memonitor lahan kebun teh adalah 80 meter, Selain itu, <i>drone</i> juga mendapatkan gambar dengan kualitas yang baik dan efisiensi daya yang lebih tinggi. Dengan demikian, ketinggian 80 meter sangat direkomendasikan untuk memonitor kebun teh karena tingkat akurasi citra mencapai 90% dan daya yang dibutuhkan relatif kecil. Penggunaan algoritma pemrograman dinamis dan Kruskal dalam pengendalian <i>drone</i> telah memberikan hasil yang efisien dan efektif untuk memantau kebun teh (Wirabudi dkk., 2019).
4	<i>Exploring the Potential of Unmanned Aerial Vehicle (UAV) Remote Sensing for Mapping Plucking Area of Tea Plantations</i>	Qingfan Zhang, Bo Wan, Zhenxiu Cao, Quanfa Zhang and Dezhi Wang (2021)	Penelitian ini mengembangkan pendekatan baru untuk memetakan area pemetikan ( <i>plucking area</i> ) di kebun teh menggunakan data terestrial yang diperoleh dari UAV ( <i>Unmanned Aerial Vehicle</i> ) seperti citra optik,

No	Judul Penelitian	Peneliti	Hasil Penelitian
			fotogrametri udara digital, dan data lidar. Empat model klasifikasi dirancang dengan menggunakan data yang berbeda dari UAV berdasarkan algoritma SVM ( <i>Support Vector Machine</i> ) dan RF ( <i>Random Forest</i> ). Hasil penelitian menunjukkan bahwa Model 4 adalah yang terbaik di antara model-model tersebut, mencapai akurasi sebesar 94,39% menggunakan algoritma RF dan 91,43% menggunakan algoritma SVM (Zhang dkk., 2021).
5	<i>Review of Feature Extraction and Matching Methods for Drone Image Stitching</i>	M. Dhana Lakshmi, P. Mirunalini, R. Priyadharsini and T. T. Mirnalinee (2019)	Dalam penelitian ini, digunakan algoritma SIFT ( <i>Scale-Invariant Feature Transform</i> ), SURF ( <i>Speeded-Up Robust Features</i> ), dan ORB ( <i>Oriented FAST and Rotated BRIEF</i> ) untuk ekstraksi fitur dari citra <i>drone</i> . Ketiga algoritma tersebut dapat efisien mendeteksi fitur-fitur dalam bentuk vektor deskriptor yang khas. Selain itu, ORB juga menghasilkan ekstraksi fitur dalam bentuk string biner. Setelah ekstraksi fitur, dilakukan pencocokan fitur menggunakan algoritma FLANN ( <i>Fast Library for Approximate Nearest Neighbors</i> ) dan BF ( <i>Brute-Force</i> ). Penelitian ini menunjukkan bahwa

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>pendekatan berbasis fitur dapat digunakan untuk menggabungkan citra drone dalam proses <i>image stitching</i>. Pendekatan ini membantu untuk mendapatkan pandangan penuh dari suatu adegan dengan menyatukan beberapa gambar menjadi satu citra yang lebih luas (Dhana Lakshmi dkk., 2019).</p>
6	<i>Image Registration Techniques and Applications: Comparative Study on Remote Sensing Imagery</i>	Mohamed Ihmeida, Hong Wei (2021)	<p>Penelitian ini melakukan perbandingan antara beberapa algoritma populer untuk registrasi citra, yaitu <i>Scale-invariant feature transform</i> (SIFT), <i>Speeded Up Robust Features</i> (SURF), <i>Oriented FAST and Rotated BRIEF</i> (ORB), KAZE, <i>Binary Robust Invariant Scalable Keypoints</i> (BRISK), dan <i>Accelerated-KAZE</i> (AKAZE) dalam berbagai skenario: rotasi (0 hingga 360 derajat), skala (25% hingga 600%), dan multitemporal. Citra-citra yang digunakan dalam eksperimen adalah citra Radar, citra Udara, dan citra <i>Unmanned Aerial Vehicle</i> (UAV). Untuk pencocokan fitur, digunakan metode <i>Nearest Neighbour Distance Ratio</i> (NNDR), sedangkan algoritma RANSAC digunakan untuk menolak pencocokan yang tidak tepat.</p>

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>Hasil eksperimen menunjukkan bahwa algoritma SIFT memberikan performa terbaik dibandingkan dengan algoritma lainnya, menunjukkan stabilitas yang kuat dan presisi yang tinggi dalam semua skenario. Sedangkan untuk aplikasi <i>real-time</i>, algoritma ORB memberikan hasil yang baik dan menjadi yang paling cepat di antara semua algoritma dalam berbagai skenario, diikuti oleh AKAZE sebagai algoritma tercepat kedua (Ihmeida &amp; Wei, 2021).</p>
7	<i>A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching</i>	Surendra Kumar Sharma, Kamal Jain and Anoop Kumar Shukla (2023)	<p>Penelitian ini melakukan perbandingan komprehensif terhadap lima belas teknik deteksi dan deskripsi fitur yang terkemuka untuk proses <i>image stitching</i> (penyatuan citra). Beberapa kontribusi dari penelitian ini adalah sebagai berikut:</p> <ul style="list-style-type: none"> <li>(1) Evaluasi memberikan gambaran yang lebih baik tentang bagaimana teknik deteksi dan deskripsi fitur bekerja pada berbagai dataset citra;</li> <li>(2) Perbandingan kumulatif dilakukan untuk keseluruhan 85 pasangan detektor-deskriptor pada berbagai dataset; dan</li> </ul>

No	Judul Penelitian	Peneliti	Hasil Penelitian
			(3) Setiap pasangan detektor dan deskriptor diberikan peringkat untuk setiap <i>dataset</i> , yang menunjukkan kinerja relatif dari detektor dan deskriptor untuk tipe <i>dataset</i> tertentu. Secara keseluruhan, hasil penelitian mengidentifikasi bahwa detektor AKAZE yang dipadukan dengan deskriptor AKAZE berhasil melampaui kombinasi lainnya untuk semua jenis dataset yang dipilih. Selain itu, detektor AGAST yang digabungkan dengan deskriptor BRIEF juga memberikan kinerja yang sangat baik untuk dataset 3 yang berisi adegan dalam ruangan (S. K. Sharma dkk., 2023).
8	<i>Mapping Tea Plantations from VHR Images Using OBIA and Convolutional Neural Networks</i>	Zixia Tang, Mengmeng Li and Xiaoqin Wang (2020)	Penelitian ini menyajikan metode <i>object-based CNN</i> untuk mengekstraksi kebun teh menggunakan citra penginderaan jauh (VHR) dengan resolusi tinggi. Secara lebih spesifik, penelitian ini menggunakan model CNN yang sudah disesuaikan ( <i>fine-tuned CNN</i> ) dalam pengaturan analisis citra berbasis objek ( <i>object-based image analysis</i> ) yang disebut sebagai <i>object-based CNN</i> . Model ini secara otomatis mengekstraksi fitur gambar tingkat tinggi untuk objek citra teh

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>yang didapatkan melalui segmentasi citra. <i>Random forest</i> kemudian digunakan untuk mengklasifikasikan objek citra. Hasil eksperimen menunjukkan bahwa metode <i>object-based</i> CNN yang diusulkan secara signifikan meningkatkan akurasi klasifikasi, sambil tetap menjaga batas-batas objek yang jelas. Akurasi klasifikasi dari metode yang sudah di-<i>fine-tuning</i> terbukti lebih tinggi dibandingkan metode tanpa <i>fine-tuning</i>, dan perbedaannya secara statistik signifikan. Selain itu, strategi <i>object-based</i> CNN yang diusulkan dan di-<i>fine-tuning</i> menggunakan citra Google Earth dapat diterapkan untuk klasifikasi kebun teh menggunakan citra GF-2 (Tang dkk., 2020).</p>
9	<i>Deep Learning Models for the Classification of Crops in Aerial Imagery: A Review</i>	Igor Teixeira, Raul Morais , Joaquim J. Sousa and António Cunha (2023)	<p>Penelitian ini melakukan tinjauan sistematis terhadap tiga puluh enam artikel dengan tujuan menjawab empat pertanyaan penelitian tentang penggunaan model <i>deep learning</i> untuk klasifikasi tanaman menggunakan data penginderaan jauh dari citra udara. Dari artikel-artikel yang ditinjau, ditemukan bahwa teknik <i>deep learning</i>, terutama yang berbasis pada jaringan saraf</p>

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>konvolusi (CNN) dan jaringan LSTM (<i>Long Short-Term Memory</i>), umumnya digunakan untuk klasifikasi tanaman dan cenderung mengungguli model pembelajaran mesin (<i>machine learning</i>) lainnya ketika data yang cukup tersedia. Pemilihan model yang tepat tergantung pada berbagai faktor seperti jenis data dan tanaman yang dianalisis, serta ukuran sampel yang tersedia.</p> <p>Para peneliti menganalisis data dari berbagai sistem pengambilan citra, termasuk satelit, UAV, dan pesawat terbang, dan menggunakan teknik seperti augmentasi data, transfer learning, dan <i>multimodal fusion</i> untuk meningkatkan akurasi klasifikasi. Faktor-faktor seperti resolusi spasial dan spektral, kualitas sampel, dan anotasi citra terbukti memiliki dampak signifikan pada akurasi klasifikasi tanaman (Teixeira dkk., 2023).</p>
10	<i>The Stress Detection and Segmentation Strategy in Tea Plant at Canopy Level</i>	Xiaohu Zhao, Jingcheng Zhang, Ailun Tang, Yifan Yu, Lijie Yan, Dongmei Chen and Lin Yuan (2022)	Penelitian ini bertujuan untuk mengembangkan strategi segmentasi cerdas yang menggunakan gabungan <i>deep learning</i> dan pengolahan citra untuk mendeteksi dan menghitung rasio kerusakan secara otomatis pada

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>gambar kanopi tanaman teh, khususnya untuk segmentasi penyakit <i>scab</i>. Metode yang diusulkan berhasil melakukan pengenalan otomatis dan membedakan berbagai jenis tekanan pada tanaman teh, serta mendapatkan batas yang tepat dari semua <i>scab stress</i> untuk mendapatkan rasio kerusakan yang akurat. Metode <i>Faster R-CNN</i> yang dikhkusukan dalam penelitian ini menggunakan kernel konvolusi yang dapat dideformasi dan sampling IoU yang seimbang untuk secara efektif mendeteksi tiga jenis tekanan tanaman teh yang khas, yaitu tungau daun teh hijau, <i>anthracnose</i>, dan luka bakar akibat sinar matahari. Performa dari metode <i>Faster R-CNN</i> yang dikhkusukan ini (<math>mAP = 76,07\%</math>) lebih baik dibandingkan dengan <i>YOLO v3</i> (<math>mAP = 65,89\%</math>) dalam skenario yang kompleks, seperti iluminasi, bayangan, <i>blur</i>, dan <i>occlusion</i> (Zhao dkk., 2022).</p>
11	<i>Tea picking point detection and location based on Mask-RCNN</i>	Tao Wang, Kunming Zhang, Wu Zhang, Ruiqing Wang, Shengmin Wan, Yuan Rao,	Penelitian ini bertujuan untuk mengidentifikasi, mendeteksi, dan segmentasi yang akurat dari kuncup dan daun teh, serta menentukan lokasi titik pemilihan teh dengan metode yang

No	Judul Penelitian	Peneliti	Hasil Penelitian
		Zhaohui Jiang, Lichuan Gu (2023)	<p>didasarkan pada jaringan saraf konvolusi berbasis wilayah (R-CNN) <i>Mask-RCNN</i>. Sebuah model pengenalan kuncup dan daun teh beserta titik pemilihan teh dibangun dalam penelitian ini. Pertama, gambar kuncup dan daun teh dikumpulkan dalam lingkungan yang kompleks, kemudian jaringan residual Resnet50 dan jaringan piramida fitur (FPN) digunakan untuk mengekstraksi fitur-fitur kuncup dan daun. Melalui jaringan proposal wilayah (RPN), dilakukan pelatihan klasifikasi dan regresi kotak pra-pemilihan pada peta fitur. Selanjutnya, metode agregasi fitur wilayah (RoIAlign) digunakan untuk menghilangkan kesalahan kuantisasi, dan peta fitur wilayah minat yang dipilih (ROI) dikonversi menjadi peta fitur berukuran tetap. Modul output dari model ini mampu melakukan fungsi klasifikasi, regresi, dan segmentasi. Posisi pemilihan kuncup dan daun teh ditentukan melalui gambar <i>mask output</i> dan algoritma penempatan.</p> <p>Seratus gambar kuncup dan daun pohon teh dalam lingkungan yang</p>

No	Judul Penelitian	Peneliti	Hasil Penelitian
			kompleks dipilih untuk pengujian. Hasil eksperimen menunjukkan bahwa rata-rata tingkat akurasi deteksi mencapai 93,95% dan tingkat <i>recall</i> mencapai 92,48%. Metode lokasi pemilihan teh yang diusulkan dalam penelitian ini terbukti lebih serbaguna dan tangguh dalam lingkungan yang kompleks (T. Wang dkk., 2023).
12	<i>Symptom recognition of disease and insect damage based on Mask R-CNN, wavelet transform, and F-RNet</i>	He Li, Hongtao Shi, Anghong Du, Yilin Mao, Kai Fan, Yu Wang, Yaozong Shen, Shuangshuang Wang, Xiuxiu Xu, Lili Tian, Hui Wang and Zhaotang Ding (2022)	Dalam penelitian ini, diusulkan sebuah kerangka kerja pengenalan gejala penyakit dan kerusakan serangga pada gambar daun teh berbasis <i>Mask R-CNN</i> , transformasi <i>wavelet</i> , dan <i>F-RNet</i> . Pertama, model <i>Mask R-CNN</i> digunakan untuk melakukan segmentasi bintik penyakit dan bintik serangga dari daun teh. Selanjutnya, transformasi gelombang diskrit dua dimensi digunakan untuk meningkatkan fitur-fitur gambar bintik penyakit dan bintik serangga, sehingga menghasilkan gambar dengan empat frekuensi. Terakhir, gambar dengan empat frekuensi tersebut secara bersamaan dimasukkan ke dalam jaringan residual berempat saluran ( <i>F-RNet</i> ) untuk mengidentifikasi gejala penyakit dan kerusakan serangga. Hasil

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>penelitian menunjukkan bahwa model <i>Mask R-CNN</i> mampu mendeteksi 98,7% bintik penyakit dan serangga, yang menjamin hampir semua bintik penyakit dan serangga dapat diambil dari daun teh. Akurasi model F-RNet adalah 88%, yang lebih tinggi daripada model lain (misalnya, SVM, AlexNet, VGG16, dan ResNet18). Oleh karena itu, kerangka kerja eksperimental ini dapat dengan akurat melakukan segmentasi dan identifikasi bintik penyakit dan serangga pada daun teh, yang tidak hanya memiliki signifikansi besar dalam identifikasi akurat penyakit tanaman teh dan hama serangga, tetapi juga memiliki nilai besar untuk penggunaan lebih lanjut dalam menggunakan kecerdasan buatan untuk mengendalikan penyakit tanaman teh dan hama serangga secara komprehensif (H. Li dkk., 2022).</p>
13	<i>The Use of VARI, GLI, and VIgreen Formulas in Detecting Vegetation In aerial Images</i>	Lim Soon Eng, Rozita Ismail, Wahidah Hashim, Aslina Baharum (2019)	<p>Penelitian melakukan perbandingan terhadap tiga metode perhitungan <i>vegetation index</i> yang hanya menggunakan RGB terhadap gambar yang diambil oleh <i>drone</i>. Ketiga metode itu adalah <i>Visible atmospherically resistant index</i></p>

No	Judul Penelitian	Peneliti	Hasil Penelitian
			<p>(VARI), <i>green leaf index</i> (GLI) dan <i>visible atmospherically resistant indices green</i> (VIgreen).</p> <p>Selama percobaan, ditemukan bahwa formula GLI terbukti merupakan teknik yang paling cocok yang paling cocok untuk menyoroti vegetasi di daerah perkotaan dan hutan berdasarkan sensitivitasnya dalam mendeteksi perbedaan antara vegetasi dan non-vegetasi (Eng dkk., 2019).</p>

Melalui tinjauan beberapa penelitian sebelumnya yang terdokumentasi dalam Tabel 2.1 di atas, penelitian sebelumnya mengenai pemanfaatan citra *drone* untuk *monitoring* perkebunan teh memberikan landasan penting bagi penelitian ini. Dua penelitian yang signifikan, yaitu oleh Luo et al. (2022) dan Shi et al. (2022), menunjukkan bahwa teknologi *Unmanned Aerial Vehicle* (UAV) mampu secara efektif mendeteksi komponen bio-kimia tanaman teh dan memonitor pertumbuhan tanaman tumpangsari. Dengan teknologi ini, peneliti dapat mengidentifikasi perbedaan tingkat aplikasi nitrogen dan dampaknya pada kualitas teh.

Pentingnya pemetaan area perkebunan teh dari citra *drone* ditekankan oleh penelitian Dhana Lakshmi et al. (2019), yang menggunakan algoritma *image stitching* untuk menggabungkan citra *drone*. Hasil penelitian ini mendukung pendekatan menyeluruh dalam pemetaan, memungkinkan analisis holistik terhadap kondisi tanaman dan lingkungan. *Image stitching* memberikan solusi untuk menciptakan representasi menyeluruh dari area perkebunan teh.

Dua penelitian lainnya, yaitu Wang et al. (2023) dan Zhao et al. (2022), menyoroti pentingnya pemetaan objek tanaman teh melalui teknik *image segmentation*, terutama dengan menggunakan *Mask Region-Based Convolutional Neural Network* (Mask R-CNN). Pemetaan ini memungkinkan analisis yang lebih fokus dan mendetail terhadap kesiapan petik, dengan memisahkan objek tanaman teh dari latar belakang.

Selanjutnya, Eng et al. (2019) menawarkan metode perhitungan *Vegetation Index* (VI), terutama *Green Leaf Index* (GLI), untuk mengevaluasi tingkat vegetasi pada tanaman teh. Penelitian ini memberikan dasar untuk pendekatan Anda dalam menggunakan GLI untuk memperhitungkan tingkat vegetasi pada tanaman teh dalam konteks kesiapan petik.

Selain itu, peneliti telah melakukan kajian pustaka tersendiri terkait dengan metode *vegetation index* yang hanya memanfaatkan citra RGB saja. Kemudian berdasarkan kajian tersebut ditemukan sebanyak 11 penelitian yang berkaitan dengan hal tersebut. Berdasarkan 11 penelitian tersebut, dilakukan pendataan terhadap metode apa saja yang digunakan. Berdasarkan pendataan didapatkan dua metode teratas yang digunakan dalam 8 penelitian yakni metode *Visible Atmospherically Resistant Index* (VARI) dan *Green Leaf Index* (GLI). Berikut ditampilkan data metode yang paling banyak digunakan serta penelitian yang membahas metode tersebut pada Tabel 2.2.

Tabel 2.2 Daftar metode *vegetation index* yang sering digunakan pada penelitian

No	Metode	Formula	Objek	Penelitian
1	<i>Visible Atmospherically Resistant Index</i> (VARI)	$VARI = \frac{(Green-Red)}{Green+Red-Blue}$	Vegetasi di daerah perkotaan dan hutan	(Eng dkk., 2019)
			jeruk, anggur, dan tebu	(Costa dkk., 2020)
			Vegetasi di daerah pinggiran air	(Cermakova dkk., 2019)
			Kandungan Klorofil pada Daun Bit Gula	(Sánchez-Sastre dkk., 2020)
			<i>Seagrass</i>	(Corresp dkk., 2018)
			Pohon Palem	(Nur Anisa dkk., 2020)

No	Metode	Formula	Objek	Penelitian
			Memantau <i>Thesa sinensis</i> Walker di Perkebunan	(Yuan dkk., 2023)
2	<i>Green Leaf Index</i> (GLI)	$GLI = \frac{2Green - Red - Blue}{2Green + Red + Blue}$	Vegetasi di daerah perkotaan dan hutan	(Eng dkk., 2019)
			Vegetasi di daerah pinggiran air	(Cermakova dkk., 2019)
			Pertumbuhan tanaman Padi	(Kazemi & Ghanbari Parmehr, 2023)
			Kandungan Klorofil pada Daun Bit Gula	(Sánchez-Sastre dkk., 2020)

Selanjutnya berdasarkan kedua metode tersebut, dilakukan eksperimen sederhana untuk menentukan metode mana yang terbaik dalam kasus tanaman teh. Dilakukanlah perbandingan antara metode *Green Leaf Index* (GLI) dan *Visible Atmospherically Resistant Index* (VARI) untuk menentukan metode yang paling cocok digunakan dalam perhitungan indeks vegetasi untuk tanaman teh. Untuk mengevaluasi kedua metode, dilakukanlah perhitungan pada piksel gambar yang mewakili warna hijau dalam gambar tanaman teh. Digunakan kriteria warna hijau dengan threshold sebesar 0.6 dari skala 0-1, di mana warna hijau dianggap dimulai dari nilai 0.6 hingga 1. Hasil dari perbandingan dapat dilihat pada Gambar 2.22.

```
● (opstitch) muramasy@Muhammads-MacBook-Air
    ndvi.py
    GLI
    Processing image with shape 1080 x 1920
    0.2800000011920929 0.4000000059604645
    4491296
● (opstitch) muramasy@Muhammads-MacBook-Air
    ndvi.py
    VARI
    Processing image with shape 1080 x 1920
    0.03999999910593033 0.1599999964237213
    4461869
```

Gambar 2.22 Perbandingan jumlah piksel yang dianggap hijau antara hasil GLI dan VARI.

Diperlihatkan pada Gambar 2.22, hasil perhitungan piksel menggunakan metode GLI dan VARI pada gambar yang sama menunjukkan bahwa GLI dapat mendeteksi atau memperhitungkan lebih banyak warna hijau atau piksel hijau. Secara kuantitatif, GLI mampu mengidentifikasi sebanyak 4,491,296 piksel hijau. Sementara VARI hanya mengidentifikasi 4,461,869 piksel hijau. Berdasarkan eksperimen ini, kami menyimpulkan bahwa GLI lebih efektif dalam mengidentifikasi piksel berwarna hijau dalam gambar tanaman teh dibandingkan dengan VARI. Oleh karena itu, GLI dapat dianggap sebagai metode yang lebih baik untuk digunakan dalam perhitungan indeks vegetasi untuk tanaman teh.

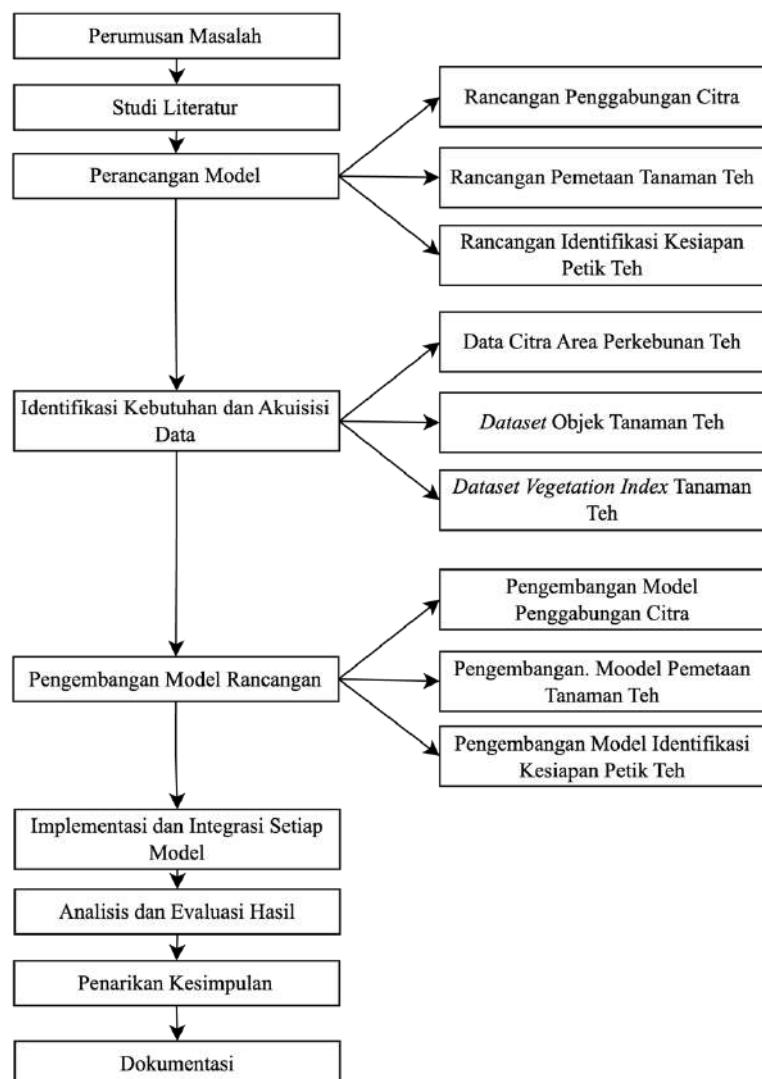
Dengan merangkum temuan dari penelitian-penelitian ini, skripsi ini berdiri di atas dasar pengetahuan yang solid dalam pemanfaatan citra *drone* untuk pemetaan perkebunan teh. Dukungan ini mencakup pemetaan area, pemetaan objek tanaman, dan evaluasi tingkat vegetasi, semua itu merujuk pada kesiapan petik.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Desain Penelitian

Desain penelitian yang diterapkan adalah metode *Research and Development* (R&D). Fokus utama dari metode *Research and Development* adalah pada pengembangan suatu produk atau inovasi. Prosesnya melibatkan serangkaian langkah untuk meningkatkan kualitas dan efektivitas produk tersebut dimulai dari penentuan masalah hingga perencanaan dan pengembangan (Mufadhol dkk., 2017). Tahapan-tahapan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain penelitian.

Berdasarkan desain penelitian pada Gambar 3.1 adapun penjelasan terkait tiap-tiap tahapan yang dilakukan adalah sebagai berikut:

### 1. Perumusan Masalah

Pada tahap permulaan penelitian, langkah yang terjadi pada tahap persiapan adalah memulai dengan mengidentifikasi permasalahan yang akan dibahas, tujuan penelitian, manfaat dari penelitian, serta batasan permasalahan. Tahapan selanjutnya adalah menentukan metode yang akan digunakan untuk menyelesaikan permasalahan tersebut, dan merencanakan desain penelitian yang akan membantu dalam penyelesaian permasalahan tersebut. Informasi yang lebih rinci mengenai proses perumusan permasalahan tersebut dapat ditemukan pada bagian Bab 1.

### 2. Studi Literatur

Langkah berikutnya adalah melakukan studi literatur terhadap materi penelitian. Studi literatur ini dilakukan dengan cara memeriksa dan memahami teori-teori yang relevan dengan topik utama penelitian ini untuk memberikan dasar dan dukungan pada penelitian. Studi literatur ini mencakup pemahaman tentang teh, monitoring perkebunan teh, teknologi *drone*, penggabungan gambar (*image stitching*), pemetaan tanaman teh, *image segmentation*, *Mask Region-based Convolutional Neural Network*, *vegetation index* dan *Green Leaf Index*.

### 3. Perancangan Model

Pada tahap ini, dilakukan penetapan kerangka kerja atau struktur umum yang akan digunakan untuk mengembangkan solusi. Ini mencakup keputusan strategis terkait dengan pendekatan umum yang akan diambil dalam mengatasi masalah penelitian. Adapun model yang dirancang antara lain yakni model penggabungan citra, model pemetaan tanaman teh dan model untuk identifikasi kesiapan petik teh.

#### a. Rancangan Penggabungan Citra

Pada tahap perancangan ini, dilakukan strategi untuk menggabungkan citra yang diperoleh dari *drone*. Penggabungan citra diperlukan agar dapat menciptakan gambaran yang komprehensif dan kontinu dari kebun teh menggunakan teknik *image stitching*.

#### b. Rancangan Pemetaan Tanaman Teh

Pada langkah ini, perancangan model pemetaan tanaman teh dilakukan untuk mengidentifikasi dan memetakan area tanaman teh dalam citra yang dihasilkan.

Penggunaan metode segmentasi objek, seperti yang diimplementasikan dengan *Mask Region-Based Convolutional Neural Networks* (*Mask R-CNN*), dapat memungkinkan pemetaan tanaman teh sehingga terpisah dari objek sekitarnya.

c. Rancangan Identifikasi Kesiapan Petik Teh

Rancangan ini berkaitan dengan pengembangan model untuk mengidentifikasi kesiapan petik dari tanaman teh. Melibatkan metode seperti analisis *vegetation index*, khususnya *Green Leaf Index* (GLI), dapat memberikan informasi tingkat vegetasi dari tanaman teh yang kemudian menjadi data untuk dibuat sebuah model dengan menggunakan *Sequential Model Neural Network* yang dapat mengidentifikasi tingkat kesiapan petik tanaman teh berdasarkan *vegetation index*.

#### 4. Identifikasi Kebutuhan dan Akuisisi Data

Pada tahapan ini bertujuan untuk mengidentifikasi kebutuhan data yang esensial untuk mendukung implementasi metodologi yang telah dirancang. Kebutuhan data mencakup informasi yang diperlukan untuk menjalankan proses *image stitching* guna menghasilkan gambar utuh perkebunan teh dari citra hasil tangkapan *drone*. Selain itu, data yang diperlukan untuk melakukan pemetaan area tanaman teh dengan menggunakan algoritma *Mask R-CNN* akan diidentifikasi secara rinci. Terakhir, data untuk perhitungan *Vegetation Index* dengan menggunakan GLI (*Green Leaf Index*) juga akan dicantumkan dalam identifikasi kebutuhan data. Pengumpulan data yang komprehensif ini menjadi landasan utama untuk memastikan bahwa seluruh proses analisis, pemetaan, dan prediksi kesiapan petik teh dapat dilakukan dengan akurat dan efektif.

#### 5. Pengembangan Model Rancangan

Pada tahap pengembangan, ide dan desain yang telah dirancang pada tahap sebelumnya diterapkan menjadi bentuk yang nyata. Ini adalah fase di mana model dan konsep-konsep abstrak diimplementasikan, dan prototipe atau produk awal dikembangkan dalam penelitian.

a. Pengembangan Model Penggabungan Citra

Pada tahap ini, model penggabungan citra yang telah dirancang akan dikembangkan lebih lanjut. Proses ini melibatkan implementasi algoritma penggabungan citra yang telah dipilih, dan kemungkinan penyesuaian parameter agar sesuai dengan karakteristik kebun teh yang bersangkutan. Penggabungan citra menggunakan teknik *image stitching*

sehingga pengembangan model dapat mencakup penentuan *overlap* antar citra, resolusi citra hasil, dan penanganan masalah-masalah seperti distorsi lensa. Pengujian dan evaluasi performa model penggabungan citra dapat dilakukan dengan membandingkan hasilnya dengan citra sebenarnya yang diambil dari kebun teh.

b. Pengembangan Model Pemetaan Tanaman Teh

Langkah ini memerlukan implementasi model pemetaan tanaman teh yang telah dirancang sebelumnya. Model ini dapat dikembangkan dengan memanfaatkan algoritma segmentasi objek, Mask R-CNN, untuk mengenali dan memetakan tanaman teh dalam citra *drone*. Proses pengembangan ini mungkin melibatkan pelatihan ulang model dengan menggunakan dataset yang sesuai dengan keadaan aktual kebun teh. Evaluasi performa model pemetaan dapat dilakukan dengan membandingkan hasil pemetaan dengan citra referensi atau data lapangan untuk memastikan akurasi dan ketelitian.

c. Pengembangan Model Identifikasi Kesiapan Petik Teh

Pengembangan model identifikasi kesiapan petik dari tanaman teh melibatkan implementasi metode *Green Leaf Index* (GLI) untuk mengukur tingkat vegetasi tanaman teh yang diterapkan pada citra *drone* yang telah diambil. Sehingga pada akhirnya dapat dibuat model yang dapat mengidentifikasi tingkat kesiapan petik tanaman teh berdasarkan *vegetation index value*. Evaluasi performa model identifikasi kesiapan petik dapat dilakukan dengan membandingkan hasil prediksi dengan pengamatan lapangan atau data acuan yang telah terverifikasi.

6. Implementasi dan Integrasi Setiap Model

Pada tahap implementasi, fokus akan diberikan pada integrasi model penggabungan citra, pemetaan tanaman teh, dan identifikasi kesiapan petik. Proses ini mengarah pada transformasi data tangkapan drone awal menjadi informasi yang menyeluruh dan informatif.

7. Analisis dan Evaluasi Hasil

Pada tahapan ini dilakukan analisis terhadap hasil pengolahan citra dan pemetaan yang telah dilakukan. Evaluasi dapat melibatkan perbandingan hasil pemetaan dengan citra referensi atau data lapangan. Analisis juga dilakukan untuk mengukur tingkat akurasi dan ketelitian dari model yang telah diimplementasikan. Kedua hal tersebut dilakukan terhadap masing-masing model baik sebelum maupun setelah integrasi termasuk didalamnya model

penggabungan citra, pemetaan tanaman teh dan juga pengidentifikasi kesiapan petik tanaman teh.

## 8. Penarikan Kesimpulan

Berdasarkan hasil analisis dan evaluasi, tarik kesimpulan terkait dengan keberhasilan implementasi model. Evaluasi dapat memberikan wawasan mengenai sejauh mana model dapat memberikan kontribusi dalam memonitor kebun teh dan menentukan kesiapan petik tanaman teh.

## 9. Dokumentasi

Dokumentasikan semua langkah-langkah yang dilakukan selama tahap implementasi dan evaluasi. Ini termasuk prosedur pengolahan citra, hasil analisis, dan kesimpulan yang ditarik. Dokumentasi yang rinci akan membantu untuk menyusun laporan akhir penelitian dan dapat menjadi referensi untuk penelitian berikutnya.

## 3.2 Alat dan Bahan Penelitian

Terdapat alat dan bahan penelitian sebagai pendukung proses penelitian. Berikut rincian dari alat dan bahan yang digunakan.

### 3.2.1 Alat Penelitian

Adapun alat yang digunakan dalam penelitian adalah sebuah perangkat komputer (laptop) dengan spesifikasi sebagai berikut:

1. Perangkat Keras (*Hardware*)
  - a. Apple M1 Processor
  - b. 8 GB RAM Memory
  - c. 250 GB Storage
2. Perangkat Lunak (*Software*)
  - a. MacOS Ventura ver. 13.4.1
  - b. Python
  - c. OpenCV
  - d. Keras, TensorFlow
  - e. Detectron2
  - f. Google Colab

g. *Code Editor*

### **3.2.2 Bahan Penelitian**

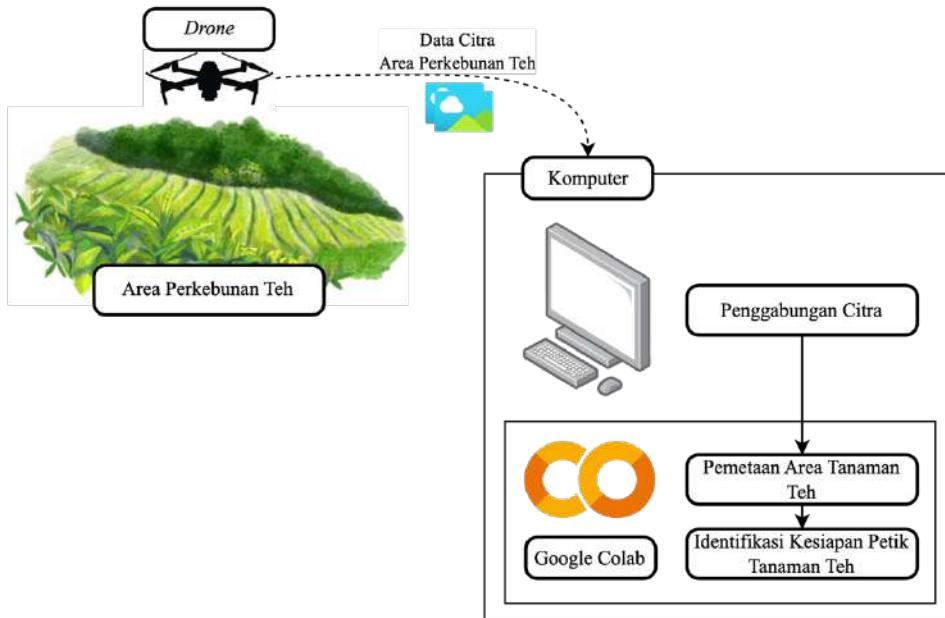
Bahan utama yang dibutuhkan dalam penelitian ini adalah data citra udara perkebunan teh yang diambil menggunakan *drone*. Data citra udara ini akan menjadi sumber utama untuk proses pemetaan dan identifikasi tingkat kesiapan tanaman teh. Data citra *drone* diambil secara langsung di Pusat Penelitian Teh dan Kina, Gambung, Ciwidey, Kabupaten Bandung terhadap dua area tanaman teh, yakni area tanaman yang siap petik dan area tanaman yang berstatus telah dipetik (tidak siap petik).

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Perancangan Model

Pada tahapan ini, dilakukan perancangan terhadam model-model untuk pemetaan dan identifikasi kesiapan petik tanaman teh. Adapun arsitektur atau gambaran besar dari sistem pemetaan dan identifikasi kesiapan petik tanaman teh diperlihatkan pada Gambar 4.1 berikut.



Gambar 4.1 Arsitektur sistem pemetaan dan identifikasi kesiapan petik tanaman teh.

Proses pertama yang dilakukan pada Gambar 4.1 adalah akuisisi data citra area perkebunan teh hasil tangkapan *drone*. *Drone* diterbangkan pada area perkebunan teh dan kemudian akan melakukan penangkapan citra secara berkelanjutan. Hasil tangkapan citra akan disimpan pada *memory card* selama *drone* terbang. Selanjutnya, setelah *drone* mendarat, data citra hasil tangkapan *drone* akan dipindahkan kedalam komputer untuk dilakukan pengolahan.

Pengolahan dilakukan menjadi tiga tahapan, yakni tahapan penggabungan citra menggunakan metode *image stitching* yang dilakukan pada komputer itu sendiri. Hasil dari penggabungan citra merupakan citra area perkebunan teh yang lebih luas. Kemudian data citra tersebut akan digunakan pada tahapan yang kedua yakni pemetaan area tanaman teh menggunakan metode *Mask R-CNN*. Citra diolah sehingga menghasilkan pemetaan area tanaman teh yang memisahkan antara objek tanaman teh dengan objek sekitarnya seperti pohon, tanah, batu dan lainnya. Pemisahan tersebut

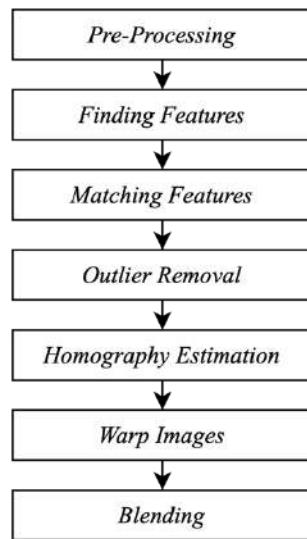
berupa adanya segmentasi pada objek tanaman teh sehingga menghasilkan *polygon* yang menjadi batasan pada area tanaman. Pengolahan dari pemetaan area tanaman teh akan dilakukan dalam *platform* Google Colab untuk *resource* yang lebih optimal terutama dalam pelatihan model nantinya. Pada akhirnya proses identifikasi kesiapan petik tanaman teh menggunakan metode GLI dilakukan terhadap area tanaman teh yang telah teridentifikasi. Dengan bantuan model *Sequential*, nantinya dibangun model yang dapat melakukan klasifikasi terhadap data GLI yang dihasilkan dari tanaman teh sehingga dapat mengidentifikasi kesiapan petik dari tanaman teh tersebut. Agar dapat tergambar lebih jelas tahapan yang dilakukan, berikut adalah diagram alir dari tahapan semua metode dalam penelitian ini ditampilkan pada Gambar 4.2



Gambar 4.2 Diagram alir tahapan penelitian dari awal hingga akhir.

#### 4.1.1 Perancangan Model Penggabungan Citra

Rancangan untuk model penggabungan citra dibuat berdasarkan tahapan proses dari teknik *image stitching* yang sebelumnya dijelaskan pada bagian kajian pustaka. Berikut rancangan model untuk proses penggabungan citra hasil tangkapan drone:



Gambar 4.3 Tahapan Proses *Image Stitching*.

Berdasarkan tahapan proses *image stitching* pada Gambar 4.3, secara garis besar, proses *image stitching* terbagi menjadi tujuh tahapan. Pertama adalah proses *pre-processing* terhadap citra yang akan digabung, Adapun proses yang dilakukan adalah *resizing image* atau perubahan resolusi citra. Kemudian selanjutnya tahapan ‘*Finding Features*’ akan dilakukan pendektsian fitur-fitur yang terdapat pada citra area perkebunan ter hasil tangkapan *drone*. Algoritma yang akan digunakan untuk deteksi dan ekstraksi fitur adalah *Scale-Invariant Feature Transform* (SIFT). Fitur-fitur tersebut akan berupa kumpulan *keypoints* yang kemudian akan dibandingkan antara fitur pada satu citra dengan citra yang lain. Pencocokan fitur dilakukan pada tahapan ‘*Matching Features*’ yang dimana setiap gambar satu sama lain dihitung nilai *confidence*. Jika makin banyak fitur yang serupa (*inliers*) antara dua gambar, maka akan semakin tinggi nilai *confidence* yang diperoleh. *Inliers* ini akan dihitung menggunakan metode *random sample consensus* (RANSAC).

Setelah melakukan proses ekstraksi dan pencocokan fitur, berdasarkan jumlah fitur kemudian jumlah *inliers* dan *confidence* dibuatlah *subset* yang berisikan citra-citra yang relevan untuk digabungkan. Selain itu juga pada proses ini gambar yang yang tidak memiliki hubungan atau tidak memiliki kecocokan fitur, *inliers* dan *confidece* yang baik akan dikecualikan dari *subset*

(*outlier removal*) yang akan dibuat. Setelah itu berdasarkan *inliers* dilakukanlah proses penyesuaian pada kamera atau bisa dikatakan sudut pandang pada citra-citra berdasarkan *homography (homography estimation)* sehingga meningkatkan keakuratan ketika proses penggabungan gambar.

Tahapan selanjutnya adalah *warp images* yang bertujuan untuk menyatukan gambar-gambar ke dalam satu bidang akhir. Hasil *warp* dari adalah sudut dan ukuran dari tiap gambar, sehingga dapat diketahui posisi akhir dari setiap gambar pada bidang akhir. Dalam proses itu dilakukan juga proses untuk penentuan garis transisi *seam masks* antar gambar yang akan digabungkan. Garis transisi ini akan digunakan pada proses *blending* untuk menentukan bagaimana nanti gambar harus disusun. Pada akhirnya dilakukan *blending* berdasarkan hasil *warp* sehingga menghasilkan sebuah gabungan gambar area perkebunan teh yang lebih luas.

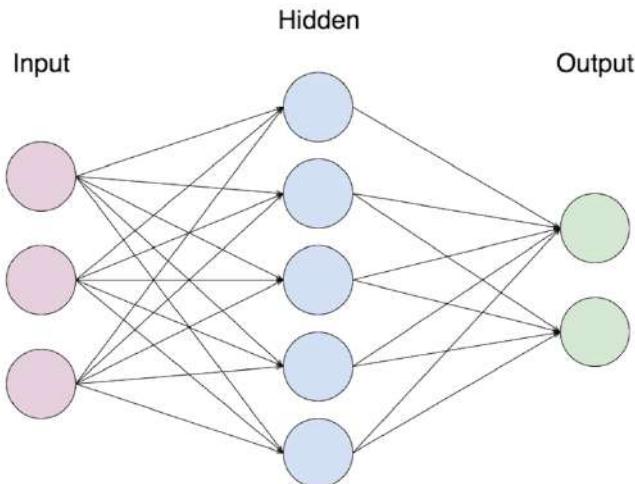
#### **4.1.2 Perancangan Model Pemetaan Tanaman Teh**

Rancangan untuk model pemetaan tanaman teh akan menggunakan metode *image segmentation* dengan *Mask Region-Based Convolutional Neural Networks (Mask R-CNN)*. Implementasi *Mask R-CNN* ini akan memanfaatkan *framework* yang bernama Detectron2. Tahapan yang dilakukan nantinya adalah dimulai dari persiapan *dataset* yang akan digunakan, dilanjutkan dengan melakukan proses konfigurasi dari model yang akan dilatih, hingga pada akhirnya dilakukan proses pelatihan model. Pembangunan dan pelatihan model nantinya akan memanfaatkan *framework* Detectron2 dan juga Google Colab *platform*. Model yang digunakan untuk dilakukan pelatihan adalah model yang telah disediakan oleh Detectron2, yakni *pre-trained model* ‘*mask\_rcnn\_R\_50\_FPN\_3x*’ dan ‘*mask\_rcnn\_R\_101\_FPN\_3x*’ lalu dilatih dengan beberapa iterasi untuk membandingkan model yang dihasilkan berdasarkan kombinasi tersebut. Pada akhirnya nanti akan dilakukan analisis terhadap hasil dari model yang dilatih berdasarkan proses pelatihan serta juga pengujian dengan *dataset uji*.

#### **4.1.3 Perancangan Model Identifikasi Kesiapan Petik Tanaman Teh**

Dalam rancangan model pendekripsi kesiapan petik tanaman teh, digunakan *Artificial Neural Network* dengan *Sequential Model*. Arsitektur model ini terdiri dari tiga lapisan utama yakni *input layer*, *hidden layer*, dan *output layer*. Penerapan model ini akan menggunakan TensorFlow dan nantinya model akan dilatih menggunakan *dataset* yang sudah disiapkan sebelumnya yang berisikan data *vegetation index (VI)* menggunakan metode *green leaf index (GLI)*. Adapun

gambaran arsitektur dari *Sequential Model Neural Network* yang digunakan ditampilkan pada Gambar 4.4.



Gambar 4.4 Arsitektur dari *Artificial Neural Network* dengan *Sequential Model*.

Pertama, lapisan *input* neuron yang banyaknya disesuaikan dengan dimensi fitur dari dataset yang berisi data *vegetation index* objek tanaman teh. Lapisan kedua adalah lapisan tersembunyi (*hidden layer*) dengan 16 neuron, menggunakan fungsi aktivasi ReLU untuk memperkenalkan non-linearitas dan memungkinkan pembelajaran fitur-fitur yang lebih kompleks. Lapisan terakhir adalah lapisan *output* dengan satu neuron dan fungsi aktivasi *sigmoid*, cocok untuk tugas *binary classification* dalam menentukan kesiapan petik tanaman teh. Model ini sengaja dirancang secara sederhana, sesuai untuk tugas *binary classification* yang fokus pada menentukan apakah tanaman teh siap petik atau tidak berdasarkan *vegetation index*. Setiap tahap arsitektur ini telah dirancang dengan pertimbangan untuk mengoptimalkan pembelajaran dari data kesiapan petik tanaman teh. Dengan *Neural Network Model Sequential* ini, diharapkan dapat memberikan hasil klasifikasi yang akurat dan mendukung penelitian mengenai pendekripsi kesiapan petik pada tanaman teh. Pelatihan dan pengembangan model nantinya akan memanfaatkan Keras Tensorflow. Pada akhirnya model akan dievaluasi berdasarkan hasil pelatihan dan pengujian dengan *dataset uji*.

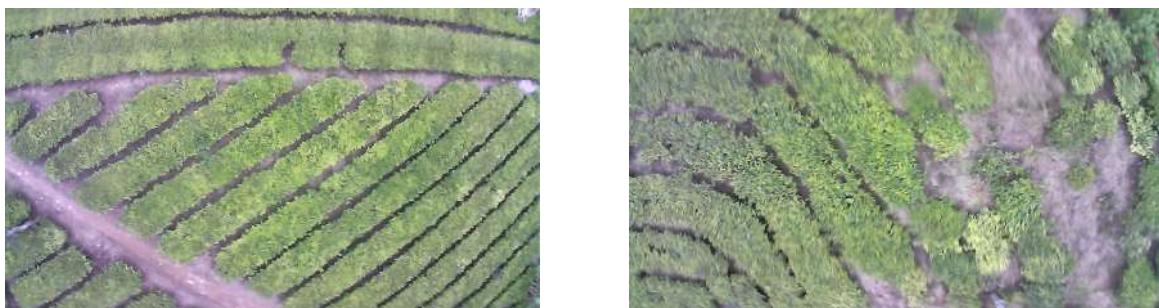
## 4.2 Identifikasi Kebutuhan Data

Pada bagian ini, dilakukan penentuan untuk data apa saja yang dibutuhkan untuk merancang dan mengembangkan model pemetaan dan identifikasi kesiapan petik tanaman teh. Sebagaimana dituliskan sebelumnya, bahwa ada tiga bagian utama yang untuk menghasilkan model pemetaan

dan identifikasi kesiapan petik tanaman teh yakni model penggabungan citra, model pemetaan tanaman teh dan model identifikasi kesiapan petik tanaman teh.

#### 4.2.1 Data Citra Area Perkebunan Teh

Data ini merupakan data utama yang merupakan data citra yang ditangkap dengan menggunakan *drone* yang menghasilkan citra area dari perkebunan teh. Data ini yang digunakan untuk keseluruhan model baik itu model penggabungan citra, pemetaan tanaman teh hingga untuk identifikasi kesiapan petik tanaman teh. Data ini didapatkan dengan melakukan pengambilan data langsung di perkebunan teh yang berada di Pusat Penelitian Teh dan Kina, Gambung, Ciwidey, Kabupaten Bandung. Citra hasil tangkapan *drone* dibagi menjadi dua jenis, yakni citra area perkebunan teh yang siap petik dan area perkebunan teh yang tidak siap petik (setelah petik). Gambaran hasil tangkapan *drone* terhadap area perkebunan teh dapat dilihat pada Gambar 4.5 dimana Gambar 4.5 sebelah kiri adalah area siap petik dan Gambar 4.5 kanan adalah area tidak siap petik.



Gambar 4.5 Pratinjau hasil tangkapan *drone* terhadap area perkebunan teh.

Data citra yang dikumpulkan sebanyak 342 data citra area perkebunan teh hasil tangkapan *drone* yang dimana 221 data merupakan data area yang siap petik dan 121 data lainnya adalah data area yang tidak siap petik. Namun data tersebut tidak sepenuhnya dapat digunakan, dikarenakan ada data citra yang hasil tangkapan citranya termasuk ke kategori tidak layak digunakan seperti fokus yang buruk sehingga tingkat *blur* tinggi, lalu ada juga beberapa tangkapan citra yang bukan menangkap area tanaman teh melainkan menangkap citra area lain seperti area pendaratan *drone* yang berupa jalanan tanah, kemudian juga data yang jarak tangkapan citra terlalu dekat/jauh.

Sebanyak 102 data citra hasil pemilahan akan secara langsung akan digunakan untuk pembangunan model penggabungan citra yang berupa 84 data citra area tanaman teh siap petik dan 18 data citra area tanaman tidak siap petik (setelah petik). Data ini bisa langsung digunakan

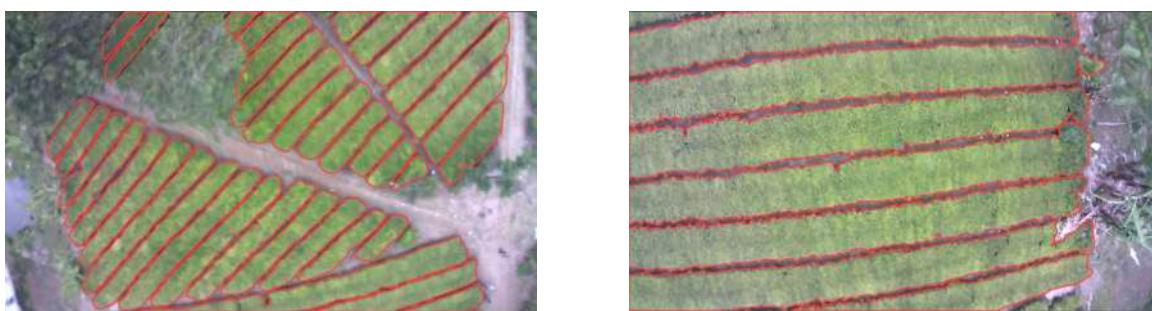
karena pada dasarnya citra hasil tangkapan *drone* ini merupakan potongan-potongan area perkebunan teh yang nantinya setelah dilakukan penggabungan menghasilkan gambaran area perkebunan teh yang lebih luas. Selain itu data citra ini juga nantinya akan diolah untuk kebutuhan pemetaan tanaman teh dan juga perhitungan *vegetation index* dari tanaman teh.

#### 4.2.2 Dataset Objek Tanaman Teh

Data yang dibutuhkan untuk pembangunan model pemetaan tanaman teh menggunakan *Mask R-CNN* adalah dataset berupa gambar dan juga informasi terkait objek didalamnya yang akan dilatih. Pada penelitian ini objek yang akan dilatih adalah tanaman teh, sehingga dibutuhkan data gambar yakni citra hasil tangkapan *drone* terhadap area perkebunan teh yang sudah didapatkan sebelumnya yang diolah terlebih dahulu sehingga terdapat juga informasi objek tanaman teh didalamnya.

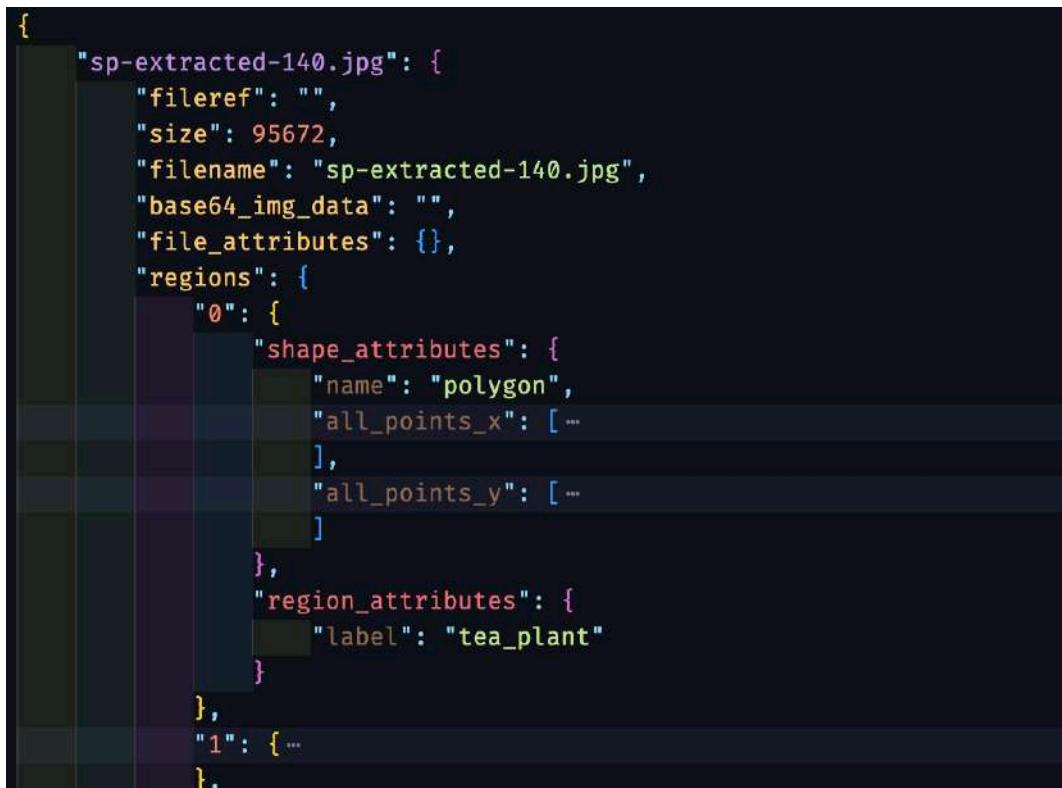
Proses pengolahan tersebut adalah melakukan *annotation* atau *labeling* pada terhadap citra area perkebunan teh. Anotasi disini berfungsi untuk melakukan pendataan terhadap citra dan menandai area mana saja yang merupakan sebuah objek teh. Anotasi tersebut berupa titik-titik yang terhubung menjadi sebuah *polygon* yang mengelilingi objek tanaman teh. Proses ini dibutuhkan untuk membantu model dalam mengenali karakteristik yang terfokus pada objek tanaman teh.

Dalam proses anotasi gambar, penulis menggunakan situs [www.makesense.ai](http://www.makesense.ai) yang merupakan situs gratis dan *open source* untuk melakukan proses anotasi atau *labeling* pada gambar. Dihasilkan sebanyak 157 data citra beserta hasil anotasi/*labelling* pada objek tanaman teh. Pada Gambar 4.6 diperlihatkan pratinjau dari contoh hasil dari anotasi atau *labeling* terhadap objek tanaman teh yang berada dalam citra hasil tangkapan *drone* area perkebunan teh.



Gambar 4.6 Pratinjau hasil anotasi/*labeling* objek tanaman teh pada citra perkebunan teh.

Hasil anotasi akan berupa data yang terdiri atas informasi dari gambar dan juga titik koordinat x dan y dari yang membentuk *polygon* pada objek tanaman teh. Data anotasi akan disimpan dalam format COCO yang merupakan sebuah struktur dalam bentuk JSON yang berisikan label/anotasi dan metadata untuk *dataset*. Struktur JSON yang digunakan dapat dilihat pada Gambar 4.7 berikut:



```
{  
    "sp-extracted-140.jpg": {  
        "fileref": "",  
        "size": 95672,  
        "filename": "sp-extracted-140.jpg",  
        "base64_img_data": "",  
        "file_attributes": {},  
        "regions": {  
            "0": {  
                "shape_attributes": {  
                    "name": "polygon",  
                    "all_points_x": [...],  
                    "all_points_y": [...]  
                },  
                "region_attributes": {  
                    "label": "tea_plant"  
                }  
            },  
            "1": {...}  
        }  
    }  
}
```

Gambar 4.7 Struktur JSON untuk menyimpan label/anotasi dan *metadata* untuk *dataset*.

Gambar 4.3 memperlihatkan struktur JSON yang akan menyimpan seluruh label atau hasil anotasi objek tanaman teh pada citra area perkebunan teh hasil tangkapan *drone*. Tertera *key* utama dari data bertuliskan ‘*sp-extracted-140.jpg*’ ini merupakan nama *file* citra. Hal tersebut menandakan bahwa isi dari data dengan *key* tersebut merupakan hasil informasi yang dibuat berdasarkan *file* citra dengan nama tersebut. *Key* seperti ‘*fileref*’, ‘*size*’, ‘*filename*’, ‘*base64\_img\_data*’ ‘*file\_attributes*’ merupakan *metadata* dari gambar.

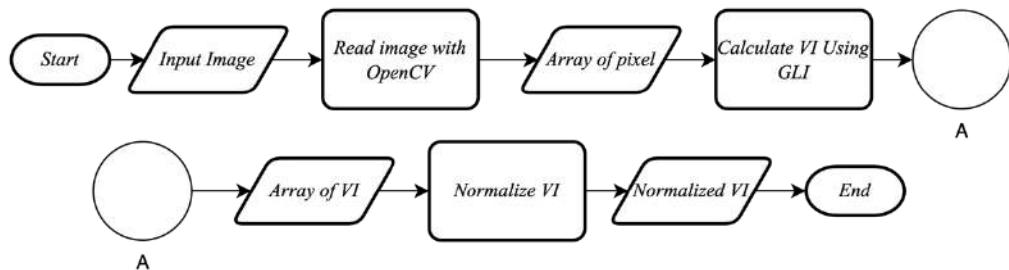
Sedangkan untuk ‘*region*’ bersikan label atau hasil anotasi objek tanaman teh pada gambar tersebut yang berisikan seluruh label yang diawali dengan *key* ‘0’ yang nantinya akan meningkat satu hingga sebanyak label atau anotasi yang telah dibuat. Data tersebut berisikan ‘*shape\_attributes*’ yang merupakan kumpulan titik x dan y yang membentuk sebuah *polygon* serta

‘*region\_attributes*’ yang berisikan nama label yang disini diberi nama ‘*tea\_plant*’ yang menandakan bahwa objek tersebut adalah tanaman teh.

Citra beserta anotasi yang telah dilakukan akhirnya menjadi *dataset* objek tanaman teh yang akan digunakan untuk *training* atau pelatihan model pemetaan tanaman teh menggunakan *Mask R-CNN*. Sehingga nantinya model dapat memisahkan objek tanaman teh dari objek sekitar seperti tanah, pohon, batu dan lain-lain.

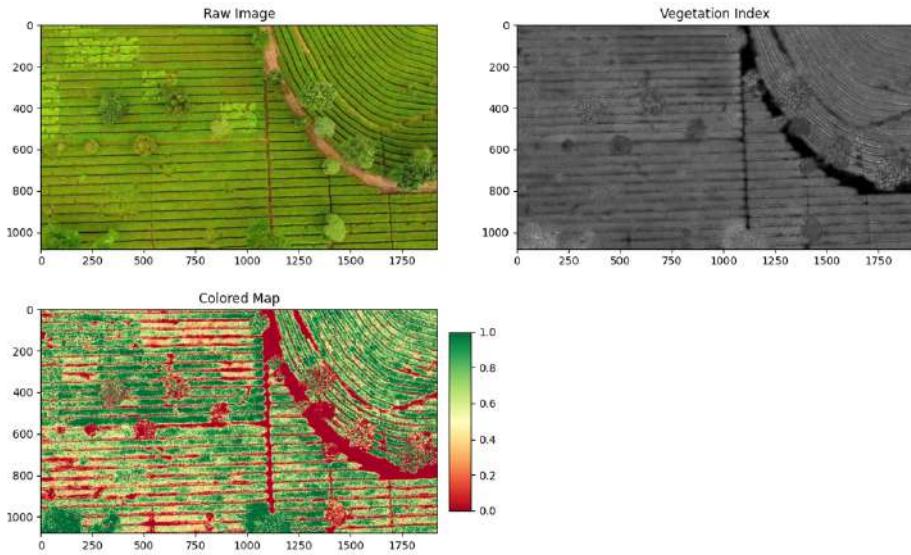
#### 4.2.3 Dataset Vegetation Index Tanaman Teh

Dalam proses identifikasi tingkat kesiapan petik tanaman teh, penelitian ini akan menggunakan *vegetation index* metode *green leaf index* (GLI) sebagai data utama yang menghitung tingkat kehijauan daun pada tanaman teh. Data indeks vegetasi tersebut digunakan untuk membangun model yang akan dilatih untuk dapat menentukan tingkat kesiapan petik tanaman teh. Perhitungan indeks vegetasi akan dilakukan pada citra hasil tangkapan area perkebunan teh menggunakan *drone*. Alur dari perhitungan dapat dilihat pada Gambar 4.8 berikut:



Gambar 4.8 Diagram alir perhitungan *vegetation index* dengan *green leaf index* (GLI).

Berdasarkan Gambar 4.8, proses perhitungan *vegetation index* (VI) diawali dengan *input* gambar atau citra area perkebunan teh hasil tangkapan *drone* yang kemudian dibaca dengan OpenCV yang kemudian menghasilkan *array of pixel* yang merepresentasikan citra tersebut. Selanjutnya dilakukan perhitungan terhadap masing-masing piksel RGB dari gambar tersebut menggunakan metode GLI. Hasil dari perhitungan adalah berupa kumpulan nilai VI atau *array of VI*. VI disini masih berupa nilai dengan rentang yang tidak menentu, tergantung pada hasil perhitungan dari setiap gambar. Maka dari itu untuk mempermudah untuk pembacaan nilai VI, dilakukan normalisasi terhadap nilai VI yang sudah dihasilkan kedalam *range* 0 sampai 1. Dimana 0 menyatakan VI rendah dan 1 menyatakan VI dengan nilai tinggi. Perhatikan Gambar 4.9.



Gambar 4.9 Gambaran hasil dari perhitungan *vegetation index* dengan GLI.

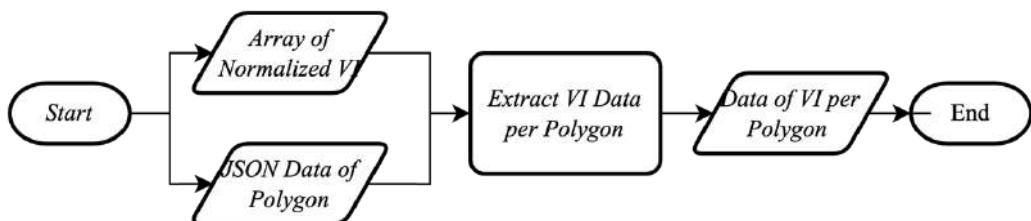
Pada Gambar 4.9 diperlihatkan tiga gambar yang merupakan gambaran atau visualisasi hasil dari proses perhitungan VI menggunakan GLI. Pada gambar pertama (kiri atas) merupakan gambar mentah atau *raw* yang dimana merupakan gambar atau citra area perkebunan teh hasil tangkapan *drone*. Selanjutnya gambar kedua (kanan atas) merupakan gambar hasil dari proses perhitungan VI menggunakan GLI, disana terlihat dimana bagian yang merupakan vegetasi atau tanaman memiliki warna lebih terang yang cenderung ke abu-abu sedangkan bagian yang bukan vegetasi seperti jalan tanah berwarna hitam atau lebih gelap. Hal tersebut menunjukkan bahwa GLI dapat melakukan perhitungan vegetasi terhadap tanaman teh dengan baik. Gambar terakhir (kiri bawah) merupakan visualisasi gambar hasil perhitungan VI yang telah dilakukan normalisasi kemudian dilakukan *color mapping* dengan warna merah-kuning-hijau terhadap setiap indexnya.

Berdasarkan proses perhitungan VI dengan GLI tersebut, maka telah dihasilkan data VI dari citra area perkebunan teh hasil tangkapan *drone*. Namun, hasil perhitungan tersebut masih berdasarkan keseluruhan gambar, yang dimana tentunya juga termasuk kedalamnya nilai VI terhadap objek lain seperti bebatuan, jalanan tanah, pohon pelindung, maupun objek lain selain objek tanaman teh. Oleh karena itu, data VI dapat dikatakan kurang valid karena mencakup keseluruhan apa yang ada pada gambar bukan hanya terfokus pada tanaman teh saja.

Pemanfaatan dataset objek tanaman teh sebelumnya yang merupakan hasil dari proses anotasi atau *labelling* objek tanaman teh dapat dijadikan solusi untuk masalah tersebut. Data anotasi yang berupa *polygon* pada objek tanaman teh dapat dijadikan batasan untuk melakukan perhitungan

VI sehingga hanya terfokus pada objek tanaman teh saja. Sehingga dapat dihasilkan perhitungan yang lebih presisi tanpa adanya gangguan dari objek lain selain tanaman teh.

Hal tersebut dapat dilakukan dengan memanfaatkan *dataset* objek tanaman teh yang merupakan hasil anotasi dari data citra area perkebunan teh. Data anotasi dapat dijadikan batasan untuk melakukan perhitungan *vegetasi index* sehingga hanya terfokus pada sebuah tanaman teh. Sehingga dapat dihasilkan perhitungan yang lebih presisi tanpa adanya gangguan dari objek lain selain tanaman teh. Adapun proses untuk ekstraksi VI data per objek tanaman teh dapat dilihat pada Gambar 4.10.



Gambar 4.10 Diagram alir ekstraksi VI data area tanaman teh berdasarkan *dataset* hasil anotasi.

Proses untuk ekstraksi VI data per objek tanaman teh hanya perlu menambahkan JSON data berisikan koordinat x dan y yang membentuk batas *polygon*. Koordinat tersebut yang digunakan untuk menjadi batasan dan hanya memilih area dalam *polygon* yakni tanaman teh dan menghapus area lain pada data VI yang telah di normalisasi. Berdasarkan dua data tersebut, selanjutnya dilakukan ekstraksi data VI per *polygon* atau per objek tanaman teh. Visualisasi dari ekstraksi data per objek tanaman teh dapat dilihat pada Gambar 4.11. Gambar 4.11 kiri menampilkan data hasil tangkapan *drone* dan Gambar 4.11 kanan menampilkan gambaran hasil ekstraksi terhadap satu area tanaman teh.



Gambar 4.11 Visualisasi ekstraksi VI pada sebuah area tanaman teh berdasarkan data anotasi.

Sehubungan data VI berupa *array*, yang dimana berisikan hasil kalkulasi VI terhadap setiap piksel RGB, maka untuk ekstraksi data yang dapat dilakukan antara lain adalah data besaran piksel pada area tanaman teh, mean (rata-rata), median (nilai tengah) dari nilai VI pada objek tanaman teh, dan juga *plotting* atau pemetaan persebaran data VI. *Plotting* dilakukan kedalam sepuluh bagian terhadap data VI dengan skala 0 sampai 1. Dataset yang dihasilkan memiliki jumlah data sebanyak 4244 data hasil ekstraksi VI terhadap lajur/objek tanaman teh. Berikut pratinjau dari *dataset* VI hasil ekstraksi ditampilkan pada Gambar 4.12.

<b>status</b>	<b>size</b>	<b>mean</b>	<b>median</b>	<b>0.0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>	<b>0.6</b>	<b>0.7</b>	<b>0.8</b>	<b>0.9</b>
0	1283284	0.5077607064883576	0.40784313725490196	218051	12671	227289	58904	240679	41202	42013	38154	33406	370915
0	1271076	0.5605319995476105	0.4885813148788929	165597	20069	177922	85548	196815	59556	61160	54702	48988	400719
0	1079664	0.6150465468833208	0.6330642060745868	100829	22451	113241	78810	143669	60031	65578	60734	59089	375232
0	1067512	0.6328920041651303	0.6558246828143022	74020	18321	87327	103671	142072	67644	74518	78139	72697	349103
1	1062648	0.56245892863390017	0.48927335640138436	131767	13853	140316	78985	171153	52840	53992	53972	48531	317239
0	996468	0.5989134100690158	0.5930795847750866	104319	14138	112427	71777	144890	53525	57749	55284	53107	329252
0	945996	0.6010645254652112	0.5989234909650134	97221	13716	105353	68803	137714	50533	54681	53540	50279	314156

Gambar 4.12 Pratinjau *dataset* VI per objek tanaman teh.

Gambar 4.12 memperlihatkan dari *dataset* VI hasil proses ekstraksi per objek tanaman teh. Adapun kolom ‘*status*’ disana merupakan kolom yang menandakan apakah tanaman teh tersebut memiliki status siap petik (1) atau tidak siap petik (0). Sedangkan data lainnya merupakan data berdasarkan citra tanaman teh dan juga perhitungan terhadap VI seperti ‘*mean*’, ‘*median*’ dan juga *plot* yang merupakan persebaran banyaknya data dari 0.0 sampai dengan 0.9. Kolom berjudul ‘*0.0*’ menandakan kolom berisikan banyaknya data/piksel dengan nilai VI dalam kisaran 0.0 sampai dengan 0.09, begitu juga dengan kolom berjudul ‘*0.1*’ yang berisikan banyaknya data/piksel dengan nilai VI dalam kisaran 0.1 sampai dengan 0.19, dan seterusnya.

### 4.3 Pengembangan Model

#### 4.3.1 Pengembangan Model Penggabungan Citra

Pada Tahapan ini model penggabungan citra yang telah dirancang kemudian lanjut untuk dikembangkan. Berdasarkan rancangan tersebut berikut hasil dari pengembangan model penggabungan citra:

##### 1. Persiapan Data Citra

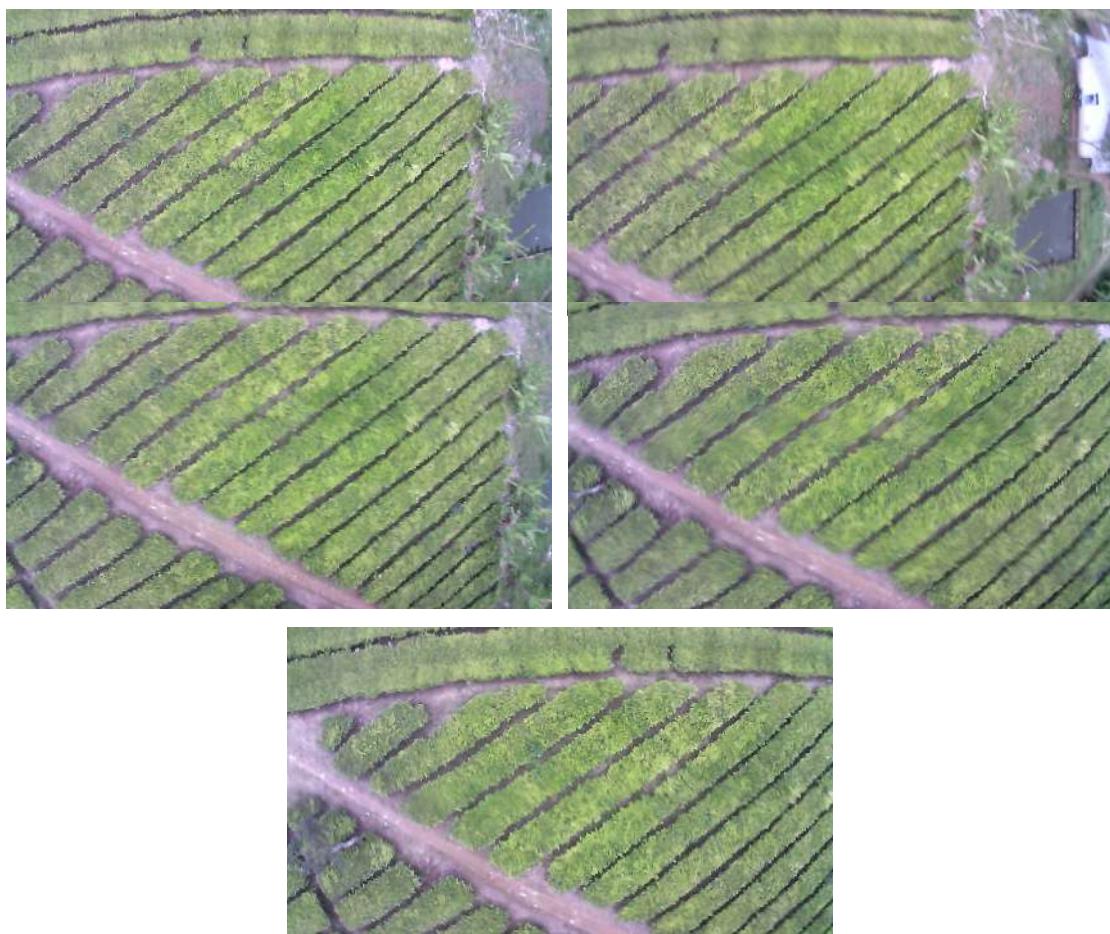
Pada proses ini, citra hasil tangkapan *drone* yang telah dikumpulkan sebelumnya akan digunakan untuk pengembangan model penggabungan citra. Kumpulan citra ini kemudian

dikelompokkan berdasarkan kesesuaian citra dilihat dari bagian *overlap* antara satu dengan yang lainnya. Dihasilkan sebanyak 9 kelompok citra terdiri dari 6 diantaranya adalah kelompok citra yang siap petik dan 3 sisanya adalah kelompok citra tidak siap petik (setelah petik). Jumlah citra perbagian kelompok dapat dilihat pada Tabel 4.1.

Tabel 4.1 Pengelompokan data citra untuk proses penggabungan gambar.

No	Kelompok Citra	Jumlah Citra
1	Siap Petik 1	28
2	Siap Petik 2	14
3	Siap Petik 3	19
4	Siap Petik 4	5
5	Siap Petik 5	7
6	Siap Petik 6	11
7	Tidak Siap Petik 1	4
8	Tidak Siap Petik 2	7
9	Tidak Siap Petik 3	7
Total Citra		102

Masing-masing dari kelompok citra tersebut akan dilakukan proses penggabungan gambar secara terpisah dan akan 9 citra hasil penggabungan citra dengan metode *image stitching*. Berikut contoh salah satu data kelompok citra yang diberi nama ‘Siap Petik 4’ yang memiliki 5 citra didalamnya ditampilkan pada Gambar 4.13.



Gambar 4.13 Pratinjau kelompok citra Siap Petik 4.

## 2. Pre-Processing

### a. Pre-Processing

Proses *resize image* ini dilakukan sebagai langkah *pre-processing* untuk gambar sehingga resolusi gambar direduksi menjadi resolusi yang lebih rendah dengan tujuan untuk mempercepat proses penggabungan citra dan mengurangi beban komputasi. Gambar 4.14 menampilkan hasil dari proses *resize image*.

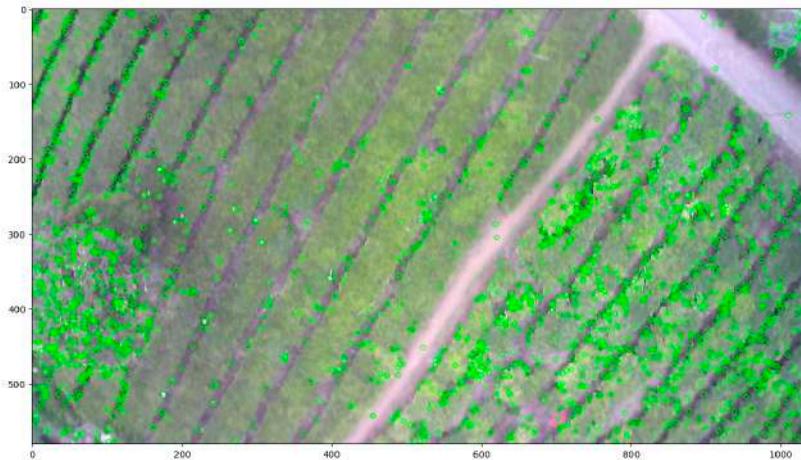
Original Size:	(1280, 720)	$\rightarrow$	921,600 px ~ 1
Medium Size:	(1033, 581)	$\rightarrow$	600,173 px ~ 0.6
Low Size:	(422, 237)	$\rightarrow$	100,014 px ~ 0.1
Final Size:	(1280, 720)	$\rightarrow$	921,600 px ~ 1

Gambar 4.14 Hasil resolusi citra terhadap proses *resize image*.

Pada Gambar 4.14 diperlihatkan gambar *original* yang memiliki resolusi awal 1080x720 dengan total piksel sebanyak 921600 piksel, citra ini bisa disebut dengan skala asli atau citra dengan skala 1. Selanjutnya adalah citra hasil reduksi resolusi menjadi *medium* dengan resolusi menjadi 1033x581 dengan total piksel sebanyak 600173 piksel atau sekitar 0.6 dari skala asli. Kemudian untuk ukuran *low* dihasilkan resolusi sebesar 442x237 dengan total piksel sebanyak 100014 piksel atau sekitar 0.1 dari skala asli. Pada akhirnya *final size* adalah ukuran semula yang sama dengan ukuran *original*.

### 3. Finding Features

Tahapan ini berisikan proses pendekripsi fitur-fitur yang terdapat pada setiap citra. Adapun metode yang digunakan untuk deteksi fitur tersebut adalah metode *Scale-Invariant Feature Transform* (SIFT). Berikut ditampilkan hasil dari pendekripsi fitur menggunakan metode SIFT pada Gambar 4.15.



Gambar 4.15 Hasil pendekripsi fitur pada citra menggunakan metode SIFT.

### 4. Matching Features

Tahapan ini merupakan tahapan dimana setiap fitur yang diekstraksi dari citra dilakukan pencocokan dengan fitur pada citra lain. Pada bagian ini juga nantinya setiap gambar dihitung tingkat *confidence* berdasarkan kecocokan fitur dengan gambar lainnya. Adapun formula yang digunakan untuk perhitungan *confidence* diambil dari penelitian terkait *image stitching* oleh (Brown & Lowe, 2007) sebagai berikut:

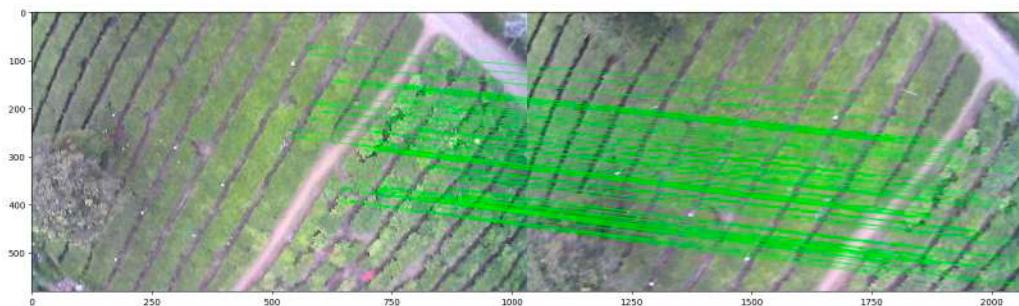
$$\text{confidence} = \text{number of inliers} / (8 + 0.3 * \text{number of matches})$$

*Number of inliers* merupakan hasil perhitungan menggunakan *random sample consensus* (RANSAC) dan *number of matches* merupakan total fitur yang cocok antar citra. Berdasarkan perhitungan tingkat *confidence* dihasilkan *array* berisikan nilai *confidence* dari satu citra terhadap citra lainnya. *Array* yang dimaksud dapat dilihat pada Gambar 4.16.

```
array([[0.          , 0.30534351, 0.91533181, 0.34965035, 0.4        , 0.6097561 , 0.60402685, 0.32786885],
       [0.30534351, 0.          , 0.36363636, 0.88372093, 1.07692308, 0.44776119, 0.36363636, 0.3539823 ],
       [0.91533181, 0.36363636, 0.          , 0.42857143, 0.38167939, 0.93385214, 1.2        , 0.4        ],
       [0.34965035, 0.88372093, 0.42857143, 0.          , 0.87662338, 1.11428571, 0.74889868, 0.29850746],
       [0.4        , 1.07692308, 0.38167939, 0.87662338, 0.          , 0.49180328, 0.38461538, 0.3125  ],
       [0.6097561 , 0.44776119, 0.93385214, 1.11428571, 0.49180328, 0.          , 1.94954128, 0.38167939],
       [0.60402685, 0.36363636, 1.2        , 0.74889868, 0.38461538, 1.94954128, 0.          , 0.29850746],
       [0.32786885, 0.3539823 , 0.4        , 0.29850746, 0.3125  , 0.38167939, 0.29850746, 0.        ]])
```

Gambar 4.16 *Output* dari pencocokan fitur dan perhitungan *confidence*.

Pada Gambar 4.16 diperlihatkan hasil dari nilai *confidence* berdasarkan pencocokan yang dilakukan antara satu citra dengan seluruh citra. Nilai tepat 0 menunjukkan bahwa itu menunjukkan bahwa sebuah citra tidak melakukan pencocokan dengan dirinya sendiri. Sehingga sebagai contoh *index* pertama pada array memiliki arti bahwa itu merupakan citra pertama yang dimana *confidence* dengan citra kedua sebesar 0.30534351, kemudian dengan gambar kedua sebesar 0.91533181 dan seterusnya. Untuk visualisasi pencocokan fitur dapat dilihat pada Gambar 4.17.

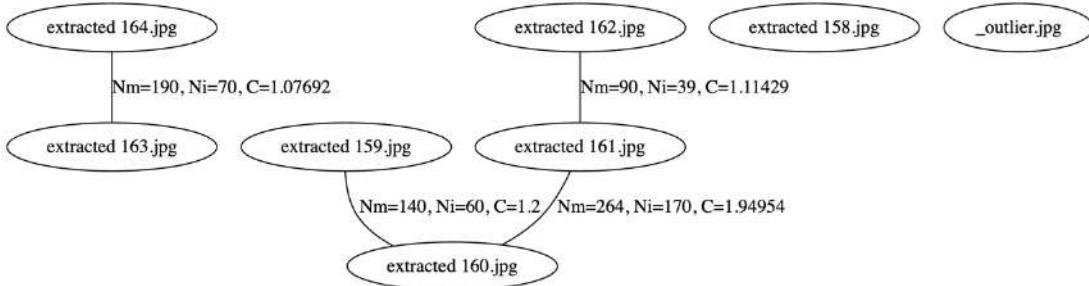


Gambar 4.17 Visualisasi pencocokan fitur terhadap dua citra.

## 5. Outlier Removal

Tahapan selanjutnya adalah tahapan dimana citra gambar yang tidak memiliki kecocokan atau tidak memiliki koneksi berdasarkan nilai *confidence* yang rendah akan dikecualikan. Sehingga *subset* hanya akan berisi kan citra-citra yang relevan karena memiliki kecocokan dan tingkat confidence yang tinggi. Adapun *threshold* dari tingkat *confidence* yang digunakan disini adalah bernilai 1. Sehingga *confidence* dengan nilai dibawah 1 dianggap tidak relevan.

Berdasarkan pencocokan fitur sebelumnya, didapatkan data mana saja yang termasuk kedalam citra yang relevan dan tidak, untuk lebih mudahnya dapat dilihat pada Gambar 4.18.



Gambar 4.18 Diagram *visualiasi* hubungan dari citra berdasarkan *confidence*.

Pada Gambar 4.18 terdapat 3 kelompok citra, yang pertama adalah 2 kelompok yang masing-masing memiliki koneksi dengan citra lainnya, seperti citra ‘*extracted 164.jpg*’ dengan ‘*extracted 164.jpg*’ dan juga kelompok satunya yang terdiri dari ‘*extracted 159.jpg*’, ‘*extracted 160.jpg*’, ‘*extracted 161.jpg*’ dan ‘*extracted 162.jpg*’. Sedangkan kelompok yang terakhir adalah kelompok yang dimana citra tidak memiliki koneksi dengan citra lain yakni ‘*extracted 158.jpg*’ dan ‘*\_outlier.jpg*’. Keterangan ‘Nm’ merupakan keterangan untuk *number of matches* yang berarti banyaknya fitur yang cocok antara dua citra tersebut. Kemudian untuk ‘Ni’ adalah *number of inliers* yang merupakan hasil perhitungan menggunakan RANSAC. Terakhir ‘C’ merupakan *confidence* atau nilai kepercayaan diri dari dua citra tersebut.

Akhirnya dibuatlah *subset* yang berisikan citra data citra yang memiliki koneksi atau relevan berdasarkan hasil perhitungan *confidence*. Adapun terdapat 2 kelompok yang memiliki masing-masing relevan, yakni kelompok dengan 2 citra dan kelompok dengan 4 citra, maka yang dijadikan *subset* adalah kelompok dengan citra terbanyak yakni yang berisikan 2 citra. Sehingga berikut hasil *subset* yang dibuat diperlihatkan pada Gambar 4.19.

```

[
    'data_kebun/siap_petik/5/extracted 159.jpg',
    'data_kebun/siap_petik/5/extracted 162.jpg',
    'data_kebun/siap_petik/5/extracted 161.jpg',
    'data_kebun/siap_petik/5/extracted 160.jpg'
]
[[0.        0.42857143 0.93385214 1.2        ],
 [0.42857143 0.        1.11428571 0.74889868],
 [0.93385214 1.11428571 0.        1.94954128],
 [1.2        0.74889868 1.94954128 0.        ]]

```

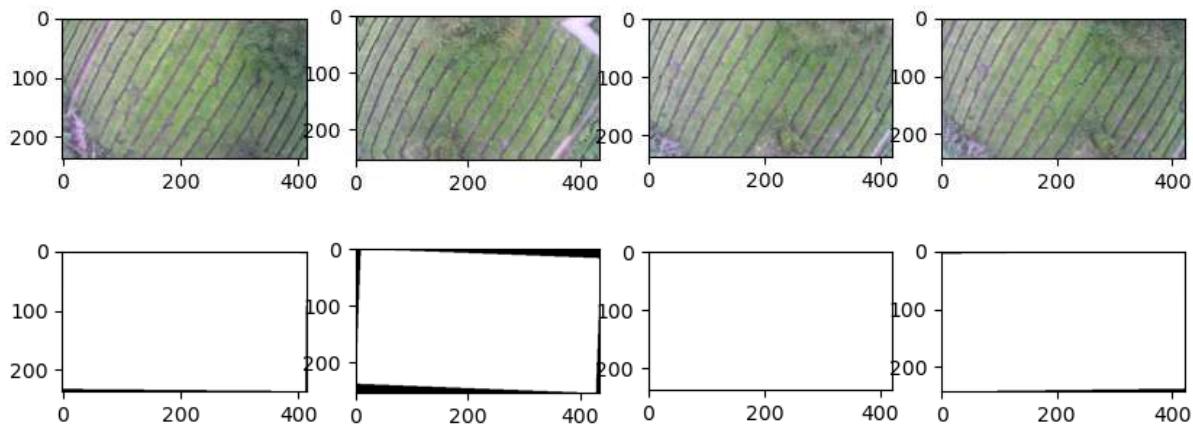
Gambar 4.19 Data *subset* berisikan *filename* citra yang relevan untuk penggabungan citra.

## 6. Homography Estimation

Pada proses ini dilakukan penyesuaian pada kamera sehingga mendapatkan *homography*. Adapun penyesuaian yang dilakukan adalah yang pertama pada *camera estimator* disini menggunakan *homography-based estimator*. *Homography-based estimator* digunakan untuk memperkirakan transformasi homografi antara dua gambar atau lebih. Homografi adalah transformasi perspektif yang memiliki delapan parameter. Ini dapat menggambarkan transformasi linier antara dua bidang datar dalam ruang tiga dimensi. Hal ini cocok digunakan ketika gambar memiliki distorsi perspektif yang signifikan atau ketika ada rotasi dan pergeseran yang besar antara gambar. Kemudian setelah itu dilakukan *adjustment* dan *correction* terhadap parameter kamera yang telah diestimasi sebelumnya sehingga menghasilkan parameter kamera yang lebih sesuai berdasarkan *adjustment* dan *correction*.

## 7. Warp Images

Tahapan berikutnya adalah *warp images* atau transformasi citra untuk menggabungkan citra sesuai dengan fitur-fitur dan penyesuaian yang telah dilakukan sebelumnya. Pada proses ini setiap citra dilakukan transformasi sehingga sesuai dengan hasil dari penyesuaian kamera sebelumnya, mulai dari estimasi *homography* hingga *adjustment* dan *correction*. Hasil dari transformasi citra dapat dilihat pada Gambar 4.20 berikut.



Gambar 4.20 Hasil dari transformasi setiap citra yang akan digabung pada proses *warping*.

Terlihat pada Gambar 4.20, tidak terlalu banyak terjadi transformasi dari setiap citra. Citra kedua memiliki transformasi yang terlihat lebih dominan dibanding citra lainnya. Hal tersebut menandakan bahwa citra telah dilakukan penyesuaian sehingga dapat sejajar dan cocok ketika dilakukan penggabungan dengan citra lainnya. Pada proses ini pun dihasilkan *corners* atau

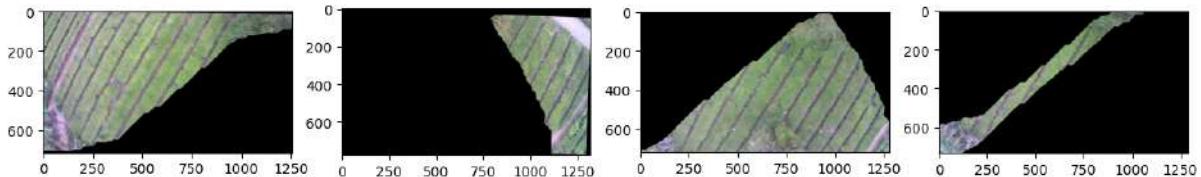
posisi dan juga *sizes* atau ukuran dari setiap citra pada *final plane*. Hasil dari posisi dan ukuran dapat dilihat pada Gambar 4.21.

```
corners: [(-862, 26531), (-471, 26578), (-640, 26648), (-725, 26632)]
sizes : [(1256, 715), (1319, 777), (1279, 720), (1287, 734)]
```

Gambar 4.21 Hasil dari posisi (*corners*) dan ukuran (*size*) akhir dari setiap citra.

Masing-masing *corners* dan *sizes* memiliki empat data yang merupakan data untuk setiap citra yang akan digabungkan. Terlihat pada Gambar 4.21, *corners* berisikan koordinat x dan y pada *final plane* atau *canvas* akhir tempat nanti citra digabungkan satu sama lain. Sedangkan untuk *sizes* berisikan ukuran *width* dan *height* dari citra, terlihat ada citra yang pada akhirnya menjadi sedikit lebih besar dan juga lebih kecil.

Selanjutnya pada tahapan ini juga dilakukan proses untuk menentukan garis transisi dari setiap citra yang digabungkan. Sehingga bagian citra yang digunakan ada bagian dari citra yang tidak dimiliki citra lainnya. *Seam mask* ini diperoleh dari citra yang sudah melewati proses *warping* sebelumnya. Hasil dari *seam mask* dapat dilihat pada Gambar 4.22.



Gambar 4.22 Hasil dari *seam mask*.

Setiap citra diperlihatkan bagaimana citra akan digabungkan berdasarkan garis transisi yang telah dihasilkan pada proses ini. Gambar 4.22 memperlihatkan gambaran tersebut, dimana masing-masing gambar telah dilakukan pemotongan atau *masking* sehingga tidak semua bagian dari citra yang digunakan.

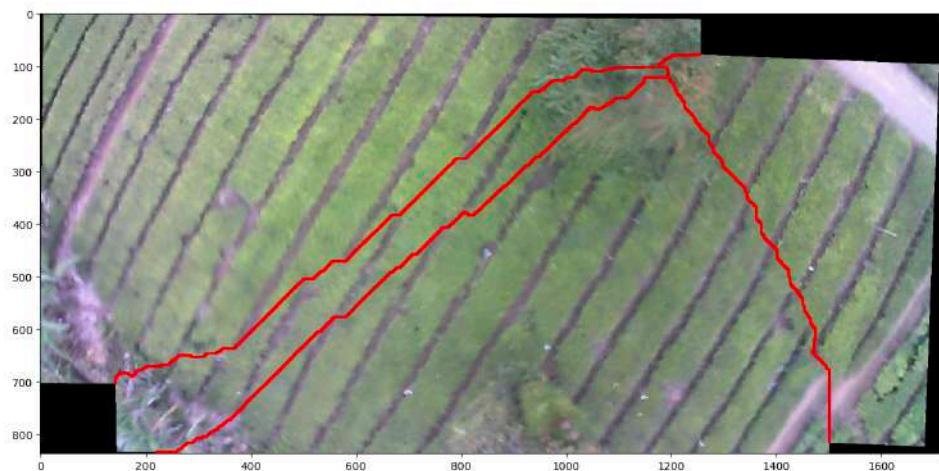
#### 8. *Blending*

Tahapan akhir dari proses penggabungan citra tentunya adalah proses *blending* atau penggabungan. Pada proses ini digunakan hasil dari proses sebelumnya yakni ukuran dan posisi gambar serta juga garis transisi yang telah ditentukan. Sehingga dihasilkan citra utuh hasil penggabungan dari setiap citra. Berikut contoh dari citra hasil dari proses penggabungan citra dapat dilihat pada Gambar 4.23.



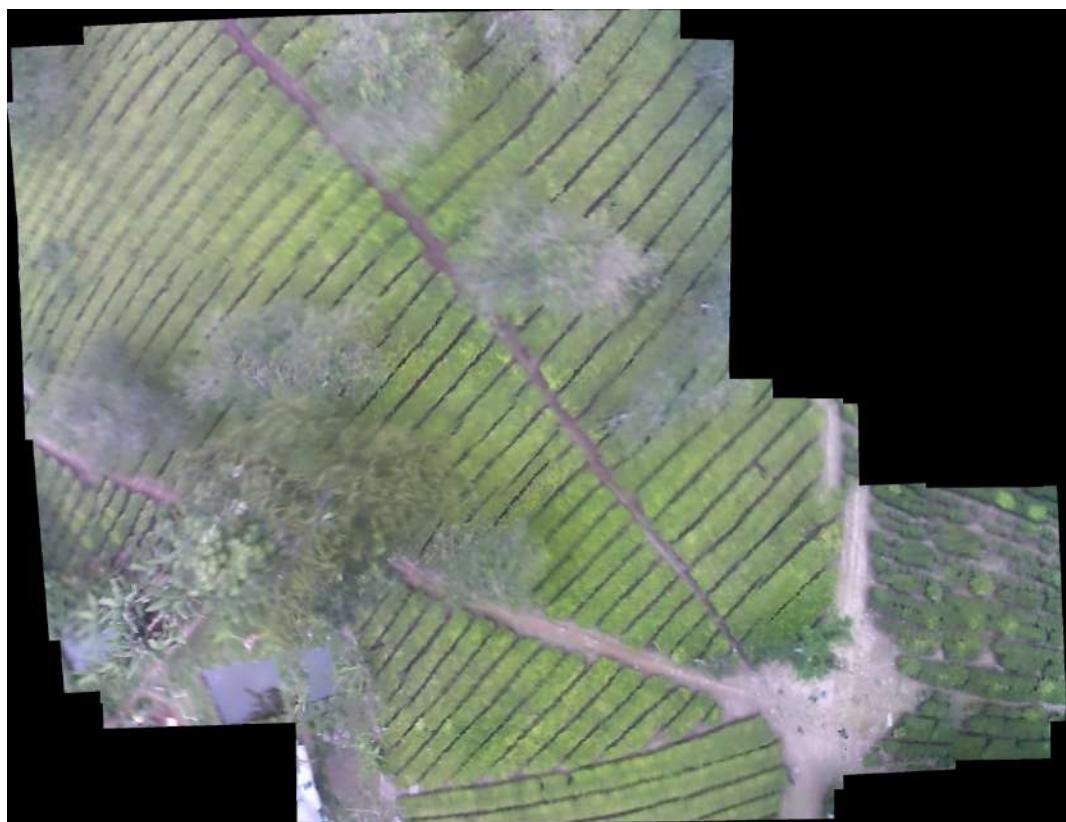
Gambar 4.23 Citra hasil proses *blending*.

Selain itu, dengan memanfaatkan *seam mask*, dapat diperlihatkan garis transisi yang telah ditentukan pada setiap citra dan divisualisasikan pada citra akhir hasil blending. Berikut hasilnya dapat dilihat pada Gambar 4.24.



Gambar 4.24 Visualisasi garis transisi setiap citra pada citra akhir hasil penggabungan.

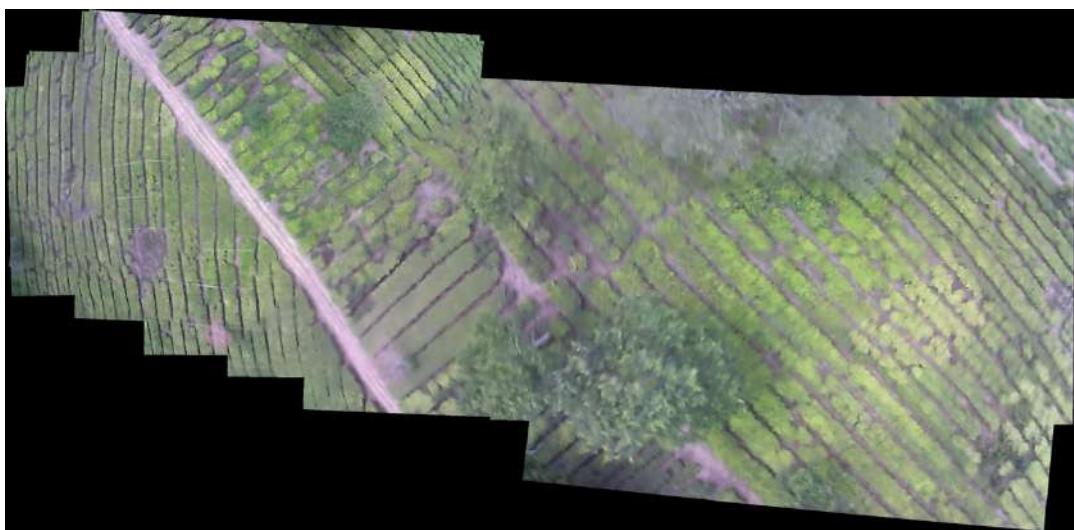
Proses penggabungan citra dilakukan terhadap setiap kelompok citra yang telah dibuat. Adapun hasil dari penggabungan citra pada setiap kelompok citra dapat dilihat pada kumpulan gambar yang ditampilkan pada Gambar 4.25, Gambar 4.26, Gambar 4.27, Gambar 4.28, Gambar 4.29, Gambar 4.30, Gambar 4.31, Gambar 4.32, Gambar 4.33.



Gambar 4.25 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 1’.



Gambar 4.26 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’.



Gambar 4.27 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’.



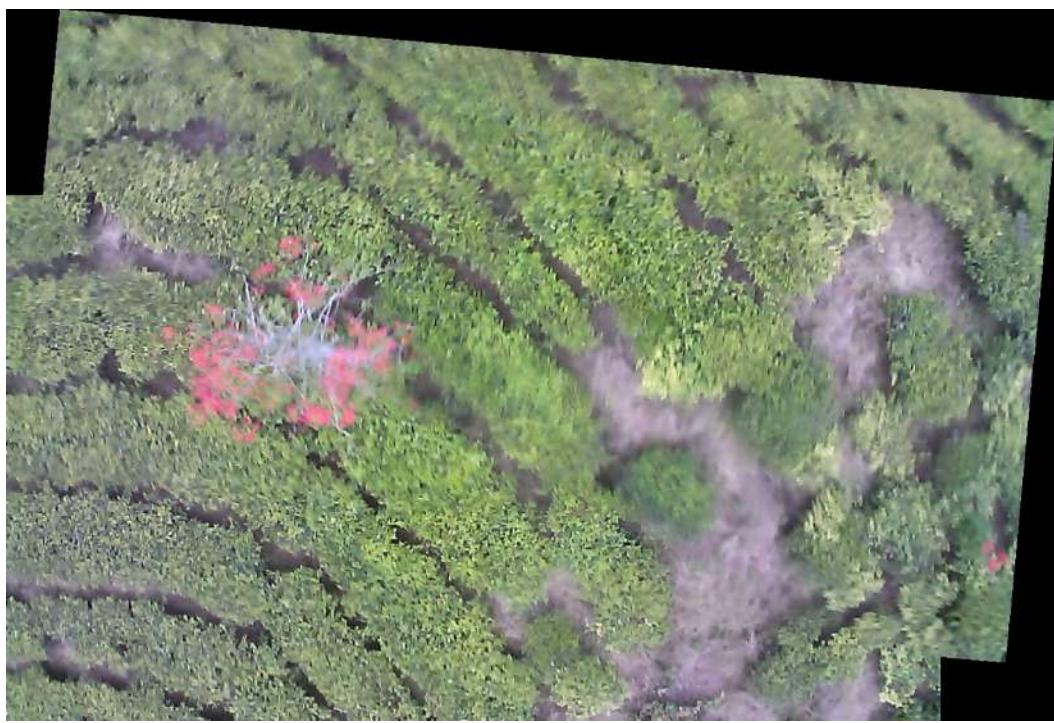
Gambar 4.28 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 4’.



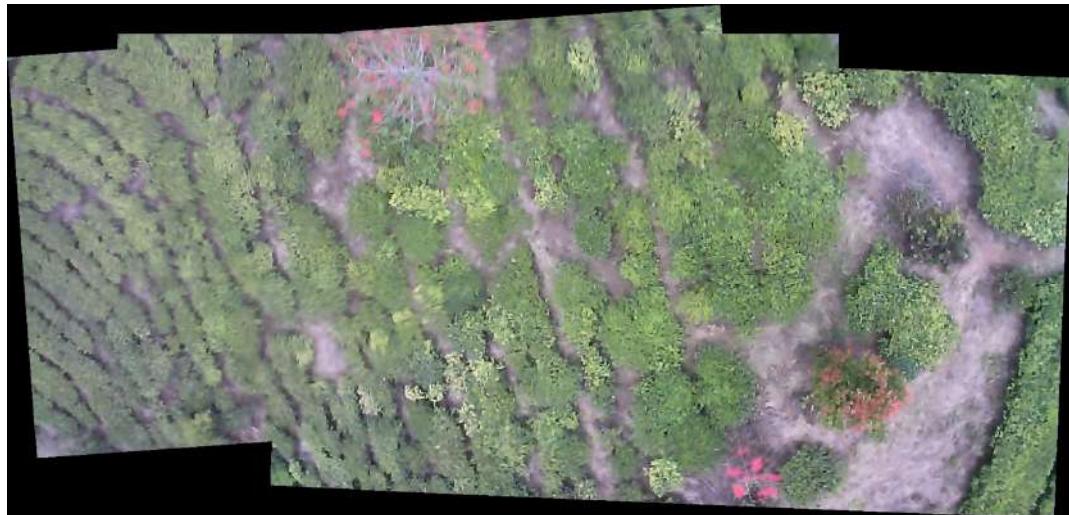
Gambar 4.29 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 5’.



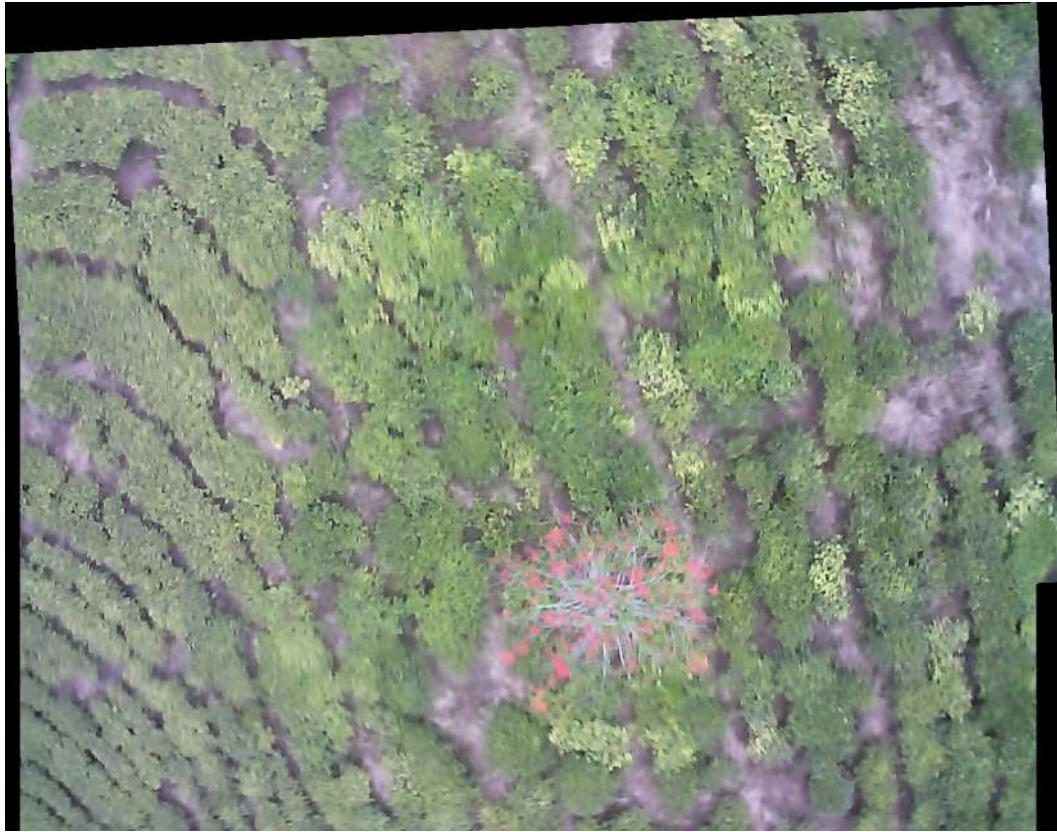
Gambar 4.30 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 6’.



Gambar 4.31 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 1’.



Gambar 4.32 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 2’.



Gambar 4.33 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 3’.

#### 4.3.2 Pengembangan Model Pemetaan Tanaman Teh

Pada tahapan pengembangan model pemetaan tanaman teh, metode yang digunakan adalah *Mask R-CNN* dengan bantuan pengimplementasian menggunakan *framework Detectron2*. Adapun proses yang dilakukan untuk pengembangan model ini ada empat tahapan mencakup: persiapan *dataset*, konfigurasi model *mask R-CNN*, pelatihan model dan akhirnya hasil dari prediksi model yang dilatih.

##### 1. Persiapan *Dataset*

*Dataset* yang digunakan untuk pembangunan model pemetaan tanaman teh merupakan data citra area perkebunan teh hasil tangkapan *drone* yang telah dilakukan proses anotasi atau *labelling* terhadap objek tanaman teh. *Dataset* tersebut kemudian didaftarkan kedalam Detectron2 yang melibatkan adanya penyesuaian format *dataset* yang sudah disiapkan ke format standar yang digunakan oleh Detectron2.

Dalam konteks ini, *dataset* teh yang sudah disiapkan memiliki format tersendiri yakni format COCO *dataset*, sehingga perlu dibuatkan fungsi khusus yang dapat mengurai dan

mempersiapkannya ke dalam format standar yang digunakan oleh Detectron2. Proses ini mencakup penyesuaian *label*, *metadata*, dan atribut lainnya agar sesuai dengan persyaratan yang diterapkan oleh model deteksi objek yang ada dalam *framework* Detectron2. Penjelasan lebih jelas dapat dilihat pada alur atau gambaran proses dari proses registrasi *dataset*:

1. Buat fungsi untuk mendapatkan informasi dataset

```
'def get_tea_dicts(img_dir)':
```

- Gabungkan *path JSON file* dengan direktori gambar
- Buka JSON file
- Inisialisasi *list 'dataset\_dicts'*
- Untuk setiap indeks dan nilai dalam data annotasi:
  - Inisialisasi *record* sebagai *dictionary* kosong
  - Ambil nama *file* gambar
  - Ambil tinggi dan lebar gambar
  - Set '*file\_name*', '*image\_id*', '*height*', dan '*width*' pada '*record*'
  - Ambil informasi *region* dari data anotasi
  - Inisialisasi *list objs* untuk menyimpan informasi objek
  - Untuk setiap *region* dalam data anotasi:
    - Ambil informasi *bounding box* dan *mode*
    - Ambil informasi segmentasi
    - Tentukan '*category\_id*'
    - Tambahkan informasi objek ke dalam *list objs*
    - Set *annotations* pada *record* dengan *list objs*
    - Tambahkan *record* ke dalam '*dataset\_dicts*'
  - Kembalikan *dataset\_dicts*

2. Registrasi *dataset* untuk *set* pelatihan dan validasi

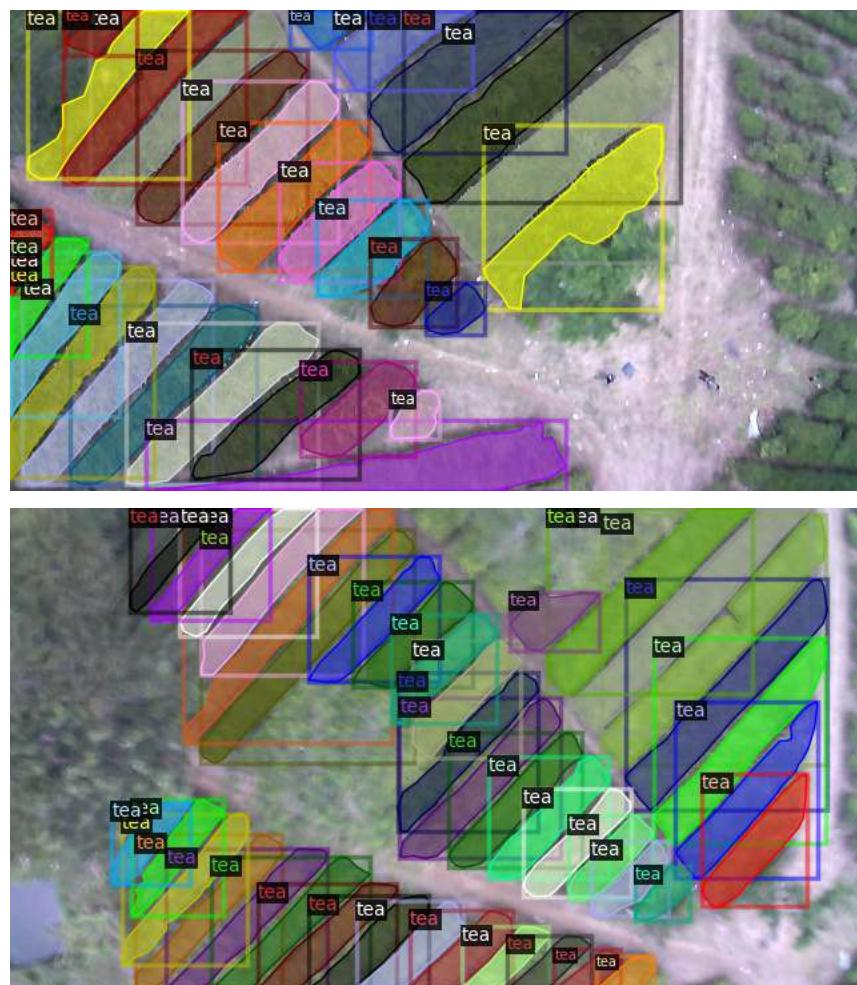
- Untuk setiap "*train*" dan "*val*":
  - Daftarkan *dataset* dengan '*DatasetCatalog.register*'
  - Set *metadata* dengan '*MetadataCatalog.get*'

4. Dapatkan *metadata* untuk *set* pelatihan

- '*tea\_metadata = MetadataCatalog.get("tea\_train")*'

Adapun *dataset* yang telah disiapkan sebelumnya berupa 157 *file* data citra beserta JSON *file* yang berisikan data hasil anotasi atau *labelling* objek tanaman teh. Selanjutnya, sebelum dilakukan pendaftaran *dataset*, dilakukan terlebih dahulu pembagian *dataset* menjadi dua bagian, yakni dataset untuk *training* dan dataset untuk *evaluation* dengan perbandingan data sebesar 15% data *training* dan 85 % data *evaluation*. Jumlah dataset yang dihasilkan adalah sebanyak 25 data untuk *dataset training* dan untuk *dataset evaluation* sebanyak 132 data.

Kemudian untuk melakukan konfirmasi bahwa *dataset* telah berhasil didaftarkan dilakukan penampilan citra dari dataset yang telah terdaftar beserta *label* dari citra tersebut. Berikut Gambar 4.34 menampilkan hasil dari tahapan persiapan *dataset* dan konfirmasi bahwa *dataset* telah terdaftar menggunakan format *dataset* pada Detectron2 untuk mengkonfirmasi bahwa *dataset* berhasil didaftarkan.



Gambar 4.34 Konfirmasi pendaftaran *dataset* dengan menampilkan citra beserta *label*.

## 2. Konfigurasi Model *Mask R-CNN*

Dalam implementasi untuk pemetaan objek tanaman teh dengan *Mask R-CNN*, perlu dilakukan konfigurasi model *Mask R-CNN* pada Detectron2. Berikut adalah gambar dari konfigurasi yang digunakan untuk model yang nantinya akan dilatih menggunakan *framework* Detectron2.

```
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("tea_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 1500
cfg.SOLVER.STEPS = []
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 512
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1
```

Gambar 4.35 Konfigurasi model *Mask R-CNN* pada *framework* Detectron2.

Pada Gambar 4.35, konfigurasi dimulai dengan inisialisasi konfigurasi menggunakan fungsi ‘*get\_cfg()*’. Selanjutnya, konfigurasi model diimpor dari file YAML yang disediakan oleh model Zoo dengan menggunakan metode ‘*merge\_from\_file*’. Dalam hal ini, konfigurasi *default* yang digunakan adalah model ‘*mask\_rcnn\_R\_50\_FPN\_3x*’ yang merupakan *pre-trained model* yang tersedia pada Detectron2. Namun selain itu, model yang akan dilakukan pelatihan adalah model ‘*mask\_rcnn\_R\_101\_FPN\_3x*’. Perbedaannya dari kedua model tersebut adalah pada arsitektur *backbone* yang dimana ‘*R\_50*’ menandakan bahwa *backbone* yang digunakan adalah ResNet-50 sedangkan ‘*R\_101*’ berarti *backbone* menggunakan ResNet-101. Model tersebut memang untuk implementasi *Mask R-CNN* dengan *backbone* ResNet 50/101 dan FPN yang telah dilatih menggunakan *dataset* COCO.

Selanjutnya, *dataset* pelatihan (*train*) dan pengujian (*test*) ditentukan dengan mengatur variabel ‘*DATASETS.TRAIN*’ dan ‘*DATASETS.TEST*’. Pengaturan ‘*DataLoader*’ juga diatur dengan menentukan jumlah pekerja (‘*NUM\_WORKERS*’) yang akan digunakan oleh ‘*DataLoader*’ untuk memuat data secara paralel. Jumlah pekerja ini mengacu pada jumlah proses yang akan digunakan untuk memuat data pelatihan secara paralel. Pada umumnya, ini adalah jumlah pekerja (*workers*) atau proses yang akan berjalan secara bersamaan untuk memuat batch data dari penyimpanan atau dataset.

Bobot awal model diatur menggunakan ‘*MODEL WEIGHTS*’, diambil dari *pre-trained model* yang sama juga yakni ‘*mask\_rcnn\_R\_50\_FPN\_3x*’ dan ‘*mask\_rcnn\_R\_101\_FPN\_3x*’. Ini memungkinkan model untuk memulai pelatihan dengan bobot yang sudah terlatih pada dataset COCO, memberikan inisialisasi yang baik untuk tugas deteksi objek khususnya pada dataset objek tanaman teh yang telah disiapkan.

Pengaturan pelatihan dan optimisasi model diatur selanjutnya. Jumlah gambar per batch (‘*IMS\_PER\_BATCH*’), laju pembelajaran awal (‘*BASE\_LR*’), jumlah iterasi maksimum (‘*MAX\_ITER*’), dan langkah-langkah penjadwalan pembelajaran (‘*STEPS*’) ditentukan. Parameter *ROI Heads* juga diatur dengan menentukan ukuran batch per gambar (‘*ROI\_HEADS.BATCH\_SIZE\_PER\_IMAGE*’) dan jumlah kelas (‘*ROI\_HEADS.NUM\_CLASSES*’). Dalam konteks ini, *dataset* hanya memiliki satu kelas, yaitu “*tea*”.

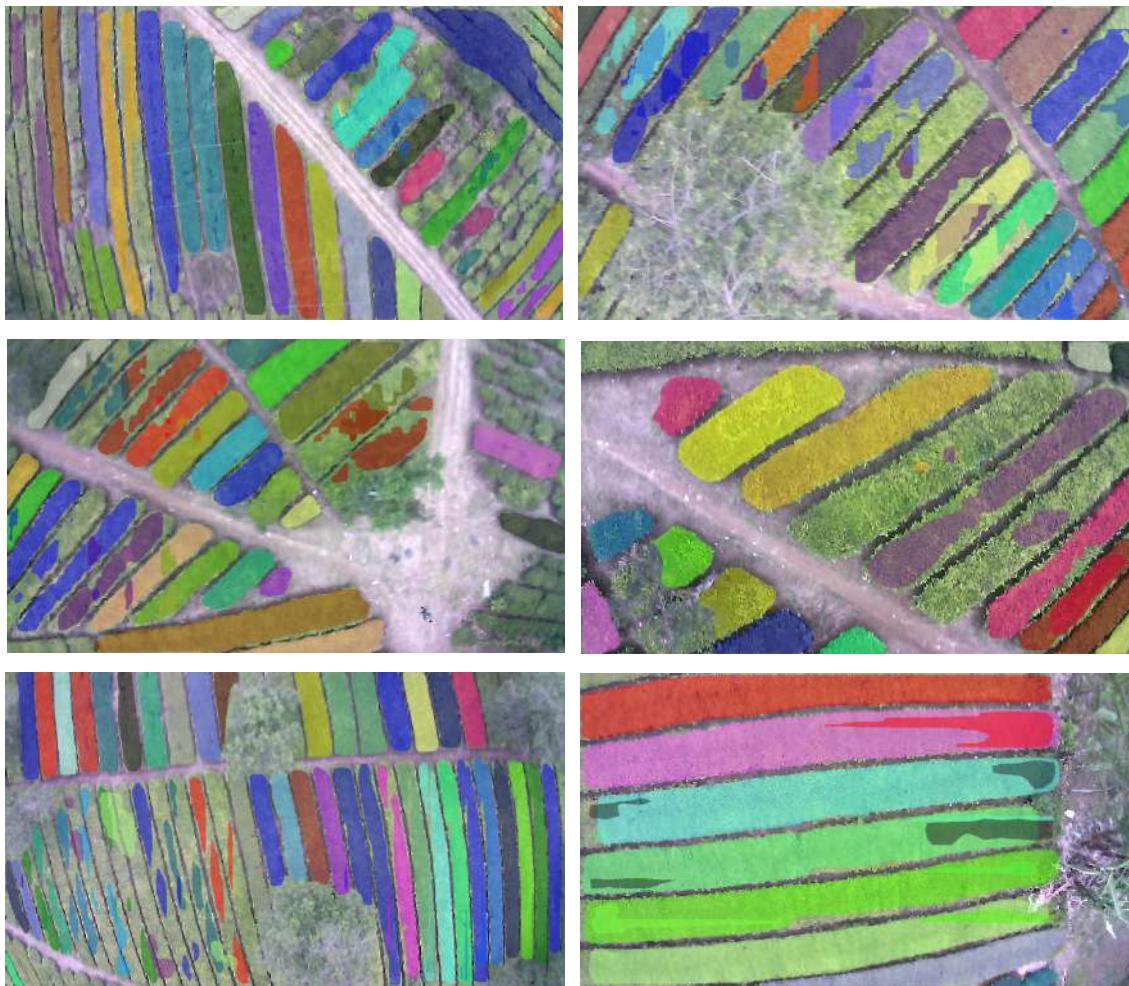
### 3. Pelatihan Model

Pelatihan model dilakukan berdasarkan konfigurasi yang telah diatur sebelumnya. Pelatihan dilakukan dengan beberapa konfigurasi yang disesuaikan antara lain adalah konfigurasi *pre-trained model* dan jumlah iterasi yang digunakan. Pada konfigurasi *pre-trained model* terdapat dua model yang digunakan untuk melakukan pelatihan yakni ‘*mask\_rcnn\_R\_101\_FPN\_3x*’ dan ‘*mask\_rcnn\_R\_101\_FPN\_3x*’. Selain itu juga pelatihan dilakukan terhadap berbagai iterasi pelatihan. Adapun jumlah iterasi yang digunakan untuk pelatihan adalah sebanyak 500, 1000, dan 1500. Terkait data yang digunakan untuk pelatihan merupakan *dataset* pelatihan (*train*) yang berisikan 132 data citra beserta. Adapun total dari label yang terdapat pada 132 data citra tersebut adalah sebanyak 3512 data. Dapat dilihat pada Gambar 4.36 yang menunjukkan jumlah *instances* atau *label* terhadap objek tanaman teh dari keseluruhan data. Sedangkan untuk kategori hanya satu yakni kategori ‘*tea*’.

category	#instances
tea	3512

Gambar 4.36 Kategori dan jumlah *instances* dari data *train*.

Berdasarkan hasil pelatihan, dihasilkan model yang mampu digunakan untuk melakukan pendekripsi objek pada tanaman teh. Gambaran dari hasil pendekripsi objek tanaman teh pada citra hasil tangkapan *drone* dapat dilihat pada Gambar 4.37. Pada gambar tersebut diperlihatkan objek tanaman teh yang dideteksi dengan cara memberikan warna pada setiap objek yang terdeteksi. Adapun pemilihan warna secara random dan satu warna merepresentasikan satu objek atau *instance* dari tanaman teh.



Gambar 4.37 Contoh hasil identifikasi area tanaman teh dengan model yang telah dilatih.

#### 4.3.3 Pengembangan Model Identifikasi Kesiapan Petik Tanaman Teh

Pengembangan model identifikasi kesiapan petik dari tanaman teh melibatkan implementasi metode *Green Leaf Index* (GLI) untuk mengukur tingkat vegetasi tanaman teh yang diterapkan pada citra *drone* yang telah diambil dan juga pelatihan model menggunakan *Artificial Neural Network* dengan *Sequential Model*. Adapun proses yang dilakukan untuk pengembangan model

ini ada empat tahapan mencakup: persiapan *dataset*, konfigurasi *Sequential Model*, pelatihan model dan akhirnya adalah evaluasi model hasil pelatihan.

### 1. Persiapan *Dataset*

*Dataset* yang digunakan untuk pembangunan model identifikasi kesiapan petik tanaman teh merupakan data hasil ekstrasi *vegetation index* (VI) dengan metode *green leaf index* (GLI) yang sudah dipersiapkan sebelumnya berupa *file CSV*. *Dataset* tersebut berisikan informasi nilai-nilai hasil pengolahan terhadap GLI yang dihasilkan dari setiap lajur atau objek teh seperti *mean* atau rata-rata GLI, lalu *median* atau nilai tengah GLI, 10 data *plot* yang merepresentasikan persebaran data dari GLI dari nilai indeks 0 sampai 1 dan lainnya.

Langkah pertama setelah data siap adalah mendaftarkannya kedalam program. Pada proses ini tidak memerlukan proses konversi atau penyesuaian data, data CSV cukup dibaca menggunakan *library Pandas* untuk melakukan pemrosesan pada data CSV di Python. Berikut Gambar 4.38 memperlihatkan visualisasi data yang telah didaftarkan kedalam program dan dibaca menggunakan *library Pandas*.

df.head()																		
		status	size	mean	median	0.0	0.1	0.2	0.30000000000000004	0.4	0.5	0.6000000000000001	0.7000000000000001	0.8	0.9			
0	0	75372	0.556002	0.466897	10410	827	11007		4009	12806	3091	3157		3129	3100	23836		
1	0	1271076	0.580532	0.488581	165597	20069	177922		85548	196815	59556	61160		54702	48988	400719		
2	0	574840	0.692300	0.772549	28345	15495	34875		38314	60625	35942	42139		40856	44231	234018		
3	0	287992	0.734559	0.810073	4922	3675	8006		20926	33431	20772	23722		26179	30926	115433		
4	0	243620	0.557853	0.478739	30763	3850	33448		19264	37663	11442	12373		11500	9412	73905		

Gambar 4.38 Penampilan *dataset* yang telah didaftarkan pada program.

*Dataset* yang didaftarkan memiliki 4244 kolom dengan 13 kolom *input* (*size, mean, median*, dan seterusnya) dan satu kolom *output* yakni *status* yang berisikan 0 atau 1. Status ini menandakan kesiapan petik dari tanaman teh dengan data input GLI tersebut, 0 menandakan bahwa tanaman tidak siap petik dan 1 untuk tanaman yang siap petik. Kemudian data *input* dan *output* dipisahkan karena nanti pelatihan akan dilakukan berdasarkan data *input* data yang nantinya merepresentasikan data kesiapan petik teh (0 atau 1).

Langkah selanjutnya adalah pembagian *dataset* menjadi data untuk *training* dan data untuk *test/evaluasi*. Disini data dibagi menjadi 75% data untuk *training* dan 25% data untuk *testing*. Berikut adalah hasil dari pembagian dataset untuk *training* dan *testing* ditampilkan pada Gambar 4.39.

```
X_train: (3183, 13)
y_train: (3183,)
X_test: (1061, 13)
y_test: (1061,)
```

Gambar 4.39 Hasil pembagian *dataset* menjadi data *train* dan *test*

Pada data hasil pembagian, terdapat ‘X\_train’ dan X\_test yang merupakan data berisikan variabel independen sebanyak 3183 data dengan kolom *input* sebanyak 13 kolom. Sedangkan y\_train dan y\_test adalah variabel yang menyimpan variabel dependen, yang mewakili keluaran yang akan diprediksi sebanyak 1061 data.

## 2. Konfigurasi Model

Pada tahapan konfigurasi model, dipersiapkan model untuk nantinya difunakan untuk *training* dan juga *testing*. Pertama, model *Sequential* dibuat, yang merupakan wadah untuk membangun model secara berurutan. Lapisan-lapisan dapat ditambahkan satu per satu, dan model akan mengalir dari *input* ke *output* sesuai dengan urutan penambahan lapisan.

Lapisan pertama adalah lapisan *Dense* dengan 16 unit neuron. Lapisan *Dense* berarti setiap neuron pada lapisan ini terhubung ke setiap neuron pada lapisan sebelumnya. *Activation function* yang digunakan adalah ReLU (*Rectified Linear Unit*), yang merupakan fungsi aktivasi umum yang mengonversi nilai negatif menjadi nol dan mempertahankan nilai positif.

Lapisan kedua adalah lapisan *Dense* lagi, namun hanya memiliki 1 unit neuron. *Activation function* yang digunakan pada lapisan ini adalah *sigmoid*. Fungsi *sigmoid* menghasilkan *output* dalam rentang 0 hingga 1, yang cocok untuk tugas klasifikasi biner. Ini mengindikasikan probabilitas kelas positif.

*Optimizer* yang digunakan untuk melatih model adalah Adam (*Adaptive Moment Estimation*). Adam adalah metode optimasi stokastik yang efisien dan populer. Laju pembelajaran (*learning rate*) untuk *optimizer* diatur sebesar 0.001. Fungsi utama Adam adalah mengoptimalkan nilai bobot (*weights*) dan bias dalam model agar dapat meminimalkan fungsi kerugian (*loss function*) selama pelatihan.

Kemudian, model dikompilasi dengan menggunakan ‘*binary\_crossentropy*’ sebagai fungsi kerugian. Fungsi ini cocok untuk tugas klasifikasi biner dan mengukur seberapa baik model memprediksi kelas target. Selain itu, metrik evaluasi yang digunakan adalah akurasi (*accuracy*),

yang memberikan informasi tentang sejauh mana model berhasil mengklasifikasikan data dengan benar.

### 3. Pelatihan Model

Pelatihan dari model identifikasi kesiapan petik teh dilakukan sebanyak 100 *epochs*, hal tersebut ditentukan melihat data latih yang cukup banyak. Parameter *epochs* menentukan berapa kali model akan melakukan iterasi terhadap seluruh dataset pelatihan. Setiap kali model melalui seluruh dataset disebut sebagai satu *epoch*. Dalam hal ini, berarti model akan melihat seluruh dataset pelatihan sebanyak 100 kali selama proses pelatihan. Dengan kata lain, model akan belajar dari data pelatihan sebanyak 100 putaran. Ada 3183 data latih yang akan digunakan dalam pelatihan dengan kolom *input* sebanyak 13 data.

Setelah pelatihan model dilakukan, model bisa digunakan untuk melakukan prediksi. Untuk mencoba menggunakan model terhadap data baru, maka dibutuhkan data dengan struktur seperti data *input* yang digunakan untuk pelatihan model. Data tersebut merupakan data ekstraksi informasi VI dengan GLI yang berisikan *size*, *mean*, *median*, dan persebaran data atau *plotting* dari 0 hingga 1 sebanyak sepuluh kolom. Berikut data yang disiapkan untuk melakukan prediksi menggunakan model hasil pelatihan dapat dilihat Gambar 4.40. Lalu berdasarkan data tersebut, dilakukan prediksi dengan model hasil pelatihan. Gambar 4.41 memperlihatkan hasil dari prediksi yang dilakukan.

```
input_data = np.array([
[49044, 0.764383986007276, 0.8352941176470591, 162, 244, 421, 2185, 6793, 3936, 3886, 4884, 6870, 19663],
[34460, 0.47829496579622593, 0.40784313725490196, 6278, 432, 6503, 2168, 6396, 1271, 1129, 928, 490, 8865],
[25108, 0.7557630252993343, 0.8196078431372551, 77, 140, 199, 1218, 3725, 2047, 1968, 2497, 3622, 9615],
[99392, 0.49128050256439587, 0.40784313725490196, 17161, 1363, 17817, 6956, 17236, 4300, 3637, 2835, 1725, 26362],
[73500, 0.7073814153112232, 0.7568627450980394, 1477, 973, 2680, 7749, 8471, 5074, 6247, 7001, 6494, 27334],
[51676, 0.5437900283013206, 0.44921184159938493, 6940, 847, 7431, 4130, 8062, 2603, 2616, 2273, 1805, 14969],
[25824, 0.7906209817308851, 0.8695886197616303, 86, 94, 169, 734, 2895, 2074, 2257, 2384, 3513, 11618],
[29312, 0.7968171619833824, 0.8831218762014611, 69, 176, 281, 774, 2888, 2527, 2344, 2525, 3882, 13846],
[105096, 0.5407396593618424, 0.44921184159938493, 13535, 2192, 14809, 10227, 15237, 5729, 5918, 4789, 3446, 29214],
[55444, 0.554392257715777, 0.4885813148788929, 5759, 1342, 6623, 7644, 6933, 3658, 3683, 3420, 1812, 14570],
[31260, 0.626816810700545, 0.6362168396770479, 2012, 450, 2377, 3560, 4390, 2043, 2221, 2562, 2289, 9356],
[98736, 0.47589072177046793, 0.40784313725490196, 17189, 1034, 17465, 4672, 17200, 3164, 2531, 2077, 1324, 24080],
[18372, 0.6798713464651305, 0.7176470588235296, 150, 96, 306, 1352, 1674, 660, 758, 1273, 1892, 3011],
[676, 0.48085579283343466, 0.40784313725490196, 115, 13, 122, 73, 91, 35, 37, 15, 6, 169],
[54340, 0.728105537393463, 0.7882352941176473, 640, 487, 1227, 4362, 7330, 4125, 4434, 5328, 6171, 20236],
[48828, 0.4957620653492726, 0.40784313725490196, 8355, 813, 8584, 3386, 8124, 2231, 1903, 1400, 900, 13132],
[188680, 0.5038840183067098, 0.40784313725490196, 18019, 1839, 18782, 6926, 18825, 4265, 4005, 3475, 2416, 30128],
[36, 0.4624118928617198, 0.39219530949634757, 7, 0, 7, 4, 6, 1, 1, 1, 0, 9],
[12036, 0.7977383369563231, 0.9211072664359863, 158, 319, 205, 283, 732, 914, 1042, 898, 1062, 6423],
[15188, 0.59388158987879763, 0.5702422145328719, 1411, 292, 1480, 1698, 2018, 1040, 915, 1019, 740, 4575],
[71876, 0.6133409563555021, 0.6103037293348712, 4550, 1648, 5978, 10105, 8370, 4835, 5740, 5399, 3939, 21312],
[19036, 0.6423091067096489, 0.6558246828143022, 685, 363, 1107, 2966, 2384, 1216, 1530, 1894, 1209, 5682]
])
```

Gambar 4.40 Gambaran data yang digunakan untuk melakukan prediksi.

```

tf.Tensor(
[0.9999999  1.        1.        1.        0.9999994  0.9999987
 0.9999989  0.99979055 1.        1.        0.9999999  1.
 1.        1.        1.        1.        1.        1.
 1.        1.        1.        1.        1., shape=(22,), dtype=float32)

```

Gambar 4.41 Contoh hasil prediksi kesiapan petik tanaman teh.

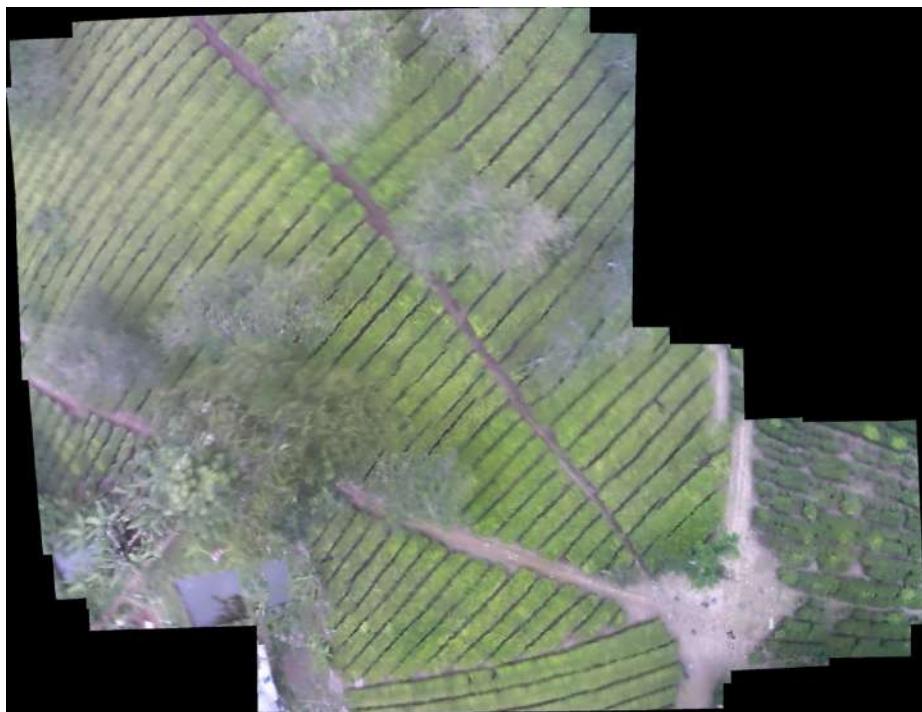
Pada Gambar 4.41 diperlihatkan hasil dari prediksi data yang dihasilkan dari VI dengan GLI. Hasil dari prediksi akan menampilkan nilai dengan *range* 0 hingga 1 yang dimana semakin mendekati 0 maka data tersebut diprediksi tidak siap petik dan semakin dekat ke 1 maka data tersebut diprediksi siap petik. Terlihat pada Gambar 4.41 nilai yang dihasilkan diatas 0.9 yang menunjukkan bahwa data terprediksi menjadi tanaman teh yang siap petik.

#### 4.4 Implementasi dan Integrasi Setiap Model

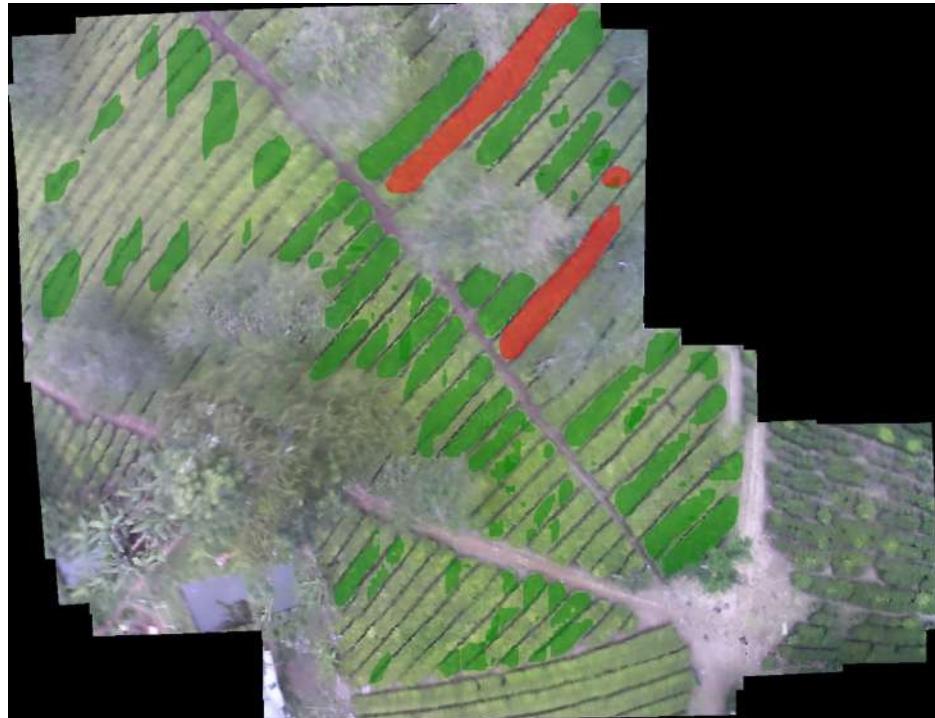
Tahapan implementasi berfokus pada integrasi model-model yang sebelumnya telah dibangun sebelumnya. Diawali dengan model penggabungan citra yang menghasilkan citra area tanaman teh yang lebih luas hasil gabungan citra. Selanjutnya keluaran dari model penggabungan citra dijadikan masukan pada model pemetaan tanaman teh. Model pemetaan tanaman teh menghasilkan hasil prediksi dari objek tanaman teh berdasarkan citra area tanaman teh yang luas dan menghasilkan data berupa batasan-batasan *polygon* dari objek tanaman teh yang ada pada citra tersebut. Pada akhirnya batasan tersebut digunakan untuk mengidentifikasi tingkat kesiapan petik dari objek tanaman teh tersebut dengan mengekstraksi informasi VI dengan metode GLI dan memprediksi tingkat kesiapannya menggunakan model yang telah dilatih. Pada akhirnya model akan menampilkan tanaman teh yang siap petik dengan warna hijau dan yang tidak siap petik dengan warna merah

Hasil dari proses penggabungan citra menghasilkan 9 citra area perkebunan teh yang lebih luas terdiri dari 6 citra untuk area siap petik dan 3 citra area tidak siap petik. Sembilan data tersebut yang akan menjadi data untuk dilakukan pemetaan dan identifikasi kesiapan petik teh. Berikut hasil dari masing-masing citra yang dilakukan pemetaan dan identifikasi kesiapan petik tanaman teh yang terdapat didalam masing-masing citra.

1. Citra Siap Petik 1



Gambar 4.42 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 1’.



Gambar 4.43 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 1’.

2. Citra Siap Petik 2

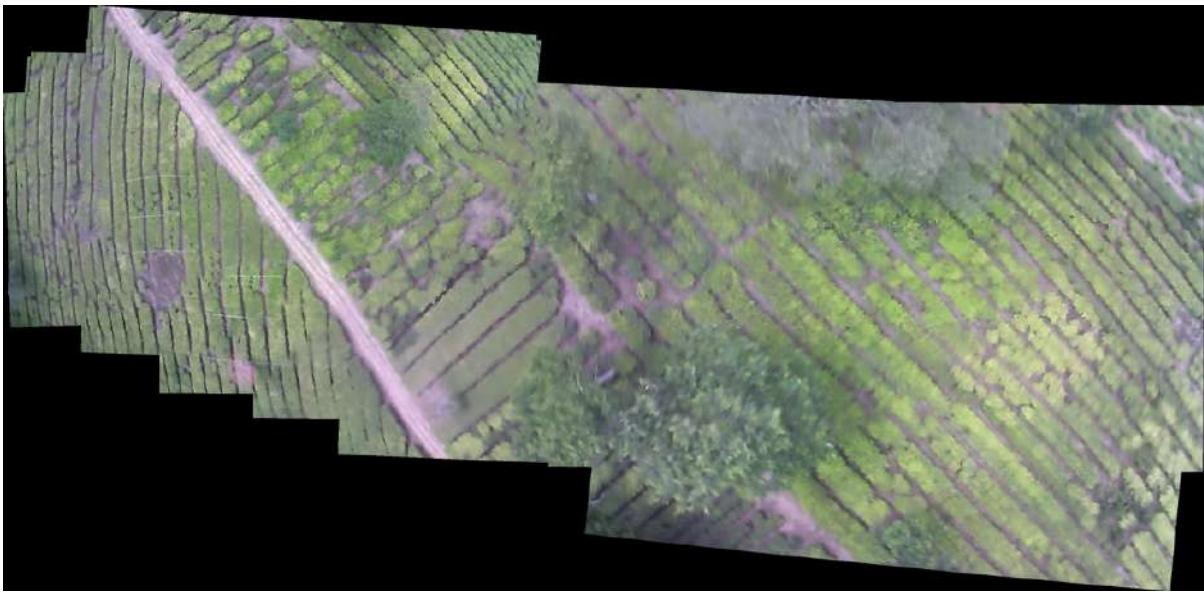


Gambar 4.44 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’.

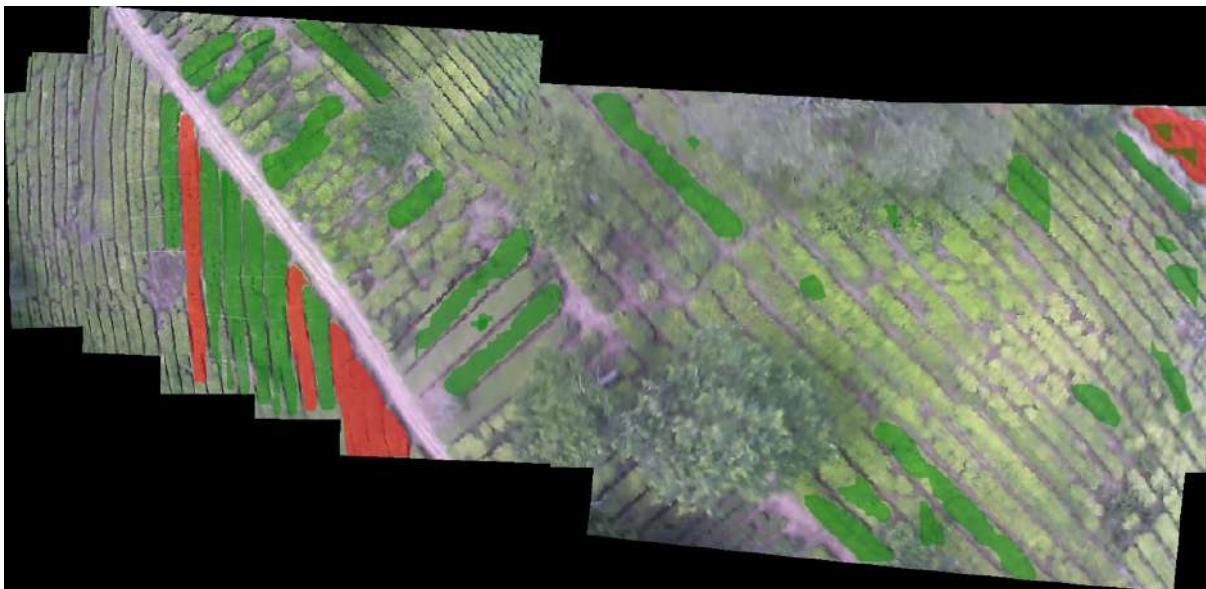


Gambar 4.45 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 2’.

3. Citra Siap Petik 3



Gambar 4.46 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’.

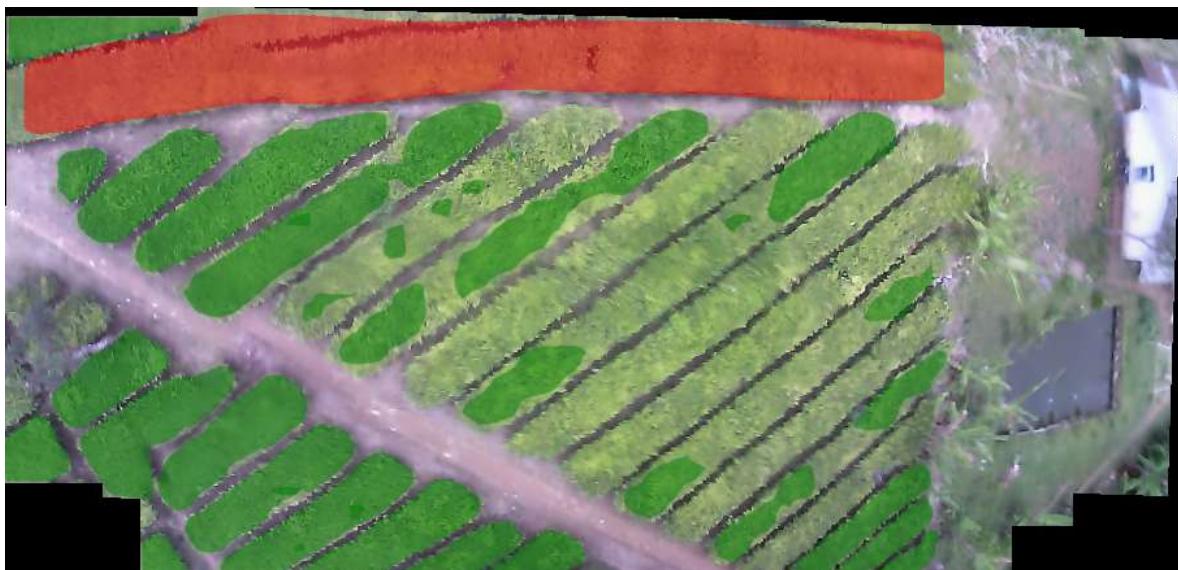


Gambar 4.47 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 3’.

4. Citra Siap Petik 4



Gambar 4.48 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 4’.

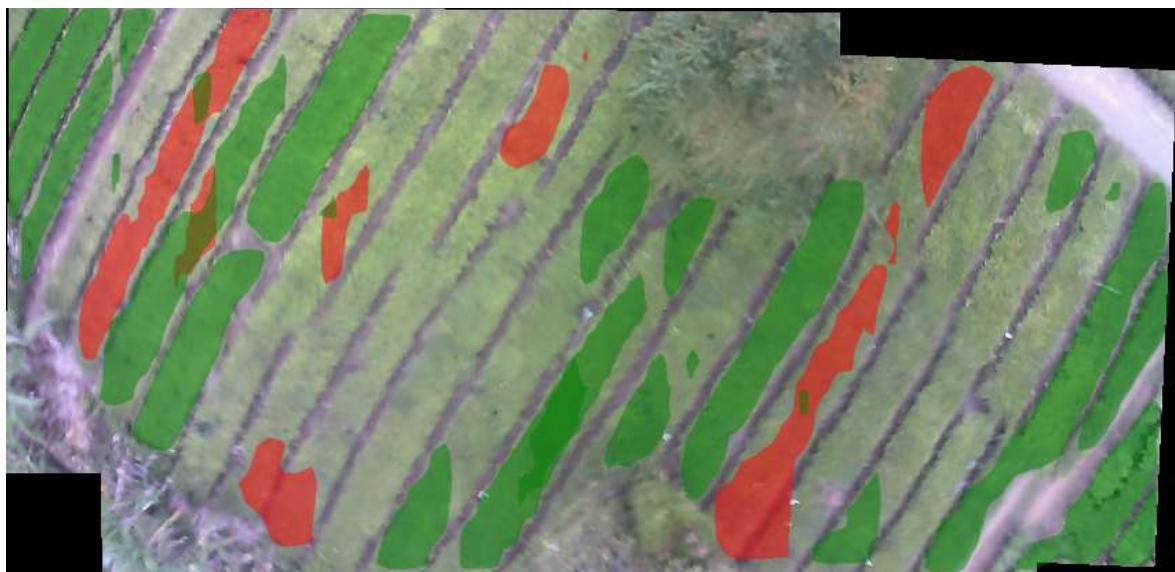


Gambar 4.49 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 4’.

5. Citra Siap Petik 5



Gambar 4.50 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 5’.



Gambar 4.51 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 5’.

6. Citra Siap Petik 6

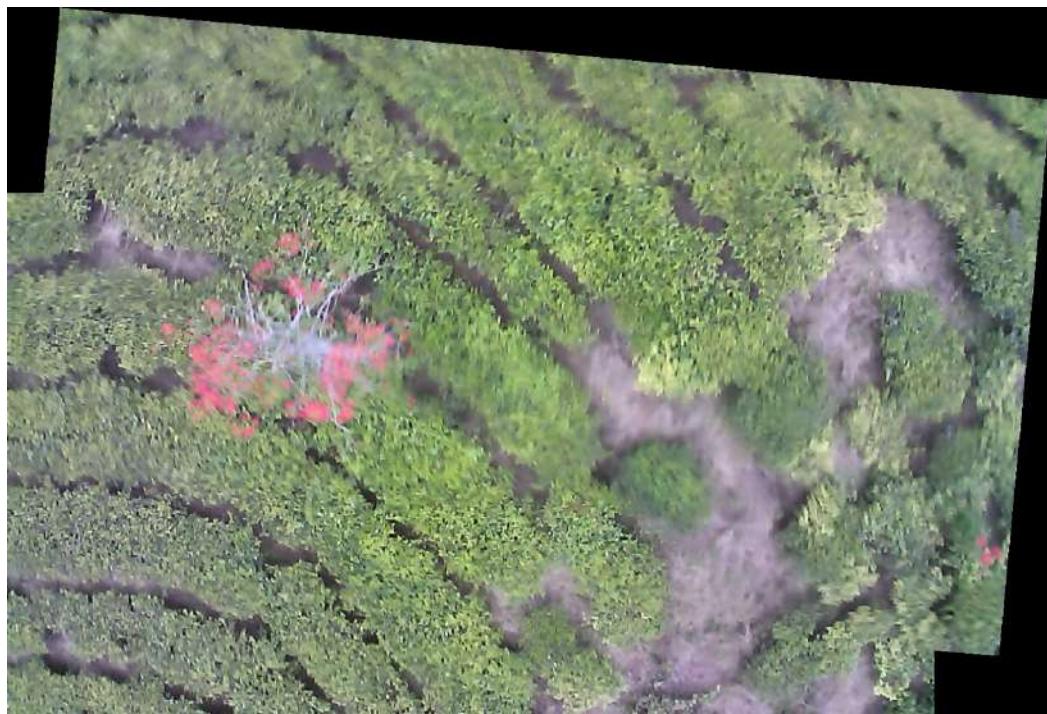


Gambar 4.52 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 6’.

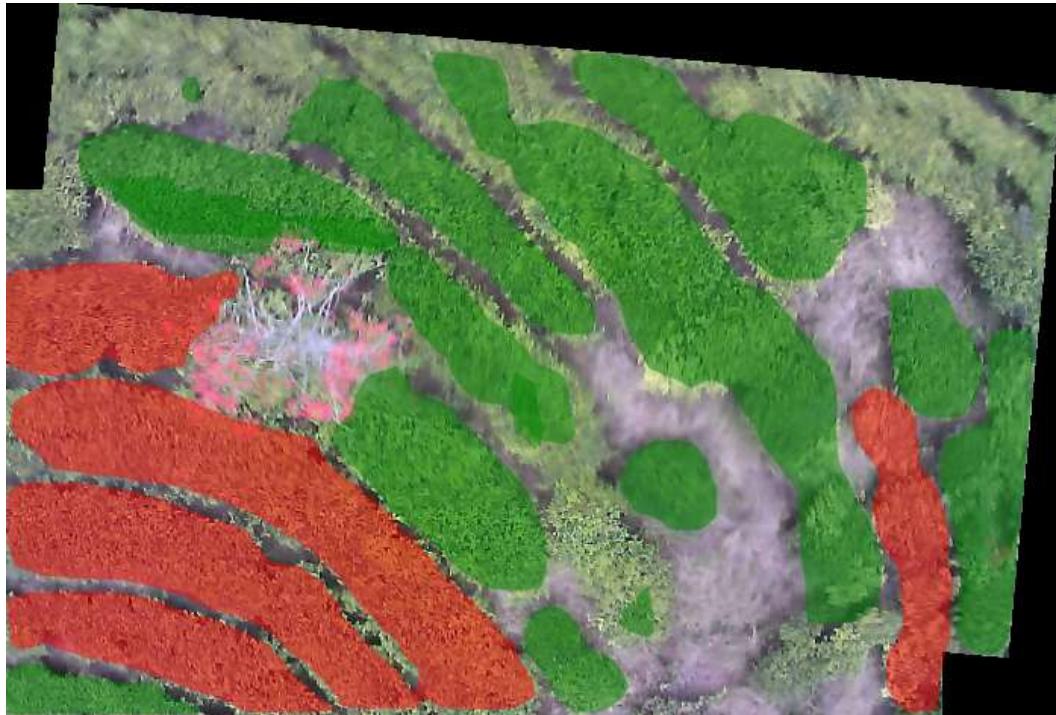


Gambar 4.53 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 6’.

7. Citra Tidak Siap Petik 1



Gambar 4.54 Hasil penggabungan citra pada kelompok citra ‘Tidak Siap Petik 1’.

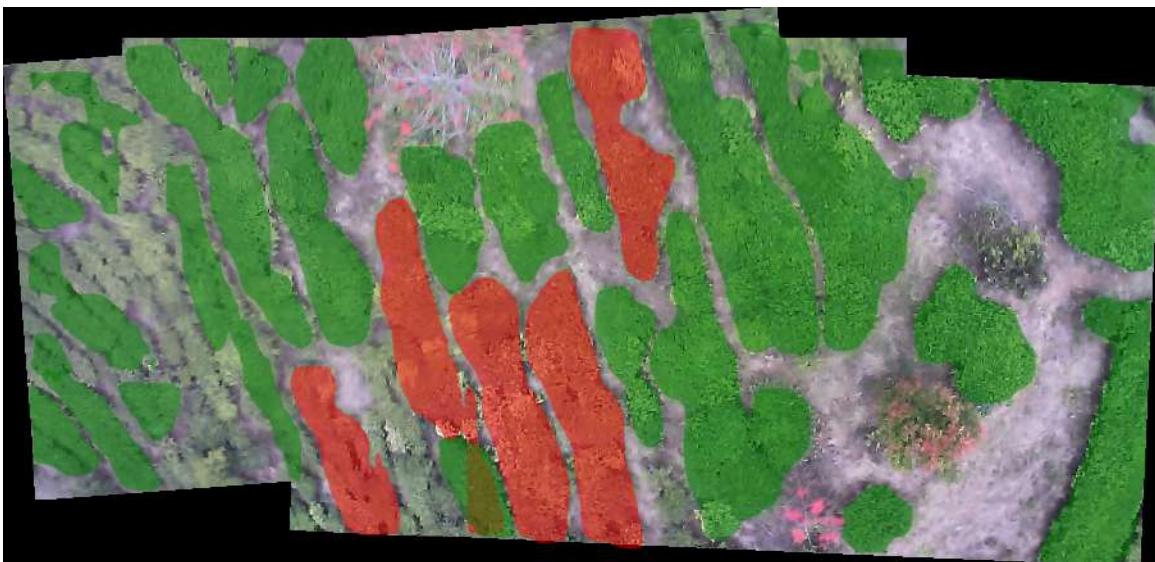


Gambar 4.55 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Tidak Siap Petik 1’.

8. Citra Tidak Siap Petik 2

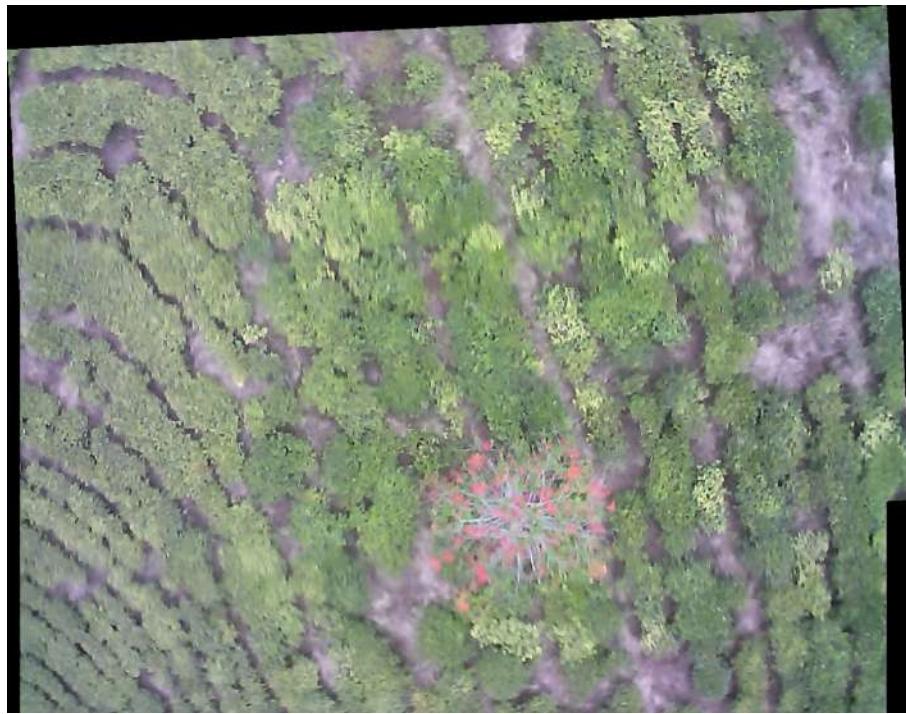


Gambar 4.56 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 2’.

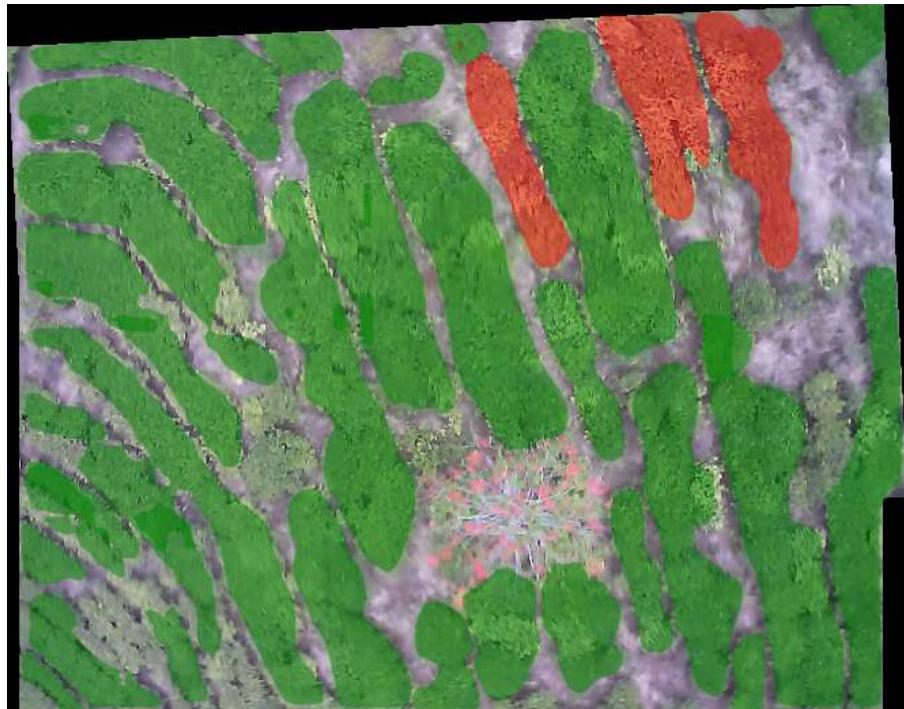


Gambar 4.57 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 2’.

9. Citra Tidak Siap Petik 3



Gambar 4.58 Hasil penggabungan citra pada kelompok citra ‘Siap Petik 3’.



Gambar 4.59 Hasil pemetaan dan identifikasi kesiapan petik tanaman teh ‘Siap Petik 3’.

## 4.5 Analisis dan Evaluasi Hasil

### 4.5.1 Evaluasi Hasil Penggabungan Citra

Berdasarkan hasil dari proses penggabungan citra terhadap kelompok gambar yang telah siapkan, dilakukan proses evaluasi terhadap hasil dari penggabungan citra yang dilakukan. Adapun evaluasi yang dilakukan adalah perhitungan akurasi dari proses pencocokan fitur dengan rumus sebagai berikut:

$$Accuracy = \frac{Ni}{Nm} \times 100\%$$

Dilihat pada formula diatas,  $Ni$  merupakan *number of inliers* atau total fitur yang termasuk kedalam fitur *inliers* berdasarkan perhitungan menggunakan RANSAC. Kemudian  $Nm$  merupakan *number of matches* atau total dari keseluruhan fitur yang cocok. Selain itu juga dilakukan perhitungan *Structural Similarity Index* (SSIM). SSIM adalah metrik evaluasi yang digunakan untuk mengukur sejauh mana dua gambar mirip dalam struktur mereka. SSIM mempertimbangkan faktor-faktor seperti intensitas *luminance*, kontras, dan struktur. Perhitungan SSIM melibatkan tiga komponen utama, yaitu penyesuaian untuk *luminance*, kontras, dan struktur, dan nilai SSIM berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan kesamaan sempurna. Implementasi SSIM menggunakan *library Skimage* di Python. Adapun formula yang digunakan adalah sebagai berikut:

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$$

Dari formula diatas, masing-masing  $l$  merupakan *luminance*,  $c$  merupakan *contrast* dan  $s$  merupakan *structure*. Masing-masing dihitung dengan formula berikut:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$
$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$
$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Berdasarkan formula diatas,  $\mu_x$  dan  $\mu_y$  adalah nilai rata-rata dari gambar  $x$  dan  $y$  secara berturut-turut, dan  $C_1$  adalah konstanta kecil untuk mencegah pembagian dengan nol. Sedangkan  $\sigma_x$  dan  $\sigma_y$  adalah deviasi standar dari gambar  $x$  dan  $y$  secara berturut-turut, dan  $C_2$  adalah konstanta kecil. Lalu  $\sigma_{xy}$  adalah kovarians antara  $x$  dan  $y$ , dan  $C_3$  adalah konstanta kecil. Berikut

adalah hasil dari perhitungan terhadap akurasi dan SSIM terhadap kelompok gambar yang dilakukan proses penggabungan citra:

Tabel 4.2 Hasil perhitungan tingkat akurasi dan SSIM.

No	Kelompok	Nm	Ni	Accuracy	Average SSIM	Time
1	Siap Petik 1	6052	3007	49,6860542	0,563798652	119,00 s
2	Siap Petik 2	1769	754	42,62295082	0,607479691	5,20 s
3	Siap Petik 3	4666	2193	46,99957137	0,44378525	27,00 s
4	Siap Petik 4	2237	1031	46,0885114	0,603727341	4,10 s
5	Siap Petik 5	684	339	49,56140351	0,677938687	2,20 s
6	Siap Petik 6	1001	466	46,5534466	0,71468813	2,80 s
7	Tidak Siap Petik 1	356	142	39,88764045	0,53352	1,20 s
8	Tidak Siap Petik 2	333	141	42,34234234	0,616391439	2,20 s
9	Tidak Siap Petik 3	702	259	36,89458689	0,56517541	2,20 s
<i>Average</i>				44,5151675	0,59183384	

Berdasarkan Tabel 4.2 diatas diperlihatkan hasil dari akurasi dan juga rata-rata dari SSIM yang telah dikalkulasi berdasarkan kelompok citra yang dipersiapkan. *Number of matches* (Nm) merupakan total dari keseluruhan fitur yang cocok pada setiap pasangan gambar yang akan digabung. Begitu pula *number of inliers* (Ni) yang merupakan total dari keseluruhan *inliers* yang termasuk kedalam fitur yang akan digunakan untuk penggabungan berdasarkan perhitungan RANSAC. Diperlihatkan bahwa rata-rata dari akurasi dari seluruh proses penggabungan adalah 44,51% yang berarti model penggabungan citra memiliki kemampuan akurasi untuk menentukan *key points* yang dapat digunakan untuk penggabungan citra sebesar 44,51%.

Selain itu dilakukan perhitungan rata-rata dari SSIM yang dihasilkan dari perhitungan SSIM setiap citra pada citra yang telah dilakukan transformasi, kemudian ditentukan sejauh mana kedua citra yakni citra asli dengan citra hasil transformasi untuk dilakukan penggabungan citra tersebut memiliki kemiripan secara struktur dan dihitung rata-ratanya. Berdasarkan Tabel 4.2 rata-rata dari keseluruhan *average SSIM* adalah 0,59 dari skala 0-1. Dengan nilai SSIM tersebut menandakan

bahwa struktur citra hasil penggabungan citra memiliki kemiripan struktur sebesar 0.59 dengan citra asli atau citra sumber.

#### 4.5.2 Evaluasi Hasil Pemetaan Area Tanaman Teh

Berdasarkan hasil dari proses pelatihan yang dilakukan berikut hasil dari model yang dilatih pada konfigurasi *backbone* dan jumlah iterasi yang berbeda. Evaluasi dilakukan terhadap *dataset* untuk evaluasi yang telah disiapkan sebelumnya dengan total ada 25 data citra beserta *label*. Gambar 4.60 berikut memperlihatkan total dari *instance* atau label objek teh pada keseluruhan gambar ada sebanyak 732 *instance* dari 25 data citra.

category	#instances
tea	732

Gambar 4.60 Kategori dan jumlah *instances* dari data *evaluation*.

Berdasarkan data tersebut, evaluasi dilakukan terhadap model yang telah dilatih. Tabel 4.3 menunjukkan lamanya waktu pelatihan pada masing masing pelatihan yang dilakukan.

Tabel 4.3 *Training time* dari pelatihan model.

No	Backbone	Iterasi	Training Time
1	ResNet-50, FPN	500	0:05:47
2	ResNet-50, FPN	1000	0:11:54
3	ResNet-50, FPN	1500	0:18:12
4	ResNet-101, FPN	500	0:07:48
5	ResNet-101, FPN	1000	0:15:42
6	ResNet-101, FPN	1500	0:24:48

Terlihat pada Tabel 4.3 bahwa waktu pelatihan dipengaruhi oleh jenis *backbone* dan juga total iterasi yang digunakan. *Backbone* dengan ResNet-101 memiliki waktu pelatihan yang lebih lama dibanding dengan ResNet-50, yang dimana masuk akal karena ResNet-101 memiliki lapisan lebih banyak dan juga kompleks yakni 101 lapisan sedangkan ResNet-50 memiliki 50 lapisan. Begitu pula kasusnya dengan iterasi, yang mana semakin banyak iterasi yang digunakan maka semakin

lama juga total waktu pelatihan atau *training time* yang dibutuhkan. Selain waktu pelatihan, terdapat pula hasil dari pelatihan model yang telah dilakukan. Pada bagian ini diperlihatkan seberapa presisi model yang dilatih untuk melakukan pemetaan terhadap objek tanaman teh. Berikut Tabel 4.4 memperlihatkan hasil dari setiap pelatihan model yang dilakukan.

Tabel 4.4 Tingkat kepresisan model hasil pelatihan.

No	<i>Backbone</i>	Iterasi	AP	AP50	AP75	APs	APm	API
1	ResNet-50, FPN	500	26.815	46.326	28.093	3.003	28.772	26.969
2	ResNet-50, FPN	1000	33.069	56.040	33.937	3.812	38.331	30.916
3	ResNet-50, FPN	1500	33.819	55.456	34.986	6.931	39.588	31.826
4	ResNet-101, FPN	500	33.239	56.410	35.078	8.317	33.314	34.501
5	ResNet-101, FPN	1000	35.086	58.524	37.809	11.782	36.627	35.065
6	ResNet-101, FPN	1500	39.852	63.966	43.668	23.264	43.855	38.167

Dapat diperlihatkan bahwa pada Tabel 4.4 hasil *average precision* (AP) dari setiap pelatihan yang dilakukan. *Average precision* dihasilkan dari perhitungan IoU (*Intersection over Union*) pada setiap hasil prediksi dengan membandingkan seberapa baik prediksi tersebut bertepatan dengan *ground truth* yang merupakan area prediksi ditambah dengan area sebenarnya. Berikut formula yang digunakan untuk menghitung IoU:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Berdasarkan formula tersebut, *Area of Overlap* merupakan luasan area yang saling tumpang tindih atau beririsan antara area hasil prediksi dan area sebenarnya (*ground truth*). Sedangkan *Area of Union* merupakan luasan keseluruhan atau gabungan dari area hasil deteksi dengan area sebenarnya (*ground truth*). Hasil prediksi kemudian diurutkan berdasarkan *threshold* skor keyakinan yang dimana disini skor yang digunakan adalah 50-95, 50, 75 dengan jenis area *small*, *medium*, dan *large*. Setelah itu pemotongan dilakukan untuk mempertahankan sejumlah prediksi terbaik berdasarkan *threshold* skor. lalu *precision* dan *recall* dihitung pada setiap langkah pemotongan untuk memahami sejauh mana prediksi model akurat dan seberapa baik model dapat mendekripsi objek. Berikut formula untuk menghitung *precision* dan *recall*:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Berdasarkan formula tersebut, TP merupakan *True Positive* yang mana dihasilkan ketika model secara benar memprediksi nilai positif. FP merupakan *False Positive* yang merupakan hasil dimana ketika model memprediksi positif padahal seharusnya adalah negatif. Sedangkan FN merupakan *False Negative* yang merupakan kebalikan dari FP dimana model memprediksi negatif namun seharusnya positif. Dengan menggunakan nilai *precision* dan *recall* pada setiap pemotongan, *precision-recall curve* dibuat. Area di bawah kurva ini dihitung untuk mendapatkan *Average Precision (AP)*, yang memberikan gambaran menyeluruh tentang kualitas deteksi model. Selain itu, ada juga metrik tambahan seperti AP50, AP75, APs, APm, dan API dihitung untuk mengevaluasi performa model pada tingkat kepercayaan/ *threshold* dan ukuran tertentu. Proses ini memberikan pemahaman menyeluruh tentang seberapa baik model dapat mengidentifikasi objek dalam berbagai kondisi, memungkinkan evaluasi yang holistik terhadap kinerja deteksi objek yang dilakukan.

Pelatihan menggunakan *backbone* ResNet-50 menunjukkan bahwa AP tertinggi didapat dari pelatihan dengan iterasi sebanyak 1500, namun peningkatan AP tidak terlalu signifikan dari penggunaan iterasi 1000. AP50 merupakan *average precision* dengan *Intersection over Union* (IoU) setidaknya 50%, begitu juga dengan AP75. Pada Tabel 4.2 terjadi penurunan hasil AP50 pada pelatihan dengan *backbone* ResNet-50 yang dimana pada iterasi 1000 mendapatkan AP50 sekitar 56% namun pada iterasi 1500 menjadi 55%. Walaupun penurunan tidak terlalu besar, ini menandakan bahwa model mengalami penurunan dalam melakukan pendekripsi objek dengan IoU setidaknya 50%. Namun jika dilihat dari APs (*small*), model tersebut memiliki APs yang lebih tinggi dibanding penggunaan iterasi yang lebih sedikit. Namun tidak dapat dipungkiri, rendahnya APs, menandakan bahwa model tidak dapat melakukan deteksi secara baik pada objek yang ukurannya kecil dan lebih presisi untuk pendekripsi objek dengan ukuran sedang atau *medium* (APm) dan ukuran besar atau *large* (API).

Hal yang serupa juga terjadi pada pelatihan model yang menggunakan *backbone* ResNet-101. Terjadi peningkatan AP dari setiap peningkatan iterasi. walaupun tidak secara signifikan, namun ini menandakan bahwa peningkatan iterasi dapat membantu meningkatkan tingkat presisi yang dihasilkan. Dibandingkan dengan pelatihan dengan *backbone* ResNet-50 yang AP terbesarnya adalah 33%, pada model dengan pelatihan menggunakan *backbone* ResNet-101, didapatkan AP

terbesarnya adalah sekitar 39% dengan iterasi sebanyak 1500 iterasi. Model hasil pelatihan tersebut juga menjadi model yang memiliki tingkat presisi tertinggi mulai dari AP, AP50, AP75, APs, APm hingga API dibandingkan dengan semua hasil pelatihan lainnya.

#### 4.5.3 Evaluasi Hasil Identifikasi Kesiapan Petik Tanaman Teh

Proses selanjutnya adalah tahapan evaluasi dari model hasil pelatihan sebelumnya. Evaluasi atau *testing* ini menggunakan *dataset test* yang sudah disiapkan berdasarkan pembagian *dataset train* dan *test* sebelumnya. Sebanyak 1061 data yang disediakan untuk melakukan proses *test*. Kemudian dilakukan evaluasi untuk akurasi dari model yang telah dilatih dan hasilnya dapat dilihat pada Gambar 4.61.

```
34/34 [=====] - 0s 2ms/step - loss: 0.4202 - accuracy: 0.8266
[0.4202022850513458, 0.826578676700592]
Loss = 0.4202022850513458
Accuracy = 0.826578676700592
```

Gambar 4.61 Hasil evaluasi *loss* dan *accuracy* dari model yang dilatih.

Berdasarkan Gambar 4.61, akurasi dari model yang dilatih mencapai 82%, hal tersebut menandakan bahwa model berhasil mengklasifikasikan dengan benar sekitar 82.66% dari sampel evaluasi yakni data *test*. Kemudian dilakukan juga perhitungan menggunakan fungsi *root mean square error* (RMSE) dengan formula sebagai berikut:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (predicted_i - actual_i)^2}$$

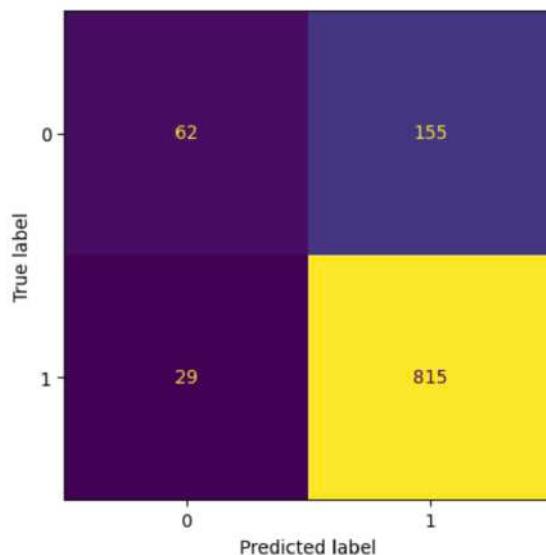
Berdasarkan formula tersebut, N merupakan jumlah *dataset testing* sedangkan *predicted* dan *actual* adalah hasil prediksi dan nilai sebenarnya. Dihasilkan nilai *loss* sebesar 0.4202 yang merupakan nilai dari fungsi RMSE pada *dataset test*. Secara umum, nilai *loss* yang lebih rendah menunjukkan kinerja yang lebih baik. Oleh karena itu, nilai *loss* sekitar 0.4202 bisa dianggap sudah cukup baik.

Selanjutnya dilakukan evaluasi berdasarkan *confusion matrix* untuk melihat performa model yang telah dilatih. Matriks ini memperbaiki gambaran tentang seberapa baik model dapat memprediksi kelas target. Matriks ini terdiri dari empat sel atau bagian utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Setiap sel dalam matriks

memberikan informasi spesifik tentang hasil prediksi model. Berikut ditampilkan gambaran dari *confusion matrix* pada Gambar 4.62 dan hasil dari *confusion matrix* pada Gambar 4.63.

		Predicted Negative (0)	Predicted Positive (1)
Actual Negative (0)	True Negative		False Positive
Actual Positive (1)	False Negative	True Positive	

Gambar 4.62 Gambaran dari *confusion matrix*.



Gambar 4.63 *Confusion matrix* dari hasil prediksi pada data *test*.

Terlihat pada Gambar 4.63 persebaran hasil prediksi pada data *test*. Sebanyak 62 data yang ber-label tidak siap petik (*False*) dan berhasil terprediksi tidak siap petik (*False*) juga. Kemudian 155 data hasil prediksi menyatakan *True* namun sebenarnya memiliki label *False*. Selanjutnya 29 data dengan label *True* terprediksi *False* dan 815 data terprediksi *True* dan memiliki label *True*.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan hasil serta analisis dan evaluasi yang dilakukan pada penelitian ini, didapatkan kesimpulan sebagai berikut:

1. Data citra area kebun teh yang lebih luas berdasarkan hasil tangkapan *drone* dapat diperoleh menggunakan metode *image stitching* melalui beberapa tahapan proses pengolahan citra yakni tahapan *pre-processing* yang merupakan tahapan guna untuk memperingan komputasi, kemudian *finding features* untuk ekstraksi fitur pada citra, lalu *matching feature* yang beruna untuk mencocokkan fitur pada suatu citra dengan dengan citra lain sehingga mengecualikan citra yang tidak relevan, kemudian *camera estimation, adjustment and correction* yang bertujuan untuk mengatur parameter kamera terhadap fitur yang telah dicocokan, selanjutnya *warping image* untuk mentransformasi citra sehingga lebih cocok ketika digabung dengan citra pasangannya, setelah itu *seam mask* untuk menemukan garis jahitan atau garis transisi antara satu citra dan citra pasangannya hingga pada akhirnya *blending* untuk menghasilkan citra akhir berdasarkan parameter yang sudah ditentukan oleh proses-proses pebelumnya. Metode *image stitching* berhasil diimplementasikan dengan menggunakan OpenCV sebagai *library* pengolahan citra dengan bahasa pemrograman Python dengan akurasi dari pencocokan fitur sebesar 44,51% dan rata-rata dari *average SSIM* bernilai 0,59.
2. Pemetaan tanaman teh berdasarkan citra area perkebunan teh hasil tangkapan *drone* dapat dihasilkan menggunakan metode *deep learning, Mask Region-based Convolutional Neural Network* (Mask R-CNN) dengan melalui proses pelatihan atau *training* model yang terkonfigurasi. Pelatihan dilakukan pelatihan dengan *dataset* citra area tanaman teh yang telah dilakukan anotasi atau *labeling* terhadap objek tanaman teh. Metode pemetaan tanaman teh berhasil diimplementasikan dengan menggunakan Detectron2 sebagai *framework* dengan Bahasa pemrograman Python. Hasil dari pelatihan model pemetaan tanaman teh memiliki rata-rata tingkat presisi dalam identifikasi tanaman teh tertinggi di 39% yang merupakan hasil pelatihan menggunakan *backbone* ResNet-101, FPN dan jumlah iterasi sebanyak 1500 iterasi.
3. Identifikasi tingkat kehijauan teh dapat dihasilkan menggunakan metode *vegetation index* (VI), *Green Leaf Index* (GLI) terhadap citra area perkebunan teh hasil tangkapan *drone*

melalui proses kalkulasi terhadap setiap piksel RGB menggunakan formula GLI yang kemudian dilakukan normalisasi terhadap nilai hasil sehingga memiliki skala indeks dari 0 hingga 1. Semakin indeks mendekati 0 menunjukkan bahwa tingkat kehijauan dari tanaman teh rendah sedangkan semakin mendekati 1 maka tingkat kehijauan tanaman teh semakin tinggi. Selanjutnya kesiapan petik teh dapat ditentukan melalui pelatihan model untuk melakukan identifikasi dengan *dataset* berupa data ekstraksi berdasar pada nilai VI yang merepresentasikan tingkat kehijauan dari sebuah tanaman teh.

4. Menentukan klasifikasi kelayakan petik tanaman teh dari hasil pengolahan citra area perkebunan teh menggunakan Artificial Neural Network dengan Sequential Model. Penentuan klasifikasi kesiapan petik tanaman teh berdasarkan citra area perkebunan teh dapat diperoleh dengan implementasi metode *Artificial Neural Network, Sequential Model*. Melalui proses pelatihan atau *training* terhadap model yang telah dikonfigurasi berdasarkan *dataset* VI dari objek tanaman teh, model pada akhirnya dapat melakukan klasifikasi dari kesiapan petik tanaman teh dengan nilai 0 yang berarti status tidak siap petik dan 1 untuk status siap petik. Metode kesiapan petik tanaman teh berhasil diimplementasikan dengan menggunakan TensorFlow, Keras sebagai *library* dengan Bahasa pemrograman Python. Hasil dari pelatihan model pemetaan tanaman teh memiliki akurasi sebesar 82%, dan nilai *loss* sebesar 0.4202 yang merupakan hasil pelatihan menggunakan satu *input layer*, satu *hidden layer* dengan 16 neuron, dan satu *output layer* dan jumlah *epoch* sebanyak 100 *epochs* selama pelatihan.

## 5.2. Saran

Adapun saran yang diberikan berdasarkan penelitian ini guna untuk perimbangan pada penelitian berikutnya adalah sebagai berikut.

1. Penambahan kuantitas data untuk dijadikan *dataset* dapat dilakukan untuk menambah akurasi dari model yang dilatih.
2. Diperlukan adanya konfirmasi lapangan terkait dengan hasil identifikasi kesiapan petik, sehingga dapat ditentukan kesalahan yang terjadi apakah karena kesalahan model atau memang keadaan lapangan yang seperti itu adanya.
3. Pembangunan *Grapichal User Interface* (GUI) dapat dilakukan untuk mempermudah penggunaan dari model-model yang telah dibuat.

4. Diperlukan penelitian lebih lanjut terkait dengan konfigurasi serta arsitektur model atau bahkan metode yang lebih baik untuk pemetaan serta identifikasi kesiapan petik teh.

## DAFTAR PUSTAKA

- Alcantarilla, P., Nuevo, J., & Bartoli, A. (2013). Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. *Proceedings of the British Machine Vision Conference 2013*, 13.1-13.11. <https://doi.org/10.5244/C.27.13>
- Barbedo, J. (2019). A Review on the Use of Unmanned Aerial Vehicles and Imaging Sensors for Monitoring and Assessing Plant Stresses. *Drones*, 3(2), 40. <https://doi.org/10.3390/drones3020040>
- Brown, M., & Lowe, D. G. (2007). Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, 74(1), 59–73. <https://doi.org/10.1007/s11263-006-0002-3>
- Cermakova, I., Komarkova, J., & Sedlak, P. (2019). Calculation of Visible Spectral Indices from UAV-Based Data: Small Water Bodies Monitoring. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–5. <https://doi.org/10.23919/CISTI.2019.8760609>
- Chadijah, S., Musdalifah, Qaddafi, M., & Firnanely. (2021). OPTIMALISASI SUHU DAN WAKTU PENYEDUHAN DAUN TEH HIJAU (*Camellia sinensis* L.) P+3 TERHADAP KANDUNGAN ANTIOKSIDAN KAFEIN, KATEKIN DAN TANIN. *Bencoolen Journal of Pharmacy*, 1(1), 59–65.
- Chaprilia, A., & Yuliawati. (2018). FAKTOR-FAKTOR YANG MEMPENGARUHI VOLUME EKSPOR TEH PTPN IX, JAWA TENGAH. *SEPA: Jurnal Sosial Ekonomi Pertanian dan Agribisnis*, 14(2), 167–175.
- Corresp, S. C., Manthachitra, V., Nualchawee, K., & Buranapratheprat, A. (2018). *Application of aerial photography with visible atmospherically resistant index by using unmanned aerial vehicles for seagrass bed classification in Kung Krabaen Bay, Thailand*. <https://doi.org/10.7287/peerj.preprints.27407v1>
- Costa, L., Nunes, L., & Ampatzidis, Y. (2020). A new visible band index (vNDVI) for estimating NDVI values on RGB images utilizing genetic algorithms. *Computers and Electronics in Agriculture*, 172, 105334. <https://doi.org/10.1016/j.compag.2020.105334>
- Cuaran, J., & Leon, J. (2021). Crop Monitoring using Unmanned Aerial Vehicles: A Review. *Agricultural Reviews*, 42(2), 121–132. <https://doi.org/10.18805/ag.r-180>
- Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012). A brief introduction to OpenCV. *2012 Proceedings of the 35th International Convention MIPRO*, 1725–1730.
- Dai, Y., & Wu, J. (2023). An Improved ORB Feature Extraction Algorithm Based on Enhanced Image and Truncated Adaptive Threshold. *IEEE Access*, 11, 32073–32081. <https://doi.org/10.1109/ACCESS.2023.3261665>
- Dhana Lakshmi, M., Mirunalini, P., Priyadharsini, R., & Mirnalinee, T. T. (2019). Review of Feature Extraction and Matching Methods for Drone Image Stitching. Dalam *Lecture Notes in Computational Vision and Biomechanics* (Vol. 30, hlm. 595–602). Springer Netherlands. [https://doi.org/10.1007/978-3-030-00665-5\\_59](https://doi.org/10.1007/978-3-030-00665-5_59)
- DJI. (2023). *Mavic Pro - DJI*. <https://www.dji.com/id/mavic>
- Effendi, D., Syakir, M., Yusron, M., & Wiratno, W. (2010). *Budidaya dan Pasca Panen Teh* (J. Jusniarti, A. Budiharto, H. T. Luntungan, & D. S. Effendi, Ed.). Pusat Penelitian dan Perkembangan Perkebunan.
- Eng, L. S., Ismail, R., Hashim, W., & Baharum, A. (2019). The Use of VARI, GLI, and VIgreen Formulas in Detecting Vegetation In aerial Images. *International Journal of Technology*, 10(7), 1385. <https://doi.org/10.14716/ijtech.v10i7.3275>

- Evangelides, C., & Nobajas, A. (2020). Red-Edge Normalised Difference Vegetation Index (NDVI705) from Sentinel-2 imagery to assess post-fire regeneration. *Remote Sensing Applications: Society and Environment*, 17, 100283.  
<https://doi.org/10.1016/j.rssae.2019.100283>
- Faruqi, U. Al. (2019). Survey Paper : Future Service in Industry 5.0. *Jurnal Sistem Cerdas*, 2(1), 67–79.
- Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). Computer vision algorithms and hardware implementations: A survey. Dalam *Integration* (Vol. 69, hlm. 309–320). Elsevier B.V. <https://doi.org/10.1016/j.vlsi.2019.07.005>
- Girshick, R. (2015). *Fast R-CNN*. <http://arxiv.org/abs/1504.08083>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). *Rich feature hierarchies for accurate object detection and semantic segmentation*. <http://arxiv.org/abs/1311.2524>
- Gitelson, A. A., Kaufman, Y. J., Stark, R., & Rundquist, D. (2002). Novel algorithms for remote estimation of vegetation fraction. *Remote Sensing of Environment*, 80(1), 76–87.  
[https://doi.org/10.1016/S0034-4257\(01\)00289-9](https://doi.org/10.1016/S0034-4257(01)00289-9)
- Gollapudi, S. (2019). Learn Computer Vision Using OpenCV. Dalam *Learn Computer Vision Using OpenCV*. Apress. <https://doi.org/10.1007/978-1-4842-4261-2>
- Gurusamy, V., Kannan, S., & Nalini, G. (2014). *REVIEW ON IMAGE SEGMENTATION TECHNIQUES*. <https://www.researchgate.net/publication/273127438>
- Hardoyo, A. B. (2013). DESAIN VERNAKULER PADA MEDIA KEMASAN TEH SEDUH. *VISUALITA*, 5(1), 2301–5144. <https://doi.org/10.33375/vslt.v5i1.1100>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. <http://arxiv.org/abs/1703.06870>
- Hernina, R., & Putra, T. A. (2021). *FOTO UDARA DIGITAL Teori dan Praktik (Menggunakan Agisoft Metashape)* (Supriatna & Wibowo Adi, Ed.). Departemen Geografi FMIPA Universitas Indonesia. <https://www.researchgate.net/publication/357517871>
- Ihmeida, M., & Wei, H. (2021). Image Registration Techniques and Applications: Comparative Study on Remote Sensing Imagery. *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, 142–148.  
<https://doi.org/10.1109/DeSE54285.2021.9719538>
- Jakhar, D., & Kaur, I. (2020). Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and Experimental Dermatology*, 45(1), 131–132.  
<https://doi.org/10.1111/ced.14029>
- Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A Survey of Deep Learning-based Object Detection. *IEEE Access*, 7, 128837–128868.  
<https://doi.org/10.1109/ACCESS.2019.2939201>
- Kaur, D., & Kaur, Y. (2014). Various Image Segmentation Techniques: A Review. *International Journal of Computer Science and Mobile Computing*, 3(5), 809–814. [www.ijcsmc.com](http://www.ijcsmc.com)
- Kazemi, F., & Ghanbari Parmehr, E. (2023). EVALUATION OF RGB VEGETATION INDICES DERIVED FROM UAV IMAGES FOR RICE CROP GROWTH MONITORING. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W1-2022(4/W1-2022), 385–390. <https://doi.org/10.5194/isprs-annals-X-4-W1-2022-385-2023>
- Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, 2548–2555.  
<https://doi.org/10.1109/ICCV.2011.6126542>

- Li, H., Shi, H., Du, A., Mao, Y., Fan, K., Wang, Y., Shen, Y., Wang, S., Xu, X., Tian, L., Wang, H., & Ding, Z. (2022). Symptom recognition of disease and insect damage based on Mask R-CNN, wavelet transform, and F-RNet. *Frontiers in Plant Science*, 13. <https://doi.org/10.3389/fpls.2022.922797>
- Li, S., Deng, Y.-Q., Zhu, Z.-L., Hua, H.-L., & Tao, Z.-Z. (2021). A Comprehensive Review on Radiomics and Deep Learning for Nasopharyngeal Carcinoma Imaging. *Diagnostics*, 11(9), 1523. <https://doi.org/10.3390/diagnostics11091523>
- Lin, J., Lin, H., & Wang, F. (2022). STPM\_SAHI: A Small-Target Forest Fire Detection Model Based on Swin Transformer and Slicing Aided Hyper Inference. *Forests*, 13(10), 1603. <https://doi.org/10.3390/f13101603>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dolí, P. (2015). *Microsoft COCO: Common Objects in Context*. <https://doi.org/10.48550/arXiv.1405.0312>
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- Luo, D., Gao, Y., Wang, Y., Shi, Y., Chen, S., Ding, Z., & Fan, K. (2022). Using UAV image data to monitor the effects of different nitrogen application rates on tea quality. *Journal of the Science of Food and Agriculture*, 102(4), 1540–1549. <https://doi.org/10.1002/jsfa.11489>
- Manaswi, N. K. (2018). Understanding and Working with Keras. Dalam *Deep Learning with Applications Using Python* (hlm. 31–43). Apress. [https://doi.org/10.1007/978-1-4842-3516-4\\_2](https://doi.org/10.1007/978-1-4842-3516-4_2)
- Manfreda, S., McCabe, M., Miller, P., Lucas, R., Pajuelo Madrigal, V., Mallinis, G., Ben Dor, E., Helman, D., Estes, L., Ciraolo, G., Müllerová, J., Tauro, F., de Lima, M., de Lima, J., Maltese, A., Frances, F., Caylor, K., Kohv, M., Perks, M., ... Toth, B. (2018). On the Use of Unmanned Aerial Systems for Environmental Monitoring. *Remote Sensing*, 10(4), 641. <https://doi.org/10.3390/rs10040641>
- Maulida, H. R., Syifa, F. T., & Afandi, M. A. (2022). Pengujian Sistem RSSI pada Perancangan Prototype Pemantauan Lahan Kebun Teh Berbasis LoRa. *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, 4(2), 117–127. <https://doi.org/10.20895/jtece.v4i2.538>
- Merz, G. M., Liu, Y., Burke, C. J., Aleo, P. D., Liu, X., Kind, M. C., Kindratenko, V., & Liu, Y. (2023). *Detection, Instance Segmentation, and Classification for Astronomical Surveys with Deep Learning (DeepDISC): Detectron2 Implementation and Demonstration with Hyper Suprime-Cam Data*. <http://arxiv.org/abs/2307.05826>
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). *Image Segmentation Using Deep Learning: A Survey*. <http://arxiv.org/abs/2001.05566>
- Mufadhol, M., Siswanto, S., Susatyono, D. D., & Dewi, M. U. (2017). The Phenomenon of Research and Development Method in Research of Software Engineering. *International Journal of Artificial Intelligence Research*, 1(1), 1. <https://doi.org/10.29099/ijair.v1i1.4>
- Nagendran, M., Chen, Y., Lovejoy, C. A., Gordon, A. C., Komorowski, M., Harvey, H., Topol, E. J., Ioannidis, J. P. A., Collins, G. S., & Maruthappu, M. (2020). Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. *BMJ*, 368, m689. <https://doi.org/10.1136/bmj.m689>

- Neupane, K., & Baysal-Gurel, F. (2021). Automatic identification and monitoring of plant diseases using unmanned aerial vehicles: A review. Dalam *Remote Sensing* (Vol. 13, Nomor 19). MDPI. <https://doi.org/10.3390/rs13193841>
- Nur Anisa, M., Rokhmatuloh, & Hernina, R. (2020). UAV application to estimate oil palm trees health using Visible Atmospherically Resistant Index (VARI) (Case study of Cikabayan Research Farm, Bogor City). *E3S Web of Conferences*, 211, 05001. <https://doi.org/10.1051/e3sconf/202021105001>
- O' Mahony, N., Campbell, S., Krpalkova, L., Riordan, D., Walsh, J., Murphy, A., & Ryan, C. (2019). Computer Vision for 3D Perception. Dalam *Advances in Intelligent Systems and Computing* (Vol. 869, hlm. 788–804). Springer Verlag. [https://doi.org/10.1007/978-3-030-01057-7\\_59](https://doi.org/10.1007/978-3-030-01057-7_59)
- Oktaviano, R., Ripanti, E. F., & Pratiwi, H. S. (2021). Implementasi Image Stitching pada Aplikasi Virtual Tour Bandar Udara Internasional Supadio. *Jurnal Sistem dan Teknologi Informasi (Justin)*, 9(3), 381. <https://doi.org/10.26418/justin.v9i3.45056>
- Özgür Çelik, M., Bunyan unel, F., Kanun, E., Alptekin, A., Bünyan Ünel, F., & Kuşak, L. (2020). *The effect of different flight heights on generated digital products: Dsm and Orthophoto*. <https://dergipark.org.tr/tr/pub/mephobj>
- Pham, N. T., Park, S., & Park, C.-S. (2021). Fast and Efficient Method for Large-Scale Aerial Image Stitching. *IEEE Access*, 9, 127852–127865. <https://doi.org/10.1109/ACCESS.2021.3111203>
- Ploeger, S., & Dasovic, L. (2021). *Issues in Object Detection in Videos using Common Single-Image CNNs*. <http://arxiv.org/abs/2105.12822>
- Pratama, R. B. (2018). Penerapan Metode Eigenface pada Sistem Parkir Berbasis Image Processing. *Jurnal Disprotek*, 9(2), 86–96. <https://doi.org/10.34001/jdpt.v9i2.801>
- Puspasari, R. E. (2010). *LAPORAN MAGANG DI PTPN IX (PERSERO) KEBUN JOLOTIGO PEKALONGAN (PROSES PRODUKSI TEH HITAM)*.
- Putri, O. N. (2020). *IMPLEMENTASI METODE CNN DALAM KLASIFIKASI GAMBAR JAMUR PADA ANALISIS IMAGE PROCESSING*. Universitas Islam Indonesia.
- Rahardjo, P., Wulansari, R., & Pranoto, E. (2015). *Aplikasi teknologi portable fertigation system pada bulan kering di perkebunan teh The application of technology portable fertigation system in the dry month in tea plantation*.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. <http://arxiv.org/abs/1506.01497>
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- Saberioon, M. M., Amin, M. S. M., Anuar, A. R., Gholizadeh, A., Wayayok, A., & Khairunniza-Bejo, S. (2014). Assessment of rice leaf chlorophyll content using visible bands at different growth stages at both the leaf and canopy scale. *International Journal of Applied Earth Observation and Geoinformation*, 32(1), 35–45. <https://doi.org/10.1016/j.jag.2014.03.018>
- Sánchez-Sastre, L. F., Alte da Veiga, N. M. S., Ruiz-Potosme, N. M., Carrión-Prieto, P., Marcos-Robles, J. L., Navas-Gracia, L. M., & Martín-Ramos, P. (2020). Assessment of RGB Vegetation Indices to Estimate Chlorophyll Content in Sugar Beet Leaves in the Final Cultivation Stage. *AgriEngineering*, 2(1), 128–149. <https://doi.org/10.3390/agriengineering2010009>

- Satish, S. (S.). (2022, Januari 5). *Satellite VS Drone Imagery: Knowing the difference and effectiveness of SuperVision Earth's complementary synergy*.  
<https://medium.com/supervisionearth/satellite-vs-drone-imagery-knowing-the-difference-and-effectiveness-of-supervision-earths-90e98b78777c>
- Seifert, E., Seifert, S., Vogt, H., Drew, D., van Aardt, J., Kunneke, A., & Seifert, T. (2019). Influence of drone altitude, image overlap, and optical sensor resolution on multi-view reconstruction of forest images. *Remote Sensing*, 11(10).  
<https://doi.org/10.3390/rs11101252>
- Setiawan, A., Yunmar, R. A., & Tanriawan, H. (2020). Comparison of Speeded-Up Robust Feature (SURF) and Oriented FAST and Rotated BRIEF (ORB) Methods in Identifying Museum Objects Using Low Light Intensity Images. *IOP Conference Series: Earth and Environmental Science*, 537(1), 012025. <https://doi.org/10.1088/1755-1315/537/1/012025>
- Shahab, M. T., Al-Safadi, I., & El-Ferik, S. (2007). *Design of a Rotational Speed Measurement System by Computer Vision for Quality Testing*.  
<https://doi.org/10.13140/RG.2.2.36693.27361>
- Sharma, P. P. (2021, Agustus 26). *A Practical Implementation of the Faster R-CNN Algorithm for Object Detection (Part 2 – with Python codes)*.  
<https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/>
- Sharma, S. K., & Jain, K. (2020). Image Stitching using AKAZE Features. *Journal of the Indian Society of Remote Sensing*, 48(10), 1389–1401. <https://doi.org/10.1007/s12524-020-01163-y>
- Sharma, S. K., Jain, K., & Shukla, A. K. (2023). A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching. *Applied Sciences*, 13(10), 6015.  
<https://doi.org/10.3390/app13106015>
- Shi, Y., Gao, Y., Wang, Y., Luo, D., Chen, S., Ding, Z., & Fan, K. (2022). Using Unmanned Aerial Vehicle-Based Multispectral Image Data to Monitor the Growth of Intercropping Crops in Tea Plantation. *Frontiers in Plant Science*, 13.  
<https://doi.org/10.3389/fpls.2022.820585>
- Siregar, A. M. R. (2023). *PENGGUNAAN TEKNOLOGI DRONE DALAM MONITORING LAHAN PERTANIAN*. <https://doi.org/10.31219/osf.io/dmu5g>
- Sishodia, R. P., Ray, R. L., & Singh, S. K. (2020). Applications of Remote Sensing in Precision Agriculture: A Review. *Remote Sensing*, 12(19), 3136. <https://doi.org/10.3390/rs12193136>
- Sita, K., & Rohdiana, D. (2021). Analisis Kinerja dan Prospek Komoditas Teh. Dalam V. Sokoastri (Ed.), *RADAR Analisis dan Opini Perkebunan* (Vol. 2).
- Subash, K. V. V., Srinu, M. V., Siddhartha, M. R. V., Harsha, N. C. S., & Akkala, P. (2020). Object Detection using Ryze Tello Drone with Help of Mask-RCNN. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 484–490. <https://doi.org/10.1109/ICIMIA48430.2020.9074881>
- Subdirektorat Statistik Tanaman Perkebunan. (2021). *STATISTIK TEH INDONESIA 2021*.
- Svyrydov, A., Kuchuk, H., & Tsiapa, O. (2018). Improving efficiency of image recognition process: Approach and case study. *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 593–597.  
<https://doi.org/10.1109/DESSERT.2018.8409201>

- Tang, Z., Li, M., & Wang, X. (2020). Mapping tea plantations from VHR images using OBIA and convolutional neural networks. *Remote Sensing*, 12(18).  
<https://doi.org/10.3390/RS12182935>
- Tareen, S. A. K., & Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1–10.  
<https://doi.org/10.1109/ICOMET.2018.8346440>
- Teixeira, I., Morais, R., Sousa, J. J., & Cunha, A. (2023). Deep Learning Models for the Classification of Crops in Aerial Imagery: A Review. Dalam *Agriculture (Switzerland)* (Vol. 13, Nomor 5). MDPI. <https://doi.org/10.3390/agriculture13050965>
- THEREZA, N., SAPUTRA, I. P. A., & HAMDADI, A. (2020). The Design of Monitoring System of Smart Farming Based on IoT Technology to Support Operational Management of Tea Plantation. *Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*.  
<https://doi.org/10.2991/aisr.k.200424.008>
- Ting, D. S. W., Pasquale, L. R., Peng, L., Campbell, J. P., Lee, A. Y., Raman, R., Tan, G. S. W., Schmetterer, L., Keane, P. A., & Wong, T. Y. (2019). Artificial intelligence and deep learning in ophthalmology. *British Journal of Ophthalmology*, 103(2), 167–175.  
<https://doi.org/10.1136/bjophthalmol-2018-313173>
- Wang, T., Zhang, K., Zhang, W., Wang, R., Wan, S., Rao, Y., Jiang, Z., & Gu, L. (2023). Tea picking point detection and location based on Mask-RCNN. *Information Processing in Agriculture*, 10(2), 267–275. <https://doi.org/10.1016/j.inpa.2021.12.004>
- Wang, Z., & Yang, Z. (2020). Review on image-stitching techniques. *Multimedia Systems*, 26(4), 413–430. <https://doi.org/10.1007/s00530-020-00651-y>
- Wiley, V., & Lucas, T. (2018). Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, 2(1), 22.  
<https://doi.org/10.29099/ijair.v2i1.42>
- Wirabudi, A. A., Munadi, R., Rusdinar, A., Rohdiana, D., & Lee, D. H. (2019). Design Autonomous Drone Control For Monitoring Tea Plantation Using Dynamic Programming and Kruskal Algorithm. *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, 103–109. <https://doi.org/10.1109/ICSIGSYS.2019.8811074>
- Woschank, M., Rauch, E., & Zsifkovits, H. (2020). A Review of Further Directions for Artificial Intelligence, Machine Learning, and Deep Learning in Smart Logistics. *Sustainability*, 12(9), 3760. <https://doi.org/10.3390/su12093760>
- Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2.  
<https://github.com/facebookresearch/detectron2>  
<https://github.com/facebookresearch/detectron2>
- Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A. M., & Wang, X. (2021). Computer Vision Techniques in Construction: A Critical Review. *Archives of Computational Methods in Engineering*, 28(5), 3383–3397. <https://doi.org/10.1007/s11831-020-09504-3>
- Yagüe, F. J., Diez-Pastor, J. F., Latorre-Carmona, P., & Osorio, C. I. G. (2022). Defect detection and segmentation in X-Ray images of magnesium alloy castings using the Detectron2 framework. <http://arxiv.org/abs/2202.13945>
- Yuan, L., Yu, Q., Zhang, Y., Wang, X., Xu, O., & Li, W. (2023). Monitoring Thosea sinensis Walker in Tea Plantations Based on UAV Multi-Spectral Image. *Phyton*, 92(3), 747–761.  
<https://doi.org/10.32604/phyton.2023.025502>

- Zhang, Q., Wan, B., Cao, Z., Zhang, Q., & Wang, D. (2021). Exploring the Potential of Unmanned Aerial Vehicle (UAV) Remote Sensing for Mapping Plucking Area of Tea Plantations. *Forests*, 12(9), 1214. <https://doi.org/10.3390/f12091214>
- Zhao, X., Zhang, J., Tang, A., Yu, Y., Yan, L., Chen, D., & Yuan, L. (2022). The Stress Detection and Segmentation Strategy in Tea Plant at Canopy Level. *Frontiers in Plant Science*, 13. <https://doi.org/10.3389/fpls.2022.949054>
- Zou, J., Han, Y., & So, S.-S. (2008). *Overview of Artificial Neural Networks* (hlm. 14–22). [https://doi.org/10.1007/978-1-60327-101-1\\_2](https://doi.org/10.1007/978-1-60327-101-1_2)