

**SISTEM DETEKSI TUNAS AIR PADA TANAMAN KAKAO**

**WATER SPROUTS DETECTION SYSTEM ON CACAO  
PLANTS**

**NURUL MAULIDYAH**



**PROGRAM PASCASARJANA  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2020**

# **SISTEM DETEKSI TUNAS AIR PADA TANAMAN KAKAO**

Tesis

Sebagai Salah Satu Syarat Untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan diajukan oleh

**NURUL MAULIDYAH**

**D032181013**

**Kepada**

**PROGRAM PASCASARJANA**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2020**

# TESIS

## SISTEM DETEKSI TUNAS AIR PADA TANAMAN KAKAO

Disusun dan diajukan oleh

**NURUL MAULIDYAH**

**Nomor Pokok D032181013**

Telah dipertahankan di depan Panitia Ujian Tesis  
pada tanggal 21 Oktober 2020  
dan dinyatakan telah memenuhi syarat

Menyetujui  
Komisi Penasehat,

Dr. Indrabayu, ST., M.Bus.Sys., MT  
Ketua

Dr. Eng. Intan Sari Areni, ST., MT  
Sekretaris

Ketua Program Studi S2  
Teknik Elektro,



Prof. Dr. Eng. Syafaruddin, ST, M.Eng.  
NIP. 197405301999031003

Dekan Fakultas Teknik  
Universitas Hasanuddin,



Prof. Dr. Ir. Muhammad Arsyad Thaha, MT  
NIP. 196012311986091001

## PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini:

Nama : Nurul Maulidyah  
Nomor Pokok : D032181013  
Program Studi : Teknik Elektro  
Konsentrasi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 18 November 2020

Yang menyatakan,



Nurul Maulidyah

## KATA PENGANTAR

Segala puji selalu dipanjatkan kepada Allah SWT Yang Maha Kuasa yang telah memberikan rahmat, hidayah dan pertolongan-Nya dalam menyelesaikan tesis, yang berjudul “**Deteksi Tunas Air Pada Tanaman Kakao.**” Tak lupa pula shalawat dan salam kepada Nabi Muhammad SAW yang telah menyinari dunia ini dengan keindahan ilmu dan akhlak yang diajarkan kepada seluruh umatnya.

Ucapan terima kasih pun penulis hanturkan kepada kepada dosen pembimbing tesis Dr. Indrabayu, ST., MT., M.Bus.Sys dan Dr. Eng. Intan Sari Areni, ST., MT. yang telah meluangkan waktunya untuk membimbing dan berkonsultasi tentang materi dalam tesis ini dan juga kepada seluruh dosen dan staf Departemen Teknik Elektro, Universitas Hasanuddin yang telah membantu dalam hal keilmuan maupun administrasi pada tahap tesis ini. Penulis menyadari bahwa tesis ini masih belum sempurna. Dengan demikian, penulis tetap mengharapkan kritik dan saran dengan harapan tulisan ini bisa memberikan manfaat kepada seluruh pihak.

Makassar, 18 November 2020

Nurul Maulidyah

## ABSTRAK

**Nurul Maulidyah.** Sistem Deteksi Tunas Air Pada Tanaman Kakao (dibimbing oleh Indrabayu dan Intan Sari Areni).

Penelitian ini bertujuan untuk mendeteksi tunas air tanaman kakao yang seringkali dianggap sebagai hama oleh petani kakao. Hal ini disebabkan karena tunas air dapat menyebabkan *cherelle wilt* dan mengurangi produksi buah kakao. Metode Mask R-CNN (*Region-based Convolutional Neural Network*) dan FCN (*Fully Convolutional Network*) digunakan untuk melakukan *instance segmentation* pada masing-masing tunas air yang dideteksi. Pada penelitian ini digunakan data latih dan data uji masing-masing sebanyak 120 dan 30 data gambar. Pengambilan gambar dilakukan di kebun kakao di Desa Awo, Kabupaten Wajo menggunakan kamera. Gambar diambil dari jarak 100 cm, 150 cm dan 200 cm dari tanaman kakao dan dari ketinggian 100 cm. Proses pengujian dilakukan pada *minimum confidence threshold* 0.1 – 0.9 untuk melihat jumlah objek yang dideteksi pada tiap konfigurasi. Hasil deteksi dievaluasi menggunakan *F1-score* di mana *F1-score* tertinggi diperoleh pada *minimum confidence threshold* 0.9 yaitu sebesar 96.6%.

Kata kunci: tunas air, kakao, Mask R-CNN, FCN, *instance segmentation*

## ABSTRACT

**Nurul Maulidyah.** Water Sprouts Detection System on Cacao Plants (supervised by Indrabayu and Intan Sari Areni)

This study aims to detect water sprouts of cocoa plants that are considered a nuisance by cocoa farmers. This is because water sprouts can cause cherelle wilt which then causes the production of cocoa pods to decrease. Mask R-CNN (Region-based Convolutional Neural Network) and FCN (Fully Convolutional Network) methods were used to perform instance segmentation instances on each detected water sprouts. This study uses training data and test data of 120 and 30 image data, respectively. Data were collected at a cocoa farm at Awo Village, Wajo Regency using camera. Images were taken from the distance of 100 cm, 150 cm and 200 cm from the cocoa plant with a height of 100 cm. The testing process is carried out at a minimum confidence threshold of 0.1 - 0.9 to see the number of objects detected in each configuration. The detection results were evaluated using the F1-score where the highest F1-score was 96.6% which was obtained at a minimum confidence threshold of 0.9.

Keywords: water sprouts, cacao, Mask R-CNN, FCN, instance segmentation

## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
PERNYATAAN KEASLIAN TESIS.....	ii
KATA PENGANTAR.....	iii
ABSTRAK.....	iv
ABSTRACT.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	ix
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	5
C. Tujuan Penelitian.....	5
D. Manfaat Penelitian.....	5
E. Batasan Masalah.....	6
F. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	9
A. Landasan Teori.....	9
B. Penelitian Terkait.....	44
C. State of The Art.....	47
D. Kerangka Pikir.....	50
BAB III METODE PENELITIAN.....	52
A. Tahapan Penelitian.....	52
B. Waktu dan Lokasi Penelitian.....	54
C. Instrumentasi Penelitian.....	54
D. Pengambilan Data.....	55
E. Perancangan Sistem.....	56
F. Analisis Kinerja Sistem.....	78
BAB IV HASIL DAN PEMBAHASAN.....	81
A. Hasil Penelitian.....	81

B. Pembahasan.....	83
BAB V KESIMPULAN DAN SARAN .....	88
A. Kesimpulan .....	88
B. Saran .....	88
DAFTAR PUSTAKA.....	90

## DAFTAR TABEL

Tabel 1. <i>State of the art</i> .....	48
Tabel 2. Performa sistem deteksi tunas air .....	81
Tabel 3. Hasil deteksi tunas air .....	82
Tabel 4. Hasil deteksi pada <i>minimum confidence threshold</i> 0.9 .....	82
Tabel 5. Performa sistem pada <i>minimum confidence threshold</i> >0.9 .....	84
Tabel 6. Objek tunas air yang tidak terdeteksi pada jarak 200 cm .....	85

## DAFTAR GAMBAR

Gambar 1. Tunas air pada tanaman kakao .....	13
Gambar 2. Ilustrasi pohon kakao .....	15
Gambar 3. Aturan koordinat untuk representasi citra digital .....	19
Gambar 4. Aplikasi <i>smoothing</i> pada citra Lena .....	20
Gambar 5. Matriks ketetangaan (a) horizontal dan vertikal, (b) diagonal	21
Gambar 6. Contoh kernel untuk <i>blurring</i> dan <i>sharpening</i> .....	22
Gambar 7. Tampilan Google Colab .....	25
Gambar 8. Tampilan halaman utama Jupyter Notebook .....	26
Gambar 9. Tampilan antar muka Jupyter Notebook .....	27
Gambar 10. Ilustrasi operasi konvolusi (Pokharna, 2016) .....	29
Gambar 11. Contoh Hasil konvolusi menggunakan sebuah kernel konvolusi (Dettmers, 2015) .....	29
Gambar 12. <i>Convolutional layer</i> dengan 1 filter .....	30
Gambar 13. <i>Convolutional layer</i> dengan 6 filter .....	31
Gambar 14. <i>Max pooling</i> (Pokharna, 2016) .....	31
Gambar 15. <i>2D bounding boxes</i> (Ambalina, 2019) .....	33
Gambar 16. <i>3D bounding box</i> (Ambalina, 2019) .....	34
Gambar 17. <i>Image classification</i> (Ambalina, 2019) .....	34
Gambar 18. <i>Lines and splines</i> (Ambalina, 2019) .....	34
Gambar 19. <i>Polygon</i> (Ambalina, 2019) .....	35
Gambar 20. <i>Semantic segmentation</i> (Ambalina, 2019) .....	35
Gambar 21. Perbandingan <i>plain network</i> dan <i>residual network</i> .....	37
Gambar 22. <i>Instance segmentation</i> (Abdulla, 2018) .....	39
Gambar 23. Arsitektur Mask R-CNN .....	40
Gambar 24. Ekstraksi fitur menggunakan ResNet101-FPN .....	42
Gambar 25. Proses RPN .....	43
Gambar 26. Menghitung IoU .....	44
Gambar 27. Kerangka pikir .....	50
Gambar 28. Diagram tahapan penelitian .....	52

Gambar 29. Skenario pengambilan gambar .....	56
Gambar 30. Contoh data dengan tiga jarak pengambilan .....	56
Gambar 31. Desain alur sistem deteksi tunas air .....	57
Gambar 32. Contoh data tunas air .....	58
Gambar 33. Flowchart proses <i>labelling</i> menggunakan VIA.....	59
Gambar 34. Pelabelan menggunakan VIA.....	60
Gambar 35. Flowchart tahap <i>training</i> di Google Colab.....	61
Gambar 36. Perintah download modul <i>Mask R-CNN</i> .....	62
Gambar 37. Perintah <i>install</i> modul <i>Mask R-CNN</i> .....	62
Gambar 38. Perintah download data .....	62
Gambar 39. Perintah <i>training</i> .....	63
Gambar 40. Tahapan <i>pre-processing</i> pada <i>Mask R-CNN</i> .....	63
Gambar 41. Contoh hasil aplikasi <i>mean subtraction</i> .....	64
Gambar 42. Perintah untuk <i>re-scale</i> citra.....	65
Gambar 43. Perintah untuk <i>padding</i> .....	66
Gambar 44. Parameter augmentasi data citra .....	67
Gambar 45. Contoh hasil augmentasi.....	68
Gambar 46. Contoh <i>anchor boxes</i> pada piksel (512, 512) .....	70
Gambar 47. Perintah untuk menghitung IoU .....	71
Gambar 48. Contoh <i>anchor</i> positif pada citra .....	72
Gambar 49. Contoh <i>anchor</i> yang telah dilakukan <i>refinement</i> .....	73
Gambar 50. <i>Bounding box</i> dengan <i>class</i> dan <i>confidence score</i> .....	74
Gambar 51. Proses testing menggunakan Jupyter .....	77
Gambar 52. Contoh hasil deteksi.....	78
Gambar 53. Bagan <i>Confusion Matrix</i> .....	79
Gambar 54. Gambar daun kering yang terdeteksi sebagai tunas air.....	86

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Otomasi di bidang pertanian sudah menjadi perhatian di berbagai negara. Meningkatnya pertumbuhan penduduk menuntut manusia untuk menghasilkan lebih banyak makanan (Syarifudin, 2016). Untuk mencukupi kebutuhan penduduk yang terus bertambah, dunia pertanian mengadopsi istilah Revolusi Pertanian 4.0, di mana pertanian diharapkan melibatkan teknologi digital dalam proses pengembangannya.

Tanaman kakao merupakan salah satu komoditas pertanian terpenting di dunia selain teh dan kopi (Diby *et al.*, 2017). Selain digunakan sebagai minuman, kakao pada dasarnya merupakan bahan utama dalam pembuatan produk olahan coklat. Karena rasanya yang enak, produk olahan coklat banyak digemari orang di seluruh dunia. Banyaknya penggemar coklat berdampak pada bertambahnya permintaan terhadap biji kakao setiap tahun. Hal ini terlihat dari meningkatnya produksi kakao dunia dari 1.5 juta ton pada tahun 1964 menjadi 4.3 juta ton pada tahun 2015 (Jagoret *et al.*, 2018). Namun peningkatan ini tidak selalu diiringi dengan peningkatan produksi biji itu sendiri. Di beberapa negara produksi biji kakao justru mengalami penurunan. Salah satu negara yang mengalami penurunan produksi ialah Indonesia. Menurut data International Cocoa Organization (ICCO), jumlah produksi biji kakao Indonesia pada tahun 2016

sebanyak 320 ribu ton (“QBCS,” 2016) dan menyusut menjadi 290 ribu ton pada tahun 2017 (Gumilar, 2018).

Tanaman kakao (*Theobroma Cacao*) merupakan tanaman yang banyak dibudidayakan di perkebunan. Di habitat alaminya, kakao akan tumbuh tinggi tetapi bunga dan buah yang dihasilkan sedikit. Berbeda dengan tanaman kakao yang dibudidayakan di kebun, di mana produksi buah dapat dimaksimalkan salah satunya dengan melakukan pemangkasan.

Pemangkasan tanaman memerlukan pengetahuan tentang apa, kapan, dan bagaimana pemangkasan itu dilakukan. Pada tanaman kakao terdapat bagian tanaman yang disebut tunas air. Tunas air tumbuh dengan cepat dan banyak muncul di batang utama. Selain itu tunas air juga menyerap banyak nutrisi tanaman yang dibutuhkan dalam pembentukan buah sehingga buah kakao beresiko kekurangan nutrisi dan menyebabkan layu pentil (buah yang masih muda) buah kakao. Hal ini menyebabkan tunas air harus segera dipangkas untuk bisa memperoleh produksi kakao yang maksimal.

Pemangkasan tunas air termasuk dalam pemangkasan pemeliharaan yang bertujuan untuk membuang cabang-cabang sekunder dan mempertahankan kerangka tanaman. Pengetahuan tentang pemangkasan dapat diperoleh melalui pelatihan yang seringkali diadakan oleh pemerintah dan pihak swasta. Namun, meski sudah dibekali dengan cukup ilmu, petani belum tentu menerapkannya di kebun. Menurut penelitian yang dilakukan oleh Baihaqi di Kabupaten Pidie, diketahui bahwa dari 86.65% petani yang

mengetahui teknik pemangkasan, hanya 6.66% yang mampu menerapkan teknik tersebut di kebunnya secara menyeluruh, sisanya hanya diterapkan sebagian karena keterbatasan biaya (Baihaqi, 2015).

Kegiatan pemangkasan bersifat padat karya. Diperlukan tenaga yang dapat secara rutin melakukan pemangkasan di kebun, terlebih untuk tunas air tanaman kakao yang sering muncul pada cabang primer dan cabang sekunder (Prawoto dkk., 2014). Biaya untuk pemangkasan dapat menghabiskan 20% atau lebih biaya produksi pra-panen (Karkee *et al.*, 2014). Di sisi lain terjadi krisis petani di mana petani kini di dominasi oleh petani usia tua, sementara generasi yang lebih muda memilih untuk bekerja di perkotaan.

Data BPS menunjukkan terjadi penurunan jumlah Rumah Tangga Petani (RTP) yang semula berjumlah 31.23 juta pada tahun 2003 dan turun sebanyak 16.3% menjadi 26.13 juta RTP pada tahun 2013. Di bidang kakao, hanya 16% petani berusia di bawah 35 tahun yang aktif mengelola kebunnya. Angka ini akan semakin mengkhawatirkan jika ditambahkan dengan jumlah petani berusia di atas 55 tahun yang mencapai 23%.

Penelitian mengenai otomasi di bidang pertanian sudah dilakukan sejak beberapa dekade terakhir. Namun untuk tanaman tertentu termasuk tanaman buah-buahan, penerapan otomatisasi telah tertinggal karena kompleksitas operasi lapangan dan sistem tanam yang tidak konsisten (Baugher and He, 2018). Saat ini sebagian besar penelitian tentang solusi otomasi untuk produksi tanaman buah difokuskan pada proses panen dan pemangkasan di mana visi mesin adalah salah satu teknologi penting yang

digunakan untuk mendeteksi dan melokalisasi dan deteksi serta rekonstruksi cabang untuk pemangkasan.

Pada penelitian sebelumnya deteksi cabang tanaman dilakukan menggunakan sensor mulai dari *laser scanner*, *3D camera*, Kinect v2 dan *RGB-D camera* dan dilakukan di perkebunan dengan arsitektur pohon yang lebih ramah untuk pengoperasian mesin (He and Schupp, 2018). Maka dari itu sebagai penelitian awal, penggunaan perangkat yang lebih sederhana seperti kamera dan data berupa gambar dapat digunakan untuk menentukan metode deteksi tunas air pada tanaman kakao.

Tunas air dapat dideteksi dari daun dan batangnya yang berwarna hijau muda hingga merah kecokelatan. Beberapa penelitian sebelumnya yang terkait penggunaan visi komputer untuk deteksi dan segmentasi pada daun antara lain Wang *et al.* (2018), (Zhang *et al.*, 2018), (Singh, 2019), (Tian *et al.*, 2019), (Xu *et al.*, 2018) dan Shi *et al.*, (2019). Masing masing penelitian tersebut menggunakan metode berbeda yang disesuaikan dengan data dan masalah yang ingin diselesaikan. Dengan mempertimbangkan warna tunas air yang menyerupai daun kering dan bentuknya yang beragam maka tantangannya ialah bagaimana membedakan tunas air dengan objek lainnya yang memiliki karakter yang mirip dengannya. Oleh karena itu dibutuhkan metode yang dapat mendeteksi objek meskipun objek targetnya memiliki kemiripan dengan objek lain disekitarnya.

Penelitian ini mengadopsi metode dari penelitian terkait yaitu metode deep learning *Mask R-CNN* yang mampu melakukan deteksi dan

segmentasi objek dengan baik meskipun objek berada pada lingkungan alaminya. Metode tersebut kemudian dikombinasikan dengan teknik *image processing* untuk mengoptimalkan hasil deteksi objek tunas air. Karena penelitian tentang deteksi tunas air pada tanaman kakao merupakan penelitian awal maka yang menjadi fokus penelitian hanya pada metode deteksi objek tunas air saja.

## **B. Rumusan Masalah**

Rumusan masalah pada penelitian ini adalah:

1. Bagaimana pengaruh *minimum confidence threshold* terhadap unjuk kerja sistem deteksi tunas air pada tanaman kakao menggunakan metode Mask R-CNN dan FCN?
2. Bagaimana pengaruh jarak pengambilan gambar terhadap hasil deteksi tunas air tanaman kakao?

## **C. Tujuan Penelitian**

Tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui pengaruh *minimum confidence threshold* terhadap unjuk kerja sistem deteksi tunas air tanaman kakao menggunakan metode *Mask R-CNN* dan FCN.
2. Mengetahui pengaruh jarak pengambilan gambar terhadap hasil deteksi tunas air tanaman kakao.

## **D. Manfaat Penelitian**

Manfaat penelitian ini adalah sebagai berikut:

1. Manfaat Teoretis
  - a. Untuk pengembangan keilmuan di bidang pengolahan citra dan pengenalan pola dan aplikasinya pada sistem deteksi tunas air tanaman kakao.
  - b. Untuk menambah kajian ilmiah dalam pengembangan pengolahan citra dan pengenalan pola.
2. Manfaat Praktis
  - a. Bagi Industri, penelitian yang diusulkan memberikan kontribusi teknologi menuju revolusi Industri 4.0
  - b. Membantu pengembangan teknologi otomasi di bidang perkebunan khususnya dalam melakukan pemangkasan otomatis pada tanaman kakao.

#### **E. Batasan Masalah**

Batasan masalah pada penelitian adalah:

1. Input berupa gambar tanaman kakao. Tanaman kakao pada penelitian ini ialah tanaman kakao yang ditanam di kebun.
2. Objek penelitian yaitu tunas air tanaman kakao.
3. Proses pengambilan gambar dilakukan saat cuaca cerah.
4. Pengambilan gambar menggunakan kamera Nikon Coolpix P610.

#### **F. Sistematika Penulisan**

Adapun sistematika penulisan pada penelitian deteksi tunas air tanaman kakao adalah:

## **BAB I PENDAHULUAN**

Bab I berisi penjelasan tentang latar belakang penelitian deteksi tunas air tanaman kakao beserta alasan pemilihan metode MRCNN. Selain itu, pada bab ini juga diuraikan rumusan masalah, tujuan, manfaat dan batasan masalah dalam penelitian ini.

## **BAB II TINJAUAN PUSTAKA**

Bab II berisi tentang landasan teori dengan sumber acuan terbaru yang mendukung perancangan sistem, beberapa penelitian yang terkait dengan penelitian yang dilakukan. Selain itu, diuraikan pula *State of The Art* yang merupakan penjelasan tentang hasil-hasil penelitian sebelumnya yang berkaitan dengan penelitian yang diusulkan berkenaan dengan deteksi tunas air pada tanaman kakao.

## **BAB III METODE PENELITIAN**

Bab Metode Penelitian berisi uraian tentang tahapan penelitian, bagaimana pengembangan dan penerapan sistem dengan *computing approach*, rancangan sistem dan proses validasi hasil dari metode yang digunakan.

## **BAB IV HASIL DAN PEMBAHASAN**

Pada Bab IV menyajikan hasil dan pembahasan penelitian serta implikasi dari penelitian yang dilakukan, di mana pada Bab IV, dipaparkan hasil pengujian sistem deteksi tunas air tanaman kakao menggunakan metode *Mask R-CNN*.

## **BAB V KESIMPULAN DAN SARAN**

Pada Bab V berisi kesimpulan mengenai sistem deteksi tunas air tanaman kakao menggunakan metode *Mask R-CNN* beserta akurasi yang dihasilkan. Pada Bab V juga di tuliskan saran perbaikan sistem untuk digunakan pada penelitian selanjutnya.

## BAB II

### TINJAUAN PUSTAKA

#### A. Landasan Teori

##### 1. Kakao

Sistematika tanaman kakao menurut ITIS (*Integrated Taxonomic Information System*) adalah sebagai berikut:

- Kingdom* : *Plantae* (tumbuhan)
- Subkingdom* : *Viridiplantae* (tumbuhan hijau)
- Division* : *Tracheophyta* (tumbuhan berpembuluh)
- Subdivision* : *Spermatophytina* (tumbuhan berbiji)
- Class* : *Magnoliopsida*
- Order* : *Malvales*
- Family* : *Malvaceae*
- Genus* : *Theobroma L.*
- Species* : *Theobroma cacao L.*

Menurut Cuatrecasas (1964) dikenal dua subjenis kakao, yaitu cacao dan *Sphaerocarpum* (chev.) Cuatr. Subjenis cacao mempunyai empat forma (taksonomi di bawah subjenis) seperti berikut.

- 1) Forma cacao. Anggotanya tipe criollo dari Amerika tengah. Bentuk biji bulat, keeping biji (kotiledon) putih, dan mutunya tinggi.
- 2) Forma pentagonum. Hanya dikenal di Meksiko dan Amerika tengah. Biji bulat besar, kotiledon putih, dan mutunya tinggi.

- 3) Forma leiocarpum. Biji bulat atau montok (plum), kotiledon putih atau ungu pucat, dan mutunya tinggi. Klon-klon Djati Runggo (DR) termasuk forma ini.
- 4) Forma lacandonense. Dikenal di dekat Chiapas, Meksiko. Forma ini termasuk kakao liar.

Berdasarkan varietasnya, terdapat tiga macam kakao yang dibudidayakan di dunia antara lain:

- 1) Criollo

Criollo merupakan tipe tanaman kakao yang menghasilkan biji kakao kering premium yang dikenal sebagai *fine flavour cocoa*, *choiced cocoa*, serta kakao mulia. Kakao Criollo dibedakan atas 2 jenis, yaitu *Central America Criollos* dan *South America Criollos*. Kakao jenis Criollo memasok kurang dari 7% produksi kakao dunia yang dihasilkan oleh negara Ecuador, Venezuela, Trinidad, Grenada, Jamaika, Srilanka serta Indonesia.

Kakao jenis Criollo memiliki ciri pertumbuhan tanaman kurang kuat dengan produksi yang rendah serta lamban berbuah. Tunas mudanya berbulu, memiliki tongkol buah berwarna hijau atau merah bila masih muda dan kuning orange ketika sudah matang, agak peka terhadap hama dan penyakit.

Buahnya tumpul, sedikit bengkok dan tidak memiliki *bottleneck*. Tekstur kulit buah kasar berbintil, tipis dan lunak. Terdapat 30 - 40 biji di setiap buah kakao. Biji berbentuk bulat telur dengan kotiledon berwarna putih pada saat kering. Dinding buah

tipis, agak lembut kadang-kadang berkerut. Daun lebih kecil dari jenis-jenis yang lainnya, mempunyai 2-3 cabang dari *jourquette*.

## 2) Forastero

Forastero merupakan tipe tanaman yang menghasilkan biji kakao kering bermutu sedang yang dikenal sebagai *ordinary cacao* atau kakao lindak serta *bulk cacao*. 93% kakao yang beredar di pasaran merupakan jenis kakao lindak yang dihasilkan Negara-negara di Afrika Barat, Brazil dan Dominica. Kakao lindak (*bulk*) adalah anggota subjenis *Sphaerocarpum* yang telah tersebar luas di daerah tropika.

Kakao Forastero memiliki ciri pertumbuhan tanaman kuat dan produktivitas tinggi, menghasilkan buah lebih cepat, relatif lebih tahan terhadap hama dan penyakit. Kulit buah agak keras namun permukaannya halus. Buah memiliki *bottleneck* ada juga yang tidak. Endosperm berwarna ungu tua dan berbentuk gepeng pada waktu basah. Kulit buah berwarna hijau dengan alur kulit buah dalam.

## 3) Trinitario

Trinitario merupakan tipe hibrida yang berasal dari persilangan alami Criollo dan Forastero. Biji kering yang dihasilkan bisa kakao mulia maupun kakao lindak. Kakao Trinitario memiliki ciri yaitu masa pertumbuhan yang cepat dan produktivitas tinggi. Bentuk buah bermacam-macam dengan warna kulit buah

berwarna merah dan hijau. Warna kotiledon berwarna ungu dan ungu tua ketika masih basah.

Kakao merupakan satu-satunya diantara 22 jenis marga *Theobroma*, suku Malvaceae yang diusahakan secara masal karena nilai ekonominya yang tinggi.

Tanaman Kakao (*Theobroma cacao* L.) merupakan salah satu komoditas perkebunan yang memiliki peranan penting dalam meningkatkan pendapatan dan kesejahteraan petani khususnya di Indonesia (Rubiyo and Siswanto, 2012). Indonesia merupakan penghasil kakao ketiga terbesar di dunia. Luas lahan tanaman kakao Indonesia lebih kurang 992.448 ha dengan produksi biji kakao sekitar 456.000 ton per tahun, dan produktivitas rata-rata 900 kg/ha (*Gambaran Sekilas Industri Kakao*, 2007). Sedangkan menurut FAO, pada tahun 2017 total produksi biji kakao Indonesia sebanyak 659.776 ton (FAO, 2019).

Kakao merupakan sumber pendapatan dan mata pencaharian utama bagi sekitar satu juta petani di Indonesia. Sekitar 60 sampai 70% produksi kakao nasional dihasilkan oleh petani kecil di Sulawesi (Tothmihaly *et al.*, 2019).

Di Sulawesi sendiri, kakao yang banyak dibudidayakan adalah jenis Sulawesi 1 (S1) dan Sulawesi 2 (S2). Kedua varietas ini merupakan klon unggulan dari varietas kakao lindak yang sudah dikenal petani lebih dari 30 tahun ("Sulawesi 1 dan 2, Klon Unggul Asal Celebes," n.d.).

**a. Batang dan cabang**

Tanaman kakao memiliki habitat asli di hutan tropis dengan naungan pohon-pohon yang tinggi, curah hujan tinggi, suhu sepanjang tahun relatif sama serta kelembaban tinggi yang relatif tetap. Tanaman kakao bersifat dimorfisme, artinya mempunyai dua bentuk tunas vegetatif. Tunas yang arah pertumbuhannya ke atas seperti yang ditampilkan pada Gambar 1 disebut dengan tunas ortotrop atau tunas air (*wiwilan* atau *chupon*), sedangkan tunas yang arah pertumbuhannya ke samping disebut dengan plagiotrop (*cabang kipas* atau *fan*).



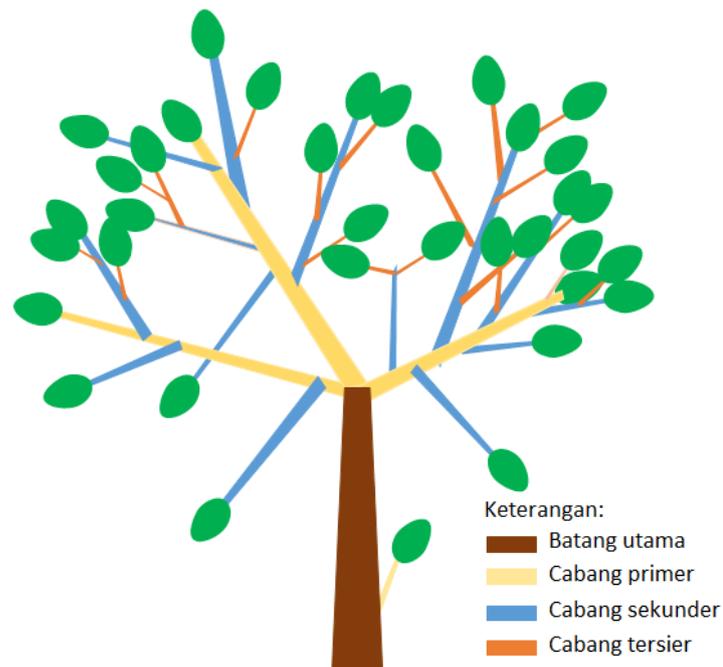
Gambar 1. Tunas air pada tanaman kakao

Tanaman kakao asal biji, setelah mencapai tinggi 0.9 – 1.5 meter akan berhenti tumbuh dan membentuk jorket (*joquette*). Jorket adalah tempat percabangan dari pola percabangan ortotrop ke plagiotrop dan khas hanya pada tanaman kakao. Pembentukan jorket didahului dengan berhentinya pertumbuhan tunas ortotrop karena ruas-ruasnya tidak memanjang.

Pada ujung tunas tersebut, stipula (semacam sisik pada kuncup bunga) dan kuncup ketiak daun serta tunas daun tidak berkembang. Dari ujung perhentian tersebut selanjutnya tumbuh 3 - 6 cabang yang arah pertumbuhannya condong ke samping membentuk sudut 0 - 60° dengan arah horizontal. Cabang-cabang itu disebut dengan cabang primer (cabang plagiotrop). Pada cabang primer (lihat Gambar 2) tersebut kemudian tumbuh cabang-cabang lateral (*fan*) sehingga tanaman membentuk tajuk yang rimbun.

Pada tanaman kakao dewasa sepanjang batang pokok tumbuh wiwilan atau tunas air (*chupon*). Dalam teknik budidaya yang benar, tunas air ini selalu dibuang, tetapi pada tanaman kakao liar, tunas air tersebut akan membentuk batang dan jorket yang baru sehingga tanaman mempunyai jorket yang bersusun.

Dari tunas plagiotrop biasanya tumbuh tunas-tunas plagiotrop, tetapi kadang-kadang juga tumbuh tunas ortotrop. Pangkasan berat pada cabang plagiotrop yang besar ukurannya merangsang tumbuhnya tunas ortotrop itu. Tunas ortotrop hanya membentuk tunas plagiotrop setelah membentuk jorket. Tunas ortotrop membentuk tunas ortotrop baru dengan menumbuhkan tunas air.



Gambar 2. Ilustrasi pohon kakao

#### b. Daun

Sama dengan sifat percabangannya, daun kakao juga bersifat dimorfisme. Pada tunas ortotrop, tangkai daunnya panjang, yaitu 7.5-10 cm sedangkan pada tunas plagiotrop panjang tangkai daunnya hanya sekitar 2,5 cm. Tangkai daun bentuknya silinder dan bersisik halus, bergantung pada tipenya. Salah satu sifat khusus daun kakao yaitu adanya dua persendian (*articulation*) yang terletak di pangkal dan ujung tangkai daun. Dengan persendian ini dilaporkan daun mampu membuat gerakan untuk menyesuaikan dengan arah datangnya sinar matahari.

Bentuk helai daun bulat memanjang (*oblongus*) ujung daun meruncing (*acuminatus*) dan pangkal daun runcing (*acutus*). Susunan daun tulang menyirip dan tulang daun menonjol ke permukaan bawah helai daun. Tepi daun rata, daging daun tipis tetapi kuat seperti perkamen.

## 2. Pemangkasan Tanaman Kakao

Pemangkasan adalah langkah pembuangan beberapa bagian pada tanaman seperti cabang dan ranting untuk mendapatkan bentuk tertentu, sehingga tanaman dapat mencapai tingkat efisiensi yang tinggi agar cahaya matahari mampu menyinari, mempermudah mendeteksi hama penyakit, serta mempermudah proses panen tanaman tersebut.

Selain itu pemangkasan juga berguna untuk mengurangi beban tanaman, sehingga keberadaan daun, ranting, dan buah yang terlampau lebat dapat dikurangi. Dengan begitu, tanaman dapat menghasilkan buah dengan kualitas dan kuantitas yang lebih baik (Agroteknologi, 2016).

### a. Macam-macam pemangkasan dan pelaksanaannya

#### 1) Pemangkasan bentuk

Pemangkasan bentuk terdiri dari dua fase, yaitu fase muda dan fase remaja. Pemangkasan pada fase muda dilakukan pada saat tanaman berumur 8 - 12 bulan dengan memotong cabang yang lemah dan mempertahankan 3 - 4 cabang yang letaknya merata ke segala arah untuk membentuk *jourquette* (percabangan). Sedangkan pemangkasan pada fase remaja, dilakukan pada saat tanaman berumur 18 - 24 bulan dengan memotong ujung cabang primer pada jarak 75 - 100 cm dari *jourquette* (percabangan) untuk memacu tumbuhnya cabang sekunder.

#### 2) Pemangkasan pemeliharaan

Pemangkasan pemeliharaan dilakukan dengan frekuensi 2 - 3 bulan sekali per pohon agar kerangka tanaman kakao yang sudah baik dapat

dipertahankan, mengatur agar penyebaran daun produktif merata, membuang bagian tanaman yang tidak dikehendaki, mengatur agar tinggi tanaman tetap pada kisaran 3 – 4 meter (Angela and Efendi, 2015).

Pemangkasan diprioritaskan pada cabang yang terserang penyakit, cabang yang menggantung, cabang *overlapping*, cabang sekunder dan tersier yang tumbuhnya kurang dari 40 cm dari pangkal cabang primer atau sekunder, cabang-cabang sekunder berikutnya diatur agar jaraknya tidak terlalu rapat satu sama lain, dan tunas air.

### 3) Pemangkasan produksi

Pemangkasan produksi dilakukan berkesinambungan dengan pemangkasan pemeliharaan. Tujuannya adalah untuk memaksimalkan produktivitas tanaman. Pemangkasan produksi dilakukan dua kali setahun pada akhir musim kemarau – awal musim penghujan serta pada pertengahan musim penghujan dengan memangkas daun-daun agar tidak terlalu rimbun sehingga sinar matahari bisa tersebar merata ke seluruh organ daun. Dengan demikian, proses fisiologis terpenting dari tanaman, yaitu fotosintesis bisa berjalan lancar sehingga sirkulasi makanan dari daun keseluruhan organ tanaman juga lancar dan dapat berproduksi secara optimal.

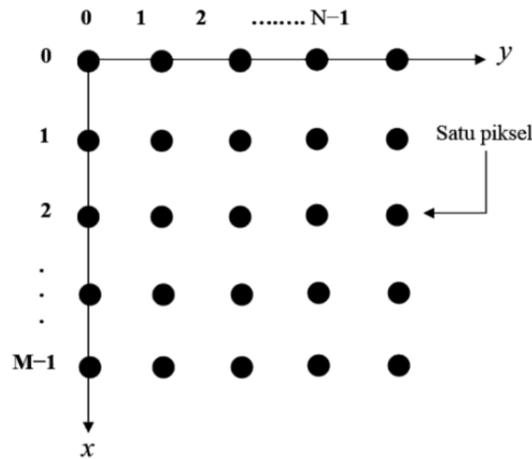
Sasaran pemangkasan produksi adalah ranting-ranting atau cabang tersier yang mendukung daun-daun tidak produktif, ranting-ranting yang sakit atau rusak dan cabang cacing. Tunas-tunas air yang tumbuh dari pangkal cabang tersier dan cabang sekunder pada jarak 15 - 25 cm dari pangkal cabang sekunder dipotong. Ranting-ranting dengan daun yang terlindung atau kurang mendapat sinar matahari juga harus dipotong.

Cabang-cabang tersier yang yang terlalu subur juga dibuang karena sering mengganggu keseimbangan pertumbuhan, demikian pula cabang-cabang kecil yang akan masuk ke dalam tajuk tanaman tetangga atau di dekatnya. Cabang yang menggantung ke bawah dikurangi daunnya agar tidak menghambat sirkulasi udara dalam kebun (Admin, 2014).

### 3. **Image Processing**

*Image processing* merupakan metode untuk mengolah citra digital menggunakan berbagai algoritma *image processing* seperti *image filtering*, *morphological operation*, *geometric transformation*, *color conversion*, *feature detection*, dan lain-lain yang bertujuan untuk memperbaiki kualitas citra ataupun untuk mengekstrak informasi dari citra tersebut (Hermawati, 2013).

Citra merupakan visualisasi objek nyata atau objek lainnya dalam format dua dimensi  $f(x,y)$ , di mana  $x$  dan  $y$  adalah koordinat bidang datar dan harga fungsi  $f$  adalah intensitas atau level keabuan (*grey level*) dari titik di gambar itu. Jika  $x$ ,  $y$  dan  $f$  semuanya berhingga (*finite*) dan nilainya diskrit, maka gambar tersebut merupakan citra digital. Citra digital merupakan format citra yang dapat diolah oleh computer. Sebuah citra digital terdiri dari elemen-elemen yang disebut sebagai *picture element*, *image element*, *pels* atau *pixels* yang merupakan elemen berhingga yang masing-masing mempunyai lokasi (lihat Gambar 3) dan nilai intensitas tertentu (Hermawati, 2013).



Gambar 3. Aturan koordinat untuk representasi citra digital

Sebuah citra digital lengkap berukuran  $M$  baris x  $N$  kolom dapat ditulis dalam bentuk matriks seperti pada Persamaan (1).

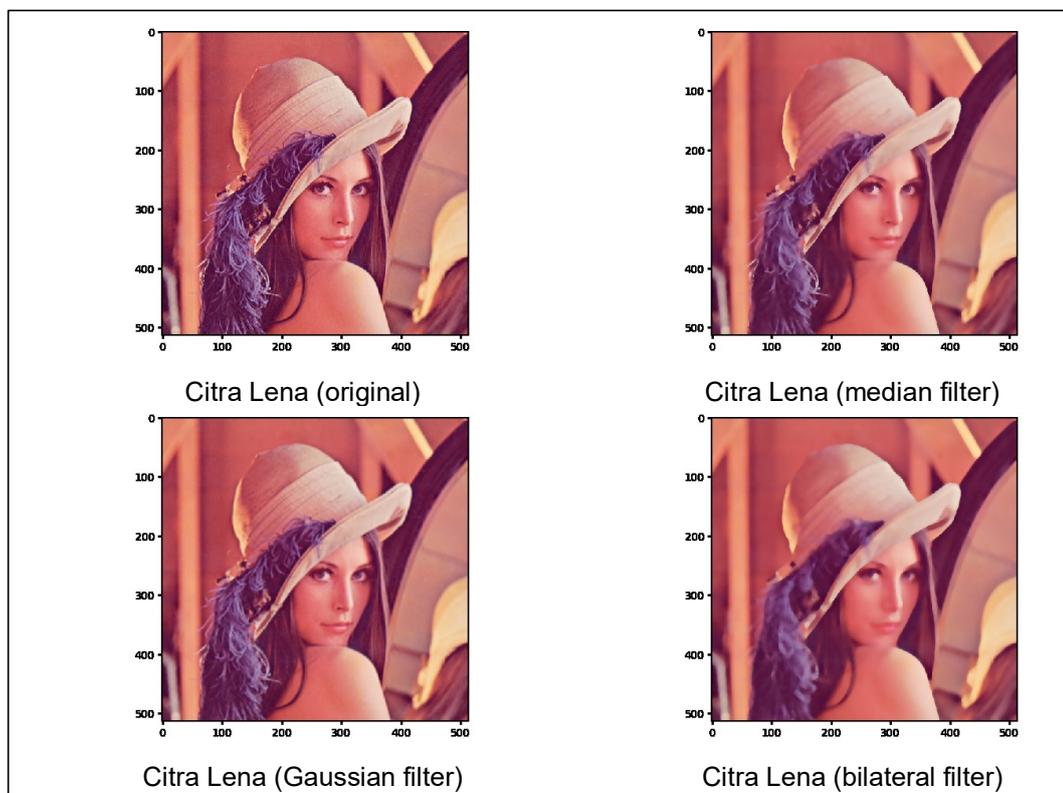
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

Banyaknya piksel yang terdapat dalam sebuah citra menyatakan resolusi citra tersebut. Semakin besar jumlah pikselnya maka semakin baik kualitas suatu citra. Resolusi citra dituliskan sebagai 320 x 240, 640 x 480, 800 x 600, 1024 x 768 dan seterusnya. Sebagai contoh 800 x 600 berarti terdapat 800 piksel baris dan 600 piksel kolom. Jumlah bit yang dibutuhkan untuk menyimpan citra digital adalah  $b = M \times N \times k$ . Di mana  $M$  = jumlah baris citra,  $N$  = jumlah kolom citra dan  $k$  = jumlah bit yang dibutuhkan untuk menyatakan sebuah nilai keabuan.

#### 4. Image filtering

*Filtering* adalah teknik untuk memodifikasi citra. Dengan mengaplikasikan *filter* pada sebuah citra maka fitur-fitur tertentu yang terdapat di dalam citra tersebut dapat ditonjolkan atau dihapus. Operasi *image processing* dengan *filtering* antara lain *smoothing*, *sharpening*, dan *edge enhancement*.

Smoothing dapat digunakan untuk pengaburan (*blurring*) dan mengurangi *noise* yang terdapat di dalam citra. *Blurring* biasa digunakan untuk menghapus detail kecil dari sebuah citra yang objeknya berukuran besar dan menjembatani jarak kecil pada garis atau kurva. Beberapa filter yang digunakan untuk smoothing di antaranya *median filter*, *gaussian filter* dan *bilateral filter* (lihat Gambar 4)



Gambar 4. Aplikasi *smoothing* pada citra Lena

Tidak seperti *smoothing*, *sharpening* atau operasi penajaman mencoba untuk menonjolkan detail yang ada di gambar. Untuk itu nilai piksel dibuat berbeda dengan piksel tetangganya.

*Filtering* merupakan operasi ketetanggaan (*neighborhood operation*) di mana nilai setiap piksel yang diberikan dalam gambar output ditentukan dengan menerapkan beberapa algoritma pada nilai-nilai piksel input tetangganya. Sebuah piksel  $p$  pada koordinat  $(x,y)$  mempunyai 4-tetangga horizontal dan vertikal dengan koordinat  $(x+1,y)$ ,  $(x-1,y)$ ,  $(x,y+1)$ ,  $(x,y-1)$ . Himpunan piksel ini disebut 4-tetangga dari  $p$  ( $N_4(p)$ ). Selain 4-tetangga horizontal dan vertikal juga terdapat tetangga diagonal dari  $p$  yang mempunyai koordinat  $(x+1,y+1)$ ,  $(x+1,y-1)$ ,  $(x-1,y+1)$ ,  $(x-1,y-1)$  dan dinotasikan dengan  $N_D(p)$  (Hermawati, 2013). Matriks ketetanggaan dapat dilihat pada Gambar 5.

	$(x-1,y)$	
$(x,y-1)$	$p$	$(x,y+1)$
	$(x+1,y)$	

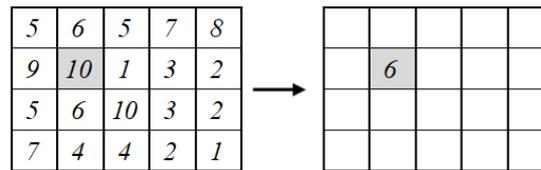
(a)

$(x-1,y-1)$		$(x-1,y+1)$
	$p$	
$(x+1,y-1)$		$(x+1,y+1)$

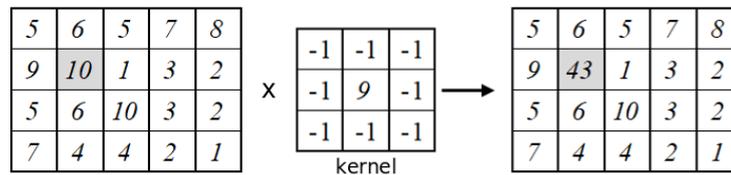
(b)

Gambar 5. Matriks ketetanggaan (a) horizontal dan vertikal, (b) diagonal

Ukuran piksel di mana fungsi manipulasi atau filter dijalankan disebut *kernel*. Pada Gambar 6 digunakan kernel berukuran 3 x 3 piksel untuk melakukan operasi *blurring* dan *sharpening*.



Bluring menggunakan Median filter



Sharpening menggunakan kernel 3x3

Gambar 6. Contoh kernel untuk *blurring* dan *sharpening*

Proses bluring pada Gambar 6 menggunakan median filter di mana kernel 3 x 3 yaitu 1,5,5,5,6,6,9,10,10 menghasilkan median 6 sehingga piksel (1,1) yang bernila 10 diganti menjadi 6. Sedangkan pada proses *sharpening* dilakukan operasi konvolusi antara kernel dan matriks ketetanggaan pada piksel (1,1).

## 5. Computer Vision

*Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati/diobservasi. *Computer vision* berhubungan dengan pencitraan otomatis dan pengolahan otomatis suatu citra berbasis komputer untuk mengestrak dan menginterpretasikan informasi. *Computer vision* bekerja dengan cara memproses data citra menggunakan kombinasi algoritma pengolahan citra dan kecerdasan buatan sehingga dapat menghasilkan informasi dan citra tersebut.

*Computer Vision* dalam melaksanakan tugas-tugas membutuhkan proses yang kompleks. Biasanya, untuk setiap aplikasi yang diberikan, keseluruhan tugas tidak dapat dilaksanakan pada sebuah tahapan tunggal. *Computer Vision* terdiri dari tahapan-tahapan seperti perolehan citra, *preprocessing*, pengekstraksian fitur, penyimpanan objek secara asosiatif, pengaksesan suatu basis pengetahuan, dan pengenalan (Robert B.Fisher, 2013).

## 6. *Pre-processing*

### Normalisasi

Normalisasi data memainkan peran penting dalam *preprocessing*. Hal ini diperlukan saat dihadapkan dengan data yang memiliki parameter unit dan skala yang berbeda. Dalam *image processing* normalisasi merupakan proses untuk mengubah kisaran nilai intensitas piksel. Nilai piksel pada sebagian besar data gambar ialah antara 0 - 255. Pada *neural network* nilai ini perlu dinormalisasi agar nilai piksel berada di antara 0 – 1. Hal ini membantu mempercepat konvergensi selama pembelajaran. Proses normalisasi dilakukan dengan membagi tiap nilai piksel dengan nilai piksel terbesar sebagaimana terlihat pada Persamaan 2:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

### *Mean subtraction*

*Mean subtraction* merupakan teknik persiapan data di mana nilai tiap piksel dikurangi dengan nilai rata-rata seluruh piksel sehingga distribusi nilai piksel dipusatkan pada nilai nol. Untuk itu pendekatan ini disebut

*centering* / pemusatan. Pemusatan  $x'$  mengharuskan nilai piksel rata-rata (mean pixel value)  $\mu$  dihitung sebelum mengurangkannya dari nilai piksel  $x$  seperti terlihat pada Persamaan 3:

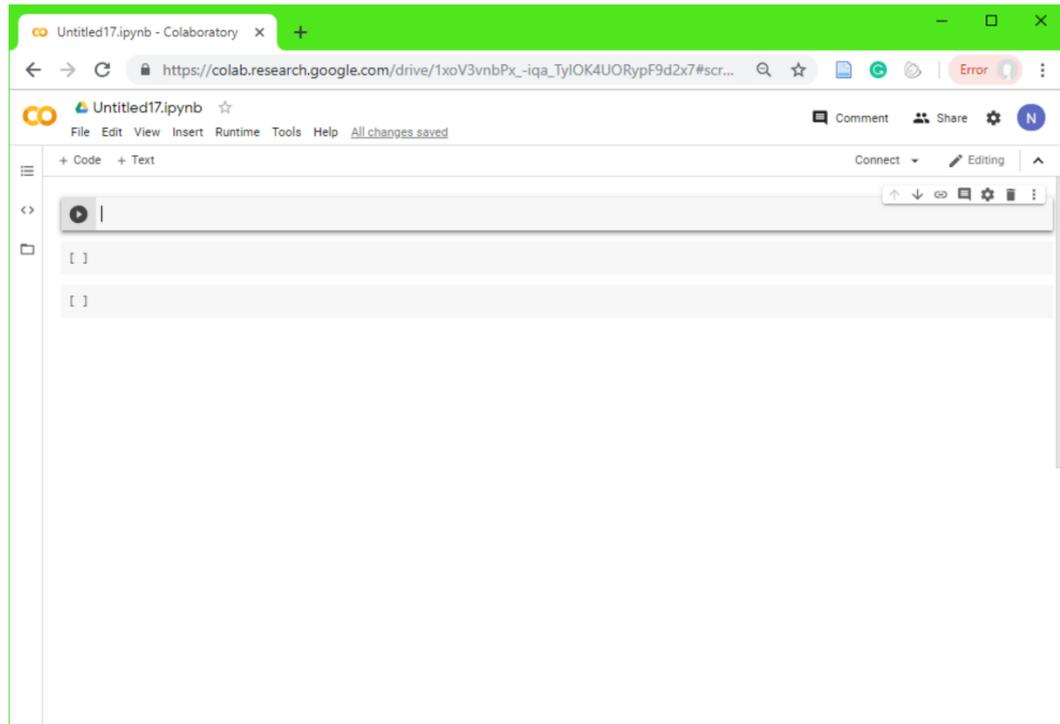
$$x' = x - \mu \quad (3)$$

Nilai rata-rata dapat dihitung untuk semua piksel dalam gambar (*global centering*) atau dihitung untuk setiap saluran (*local centering*) jika gambar yang digunakan berwarna.

## 7. Google Colaboratory

Google Colaboratory atau disebut juga Google Colab adalah *tools* yang dikeluarkan oleh *Google Internal Research*. *Tools* ini dibuat untuk membantu para Peneliti dalam mengolah data, khususnya di bidang *Machine Learning*. Dengan Google Colab pengguna dapat mengakses sumber daya yang powerful melalui browser secara bebas untuk mengeksekusi kode dan membangun aplikasi menggunakan berbagai *library* yang tersedia seperti Numpy, OpenCV, Keras, TensorFlow, dan lain-lain.

Google Colab seperti terlihat pada Gambar 7 memiliki tampilan yang mirip dengan Jupyter Notebook dan dibuat di atas *environment* Jupyter yang tidak memerlukan pengaturan terlebih dahulu sebelum digunakan dan berjalan sepenuhnya pada *cloud* dengan memanfaatkan media penyimpanan Google Drive (Digmi, 2018).



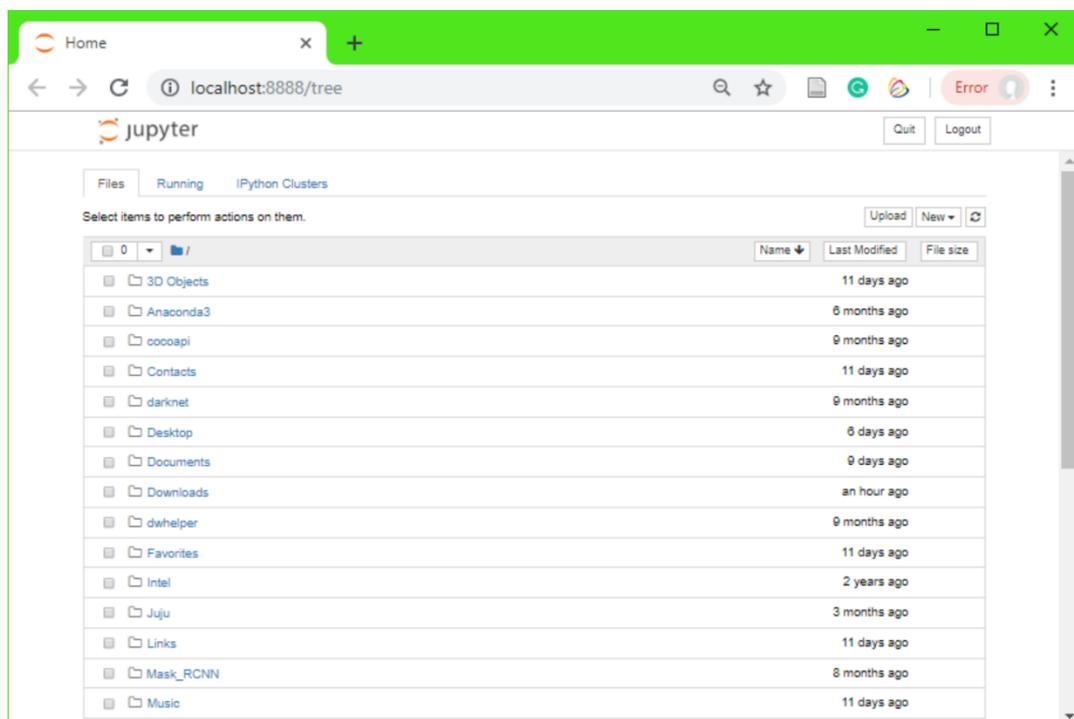
Gambar 7. Tampilan Google Colab

Selain bahasa python, terdapat beberapa perintah yang bisa digunakan di Google Colab seperti perintah git, pip dan sed. Git merupakan fasilitas *version control system* yang banyak digunakan oleh programmer untuk berbagi paket program dan membangun aplikasi secara bersama-sama dengan menggunakan perintah dasar seperti git init (membuat repositori/tempat penyimpanan paket program), git add (menambah file baru pada repositori), git clone (menyalin repositori), dll ("Git," n.d.).

Pip adalah salah satu pengelola paket python. Perintah pip merupakan baris program yang dapat dieksekusi di command prompt/penerjemah baris perintah yang ada di Windows ataupun python interpreter. Adapun sed merupakan stream editor dari Unix yang digunakan untuk manipulasi *script* tanpa harus membuka file sumbernya.

## 8. Jupyter Notebook

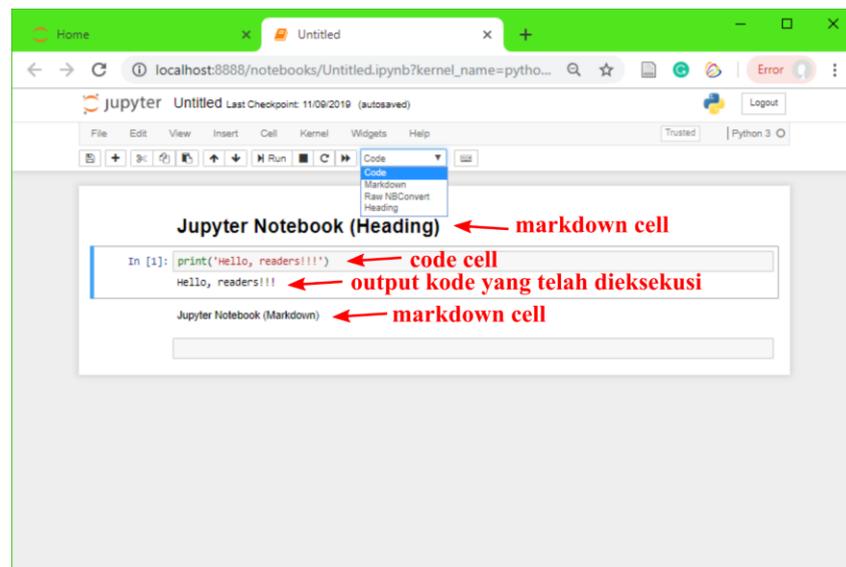
Jupyter Notebook merupakan aplikasi yang digunakan untuk membangun aplikasi untuk keperluan *scientific computation* seperti *data science*, *machine learning*, *data processing*, analisis, prediksi, dan lain sebagainya menggunakan bahasa pemrograman python. Jupyter Notebook memungkinkan penggunaanya untuk mengintegrasikan antara kode dengan output di dalam satu dokumen secara interaktif (Faqih, 2019).



Gambar 8. Tampilan halaman utama Jupyter Notebook

Jupyter Notebook merupakan aplikasi bawaan dari *platform* Anaconda yang berjalan di halaman *browser*. Anaconda sendiri merupakan platform berbasis *open source* untuk bahasa pemrograman Python dan R. Anaconda diciptakan untuk memudahkan pengguna dalam mengelola paket Python yang terdapat di sistem (Wijaya, 2019).

Saat membuka Jupyter Notebook maka yang pertama tampil adalah halaman utama sebagaimana terlihat pada Gambar 8. Adapun antarmuka dari Jupyter Notebook terdiri dari *cell* yang merupakan area di mana kode dan teks dituliskan. Untuk mengeksekusi kode dan menampilkan *output* dapat dilakukan dengan menekan SHIFT+ENTER pada *keyboard* atau tombol RUN di Toolbar pada *cell* aktif (Faqih, 2019). Pada Gambar 9 dapat dilihat bahwa *cell* untuk menulis dan mengeksekusi kode python disebut *code cell* sedangkan *cell* untuk menuliskan teks yang terformat disebut *markdown cell*.



Gambar 9. Tampilan antar muka Jupyter Notebook

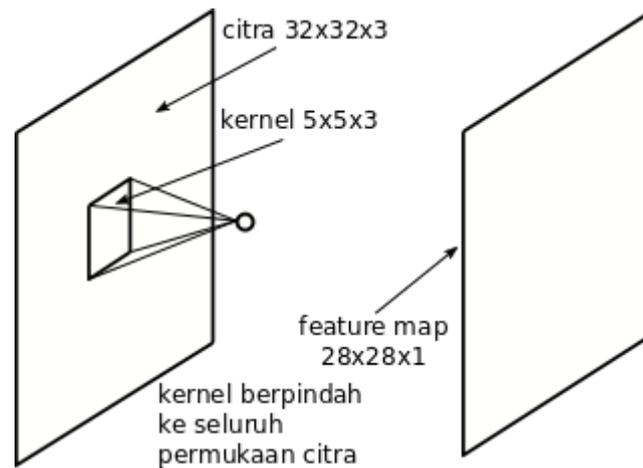
## 9. Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah kelas jaringan saraf tiruan yang menggunakan lapisan konvolusional dalam menyaring input untuk memperoleh informasi yang berguna. Aplikasi CNN mencakup berbagai sistem pemrosesan citra (*image recognition*, *image classification*,

pelabelan video, analisis teks) dan suara (*speech recognition*, *natural language processing*, klasifikasi teks), serta sistem AI seperti robot, asisten virtual, dan mobil *self-driving*.

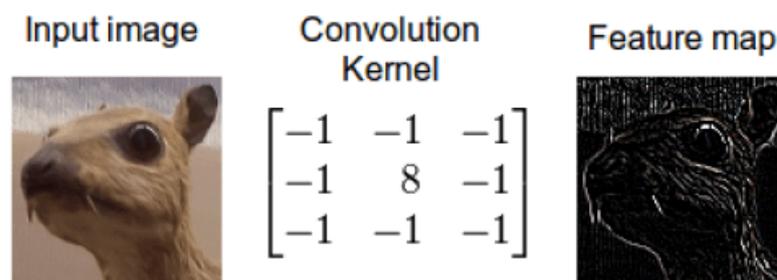
Jaringan konvolusional tersusun atas *input layer*, *output layer*, dan beberapa lapisan tersembunyi (*hidden layer*). CNN berbeda dari *artificial neural network* (ANN) yang mana inputnya berupa vektor. Pada CNN, lapisan inputnya berupa gambar yang diatur dalam tiga dimensi (dimensi lebar, tinggi, dan kedalaman). Dimensi kedalaman terdiri dari tiga matriks piksel, masing-masing matriks mewakili satu saluran warna merah, hijau dan biru. Sebuah piksel terdiri dari nilai integer antara 0 dan 255 di setiap saluran. *Hidden layer* dapat berupa *convolutional kernel* yang terdiri dari matriks tunggal. *Output layer* yang dihasilkan berupa gambar yang disebut *feature map*.

Pada Gambar 10 diperlihatkan input citra dengan dimensi  $32 \times 32 \times 3$  kemudian diberikan filter  $5 \times 5 \times 3$  yang akan bergeser sesuai besar *stride* yang telah ditentukan untuk memindai gambar. *Stride* adalah parameter yang menentukan berapa jumlah pergeseran filter. Setiap pergeseran akan diperoleh sebuah nilai yang merupakan hasil operasi konvolusi antara filter dan potongan gambar.



Gambar 10. Ilustrasi operasi konvolusi (Pokharna, 2016)

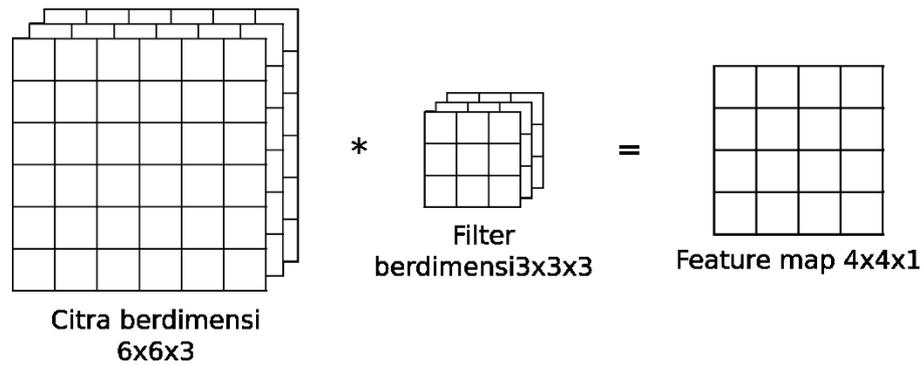
Contoh hasil konvolusi menggunakan *convolutional kernel* pendeteksi tepi pada CNN dapat dilihat pada Gambar 11 berikut:



Gambar 11. Contoh Hasil konvolusi menggunakan sebuah kernel konvolusi (Dettmers, 2015)

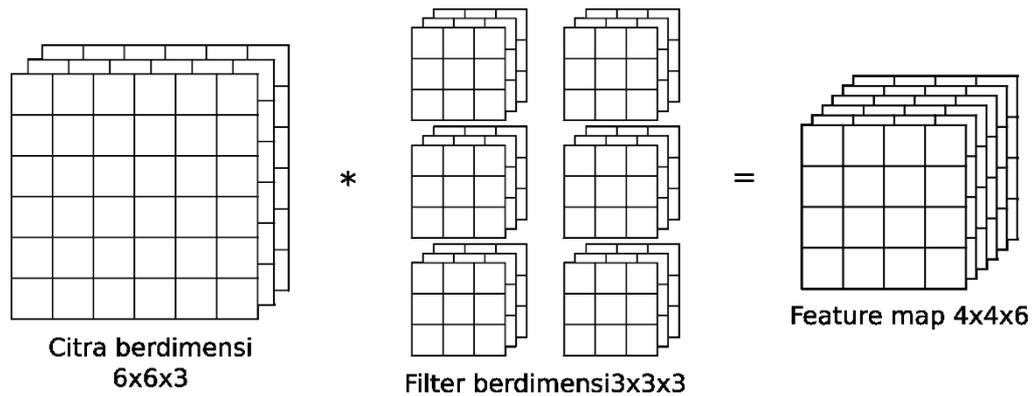
#### a. **Convolutional Layers**

*Hidden layer* adalah kombinasi dari lapisan konvolusi (*convolutional layers*), lapisan penyatuan (*pooling layers*), lapisan normalisasi (*normalization layer*), dan lapisan yang terhubung sepenuhnya (*fully connected layers*).



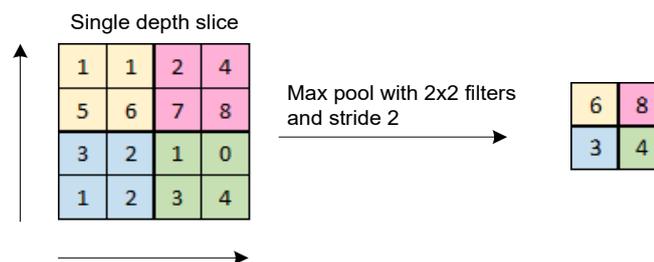
Gambar 12. *Convolutional layer* dengan 1 filter

Lapisan konvolusional terdiri dari satu set filter independen. Jika terdapat suatu citra berdimensi  $6 \times 6 \times 3$  dan dikonvolusi menggunakan sebuah filter berdimensi  $3 \times 3 \times 3$ , maka akan dihasilkan sebuah *feature map* yang berukuran  $4 \times 4 \times 1$  sebagaimana terlihat pada Gambar 12. Jika citra tersebut dikonvolusi menggunakan satu set filter terdiri atas 6 filter berdimensi  $3 \times 3 \times 3$ , maka setiap filter secara independen dikonvolusi dengan gambar dan menghasilkan 6 *feature map* yang masing-masing berukuran  $4 \times 4 \times 1$  sebagaimana terlihat pada Gambar 13. *Convolutional layer* akan menghitung output dari neuron yang terhubung ke daerah lokal dalam input, masing-masing menghitung produk titik antara bobot mereka dan wilayah kecil yang terhubung ke dalam volume input. *Convolutional layer* bertindak sebagai filter matematika yang membantu komputer menemukan tepi gambar, area gelap dan terang, warna, dan detail lainnya, seperti tinggi, lebar dan kedalaman.



Gambar 13. *Convolutional layer* dengan 6 filter

*Pooling layer* berfungsi untuk mengurangi dimensi dari *feature map* atau lebih dikenal dengan langkah untuk *downsampling*, sehingga mempercepat komputasi karena parameter yang harus diupdate semakin sedikit dan mengatasi *overfitting*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. *Max Pooling* digunakan untuk menentukan nilai maksimum tiap pergeseran filter seperti yang terlihat pada Gambar 14, sementara *Average Pooling* akan menentukan nilai rata-ratanya. *Pooling layer* beroperasi pada setiap *feature map* secara independen.



Gambar 14. *Max pooling* (Pokharna, 2016)

*Activation function* berfungsi untuk membuat *neural network* menjadi *non-linear*. Beberapa fungsi aktivasi yang sering digunakan diantaranya Sigmoid, Tanh dan *Rectified Linear Unit* (ReLU).

*Fully-connected* (FC) *layer* adalah lapisan neuron yang terdapat pada lapisan terakhir CNN. Neuron dalam FC *layer* memiliki koneksi penuh ke semua fungsi aktivasi di lapisan sebelumnya. FC *layer* menentukan hubungan antara posisi fitur dalam gambar dan kelas.

#### **b. *Stride***

*Stride* adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *convolutional filter* akan bergeser sebanyak 1 piksel secara horizontal lalu vertikal.

Semakin kecil *stride*, maka akan semakin detail informasi yang diperoleh dari sebuah input, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Namun penggunaan *stride* yang kecil tidak selalu akan mendapatkan performa yang bagus.

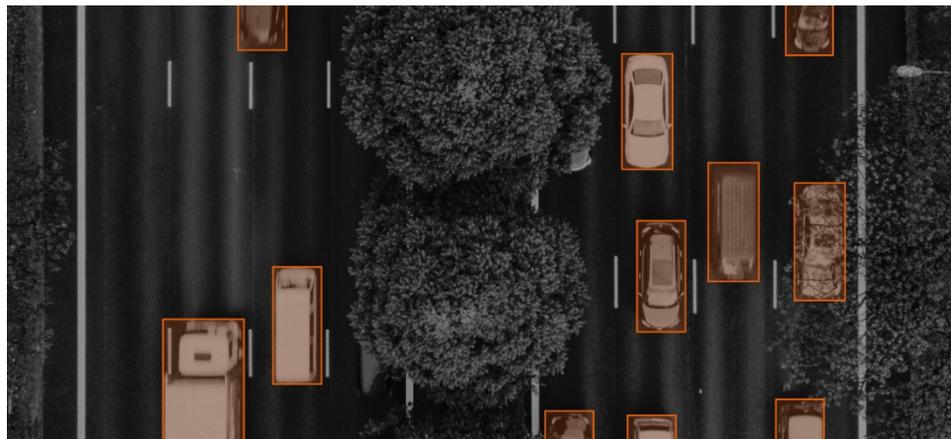
### **10. *Image Annotation***

*Image annotation* ialah pemberian anotasi berupa label pada gambar. Label-label ini telah ditentukan sebelumnya dan dipilih untuk memberikan informasi berupa kelas pada model visi komputer tentang apa yang ditampilkan dalam gambar. Pemberian anotasi pada setiap gambar dapat bervariasi. Pada beberapa kasus (klasifikasi gambar) hanya diperlukan satu anotasi untuk mewakili konten seluruh gambar dan pada kasus beberapa objek dalam gambar diberi anotasi berbeda sesuai dengan kelasnya. Pada

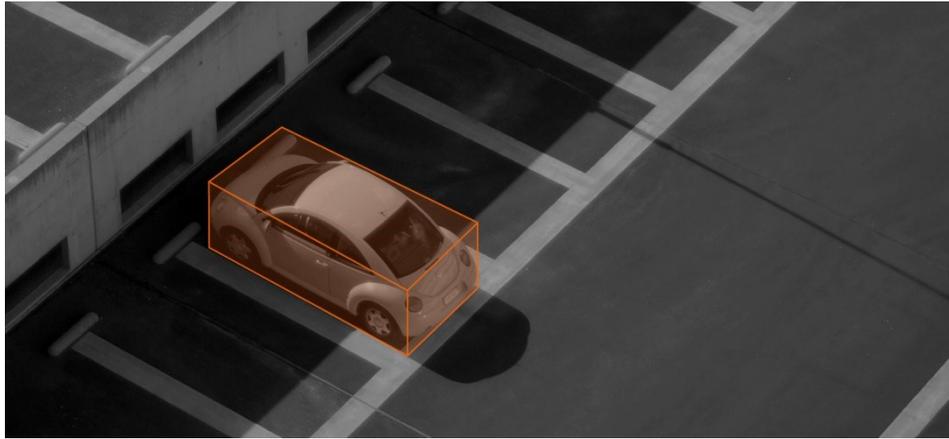
tesis ini pemberian anotasi dilakukan pada objek tertentu dalam hal ini tunas air.

Untuk melakukan anotasi, diperlukan tiga hal, yaitu: (1) gambar; (2) orang yang melakukan anotasi pada gambar; (3) *platform* untuk anotasi gambar.

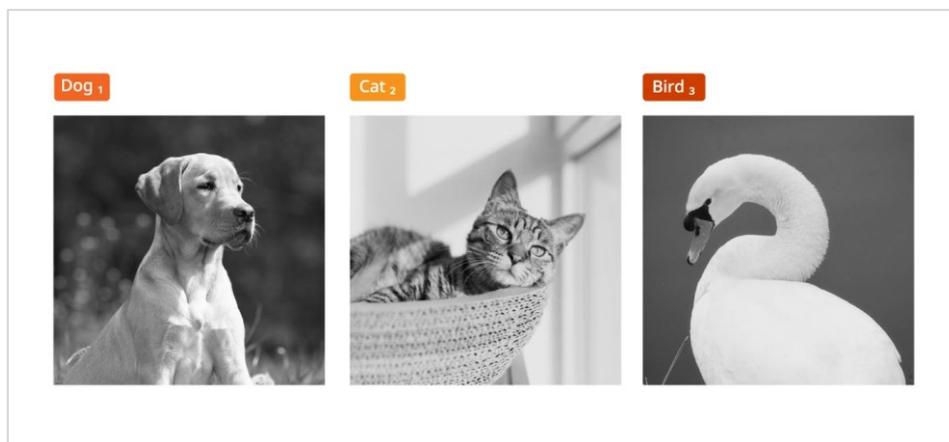
Terdapat 5 macam anotasi gambar yang biasa dipakai dalam visi komputer, di antaranya yaitu: *2D and 3D bounding boxes* (Gambar 15 dan Gambar 16), *Image classification* (Gambar 17), *Lines and Splines* (Gambar 18), *polygon* dan *semantic segmentation* (Gambar 19 dan Gambar 20) (Ambalina, 2019).



Gambar 15. *2D bounding boxes* (Ambalina, 2019)



Gambar 16. *3D bounding box* (Ambalina, 2019)



Gambar 17. *Image classification* (Ambalina, 2019)



Gambar 18. *Lines and splines* (Ambalina, 2019)



Gambar 19. *Polygon* (Ambalina, 2019)



Gambar 20. *Semantic segmentation* (Ambalina, 2019)

## 11. *Residual Network (ResNet)*

Tujuan utama dari *Residual Network* (ResNet) ialah untuk membangun *neural network* yang lebih dalam (jaringan dengan banyak lapisan). Semakin dalam sebuah jaringan saraf *convolutional* maka *training error* secara bertahap terus berkurang (Ruiz, 2019). Namun, untuk jaringan saraf tradisional, pada kedalaman tertentu *training error* akan kembali meningkat. Ini dapat diatasi dengan pemetaan identitas (Maladkar, 2018).

*Neural network* harus dapat dengan mudah menyelesaikan fungsi identifikasi, di mana output dari suatu fungsi  $f(x)$  menjadi input itu sendiri seperti pada Persamaan 4 (Ruiz, 2019).

$$f(x) = x \quad (4)$$

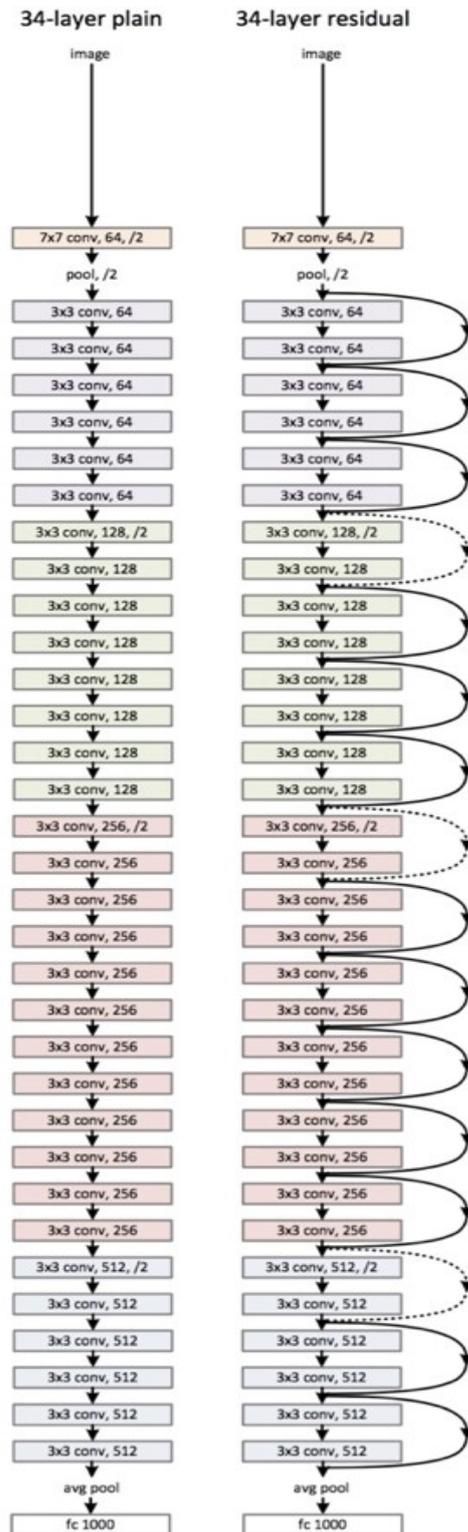
Dengan logika yang sama, jika mem-*bypass* input ke lapisan pertama dari model untuk menjadi output dari lapisan terakhir dari model, jaringan dapat memprediksi fungsi apa pun yang sedang dipelajari sebelum input ditambahkan ke dalamnya sebagaimana terlihat pada Persamaan 5 (Ruiz, 2019).

$$f(x) + x = h(x) \quad (5)$$

Dengan ResNets, gradien dapat mengalir langsung melalui *skip connection* dari filter awal ke lapisan berikutnya (Ruiz, 2019).

ResNets dapat memiliki ukuran variabel, tergantung pada seberapa besar masing-masing lapisan model, dan berapa banyak lapisan yang dimilikinya (Ruiz, 2019).

Menurut Kaiming He *et al.*, ResNet (Gambar 21) terdiri dari satu konvolusi dan *pooling* (oranye) diikuti oleh 4 lapisan yang serupa di mana setiap lapisan mengikuti pola yang sama. Masing-masing melakukan konvolusi 3x3 dengan dimensi *feature map* ( $F$ ) [64, 128, 256, 512], melewati input setiap 2 konvolusi. Selain itu, dimensi lebar ( $W$ ) dan tinggi ( $H$ ) tetap konstan di seluruh lapisan (He *et al.*, 2015)



Gambar 21. Perbandingan *plain network* dan *residual network*

Garis putus-putus pada Gambar 21 menunjukkan bahwa telah ada perubahan dalam dimensi volume input (disebabkan oleh pengurangan saat konvolusi). Dapat diperhatikan bahwa pengurangan antar lapisan ini dicapai dengan peningkatan pada langkahnya, dari 1 menjadi 2, pada lilitan pertama dari setiap lapisan, bukan dengan operasi penyatuan, yang biasa disebut sebagai *down sampling*.

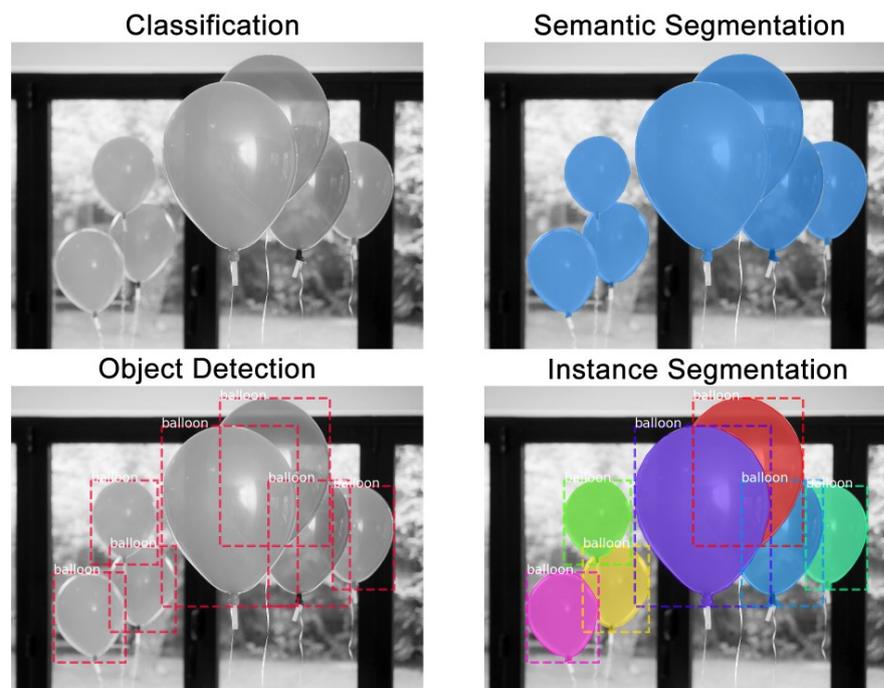
Secara umum, *deep convolutional neural network* terdiri dari banyak lapisan yang ditumpuk dan dilatih untuk melakukan tugas yang diberikan. Jaringan konvolusionel mempelajari fitur-fitur level rendah hingga level tinggi dan pada akhirnya memperoleh *convolutional feature map* (peta fitur konvolusi) yang memiliki dimensi spasial yang jauh lebih kecil dari gambar asli, tetapi memiliki kedalaman yang lebih besar.

Peta fitur konvolusi mengkodekan semua informasi untuk gambar namun tetap mempertahankan lokasi fitur yang telah dikodekan relatif terhadap gambar asli. Sebagai contoh, jika terdapat titik merah di sisi kanan atas pada gambar dan lapisan konvolusionel mengaktifkannya, maka informasi untuk titik merah itu akan tetap di sisi kanan atas peta fitur konvolusionel.

## **12. Mask R-CNN**

Mask R-CNN adalah *framework deep learning* yang diperkenalkan oleh tim *Facebook AI Research* (FAIR) pada tahun 2017. *Framework* ini dapat melakukan deteksi objek pada gambar yang sekaligus menghasilkan *mask segmentation* untuk setiap *instance* atau biasa disebut dengan *instance segmentation* (He et al., 2018). Berdasarkan Gambar 22 dapat

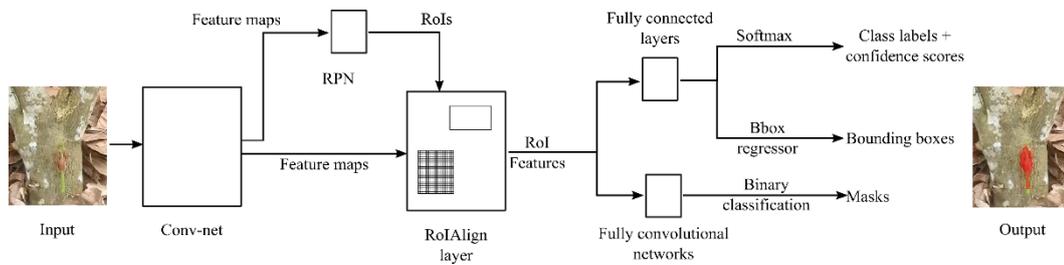
dilihat bahwa *instance segmentation* mirip dengan *object detection* di mana masing-masing objek dideteksi satu per satu namun kemudian dikombinasikan dengan *semantic segmentation* sehingga setiap objek dapat diketahui kelas, lokasi serta dapat dibedakan pikselnya.



Gambar 22. *Instance segmentation* (Abdulla, 2018)

Dalam melakukan deteksi, Mask R-CNN dapat dibagi menjadi tiga bagian, yaitu: (i) *feature extraction network*, (ii) *region-proposal network*, dan (iii) *instance detection and segmentation networks* sebagaimana terlihat pada Gambar 23 (Shi *et al.*, 2019). Untuk mengekstrak fitur, Mask R-CNN mengaplikasikan arsitektur *multiple backbone*. Beberapa *backbone* yang digunakan untuk Mask R-CNN ialah ResNet, ResNext serta FPN. Melalui uji coba, kombinasi ResNet-FPN menghasilkan akurasi dan kecepatan di atas rata-rata dalam mengekstrak fitur. Pada proses *Region*

*Proposal Network* (RPN) akan dihasilkan *Region of Interest* (RoI) yang selanjutnya menjadi input pada tahap *instance detection and segmentation networks*.



Gambar 23. Arsitektur Mask R-CNN

Pada proses *training Mask R-CNN* terdapat beberapa parameter yang diatur antara lain *epoch* atau banyaknya siklus pelatihan, *steps per epoch* atau banyaknya iterasi per *epoch*, *validation step* atau banyaknya validasi yang dilakukan di akhir setiap *epoch* (semakin banyak validasi yang dilakukan dapat meningkatkan akurasi tetapi dapat memperlambat proses training), *backbone* arsitektur jaringan, jumlah kelas, dimensi maximum dan minimum citra (untuk proses *resize* dan *padding*), *RPN NMS threshold* dan *mean pixel* dari dataset latih.

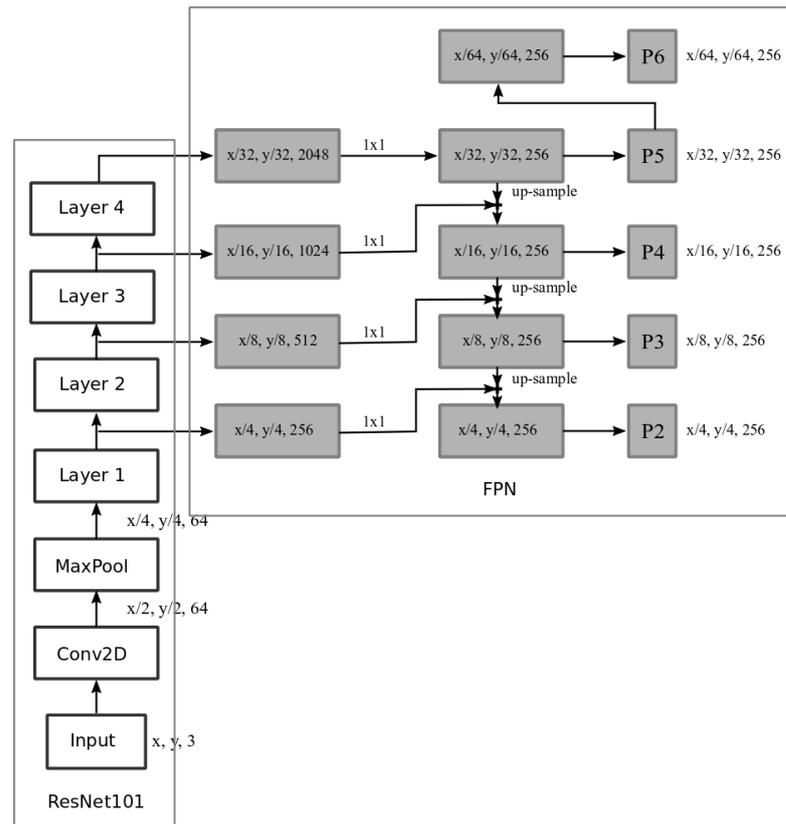
#### a. **Feature extraction**

Ekstraksi fitur atau *feature extraction* bertujuan untuk mendapatkan informasi yang terdapat dalam citra tunas air dan merepresentasikan informasi tersebut dalam ruang dimensi yang lebih rendah sehingga tugas klasifikasi pola menjadi mudah.

*Mask R-CNN* menggunakan kombinasi arsitektur ResNet101 dan FPN (*Feature Pyramid Network*) untuk menghasilkan fitur RoI saat ekstraksi fitur dilakukan. FPN adalah komponen dasar dalam sistem rekognisi untuk mendeteksi objek pada skala yang berbeda menggunakan gambar yang sama.

FPN menggunakan *feature map* yang bervariasi sehingga menghasilkan informasi yang lebih berkualitas. FPN merupakan *feature extractor* yang didesain menggunakan konsep piramid namun lebih unggul dalam hal kecepatan dan akurasi. FPN memproses informasi dengan dua cara yaitu dari bawah ke atas (*bottom-up*) dan dari atas ke bawah (*top-down*).

Pemrosesan data *bottom-up* mengekstrak fitur menggunakan ResNet, yang mana semakin ke atas dimensi spasial setiap lapisan semakin berkurang sedangkan nilai semantiknya semakin bertambah. Pemrosesan secara *top-down* meningkatkan resolusi lapisan semantik namun lokasi objek tidak presisi. FPN menambahkan koneksi lateral antar lapisan yang direkonstruksi dan *feature map* yang sesuai untuk membantu detektor memprediksi lokasi dengan lebih baik. Koneksi lateral adalah operasi konvolusi dan penjumlahan antara dua level yang sesuai dari kedua *pathway*. FPN melampaui *single ConvNet* utamanya karena mempertahankan fitur semantik pada berbagai resolusi.



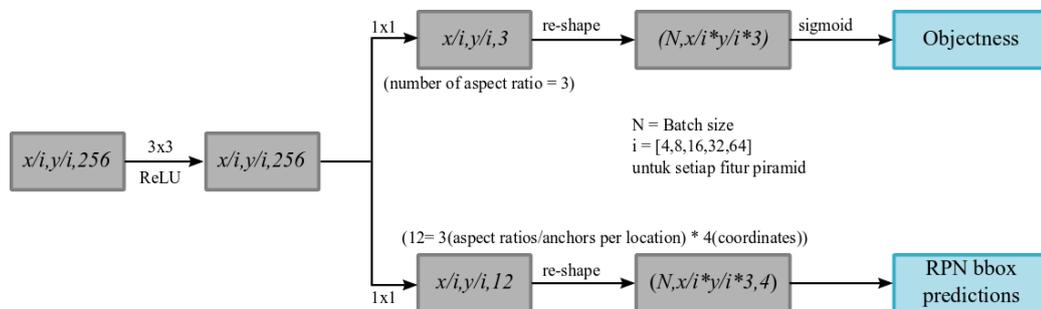
Gambar 24. Ekstraksi fitur menggunakan ResNet101-FPN

Pada Gambar 24 dapat dilihat proses ekstraksi di mana ukuran setiap lapisan *feature map* dikurangi setengahnya sedangkan jumlah *feature map* digandakan, menjadi dua kali lipat. Dari ResNet101 diekstrak empat *feature map* (*layer-1*, *layer-2*, *layer-3* dan *layer-4*). Untuk menghasilkan *feature map* terakhir digunakan pendekatan yang disebut *top-bottom pathway*. Pendekatan *top-bottom pathway* dimulai dari *feature map* teratas ( $x/32, y/32, 256$ ) dan mulai turun ke *feature map* yang lebih besar, dengan mengaplikasikan operasi *up-sample*. Konvolusi  $1 \times 1$  dilakukan untuk menurunkan jumlah saluran ke 256 sebelum pengambilan sampel. Kemudian ditambahkan elemen ke *output up-sample* dari iterasi sebelumnya. Semua *output up-sample* diaplikasikan lapisan konvolusi  $3 \times$

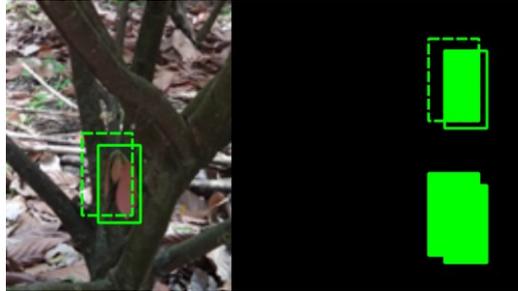
3 untuk menghasilkan 4 *feature map* terakhir (P2, P3, P4, P5) sedangkan *feature map* ke-5 (P6) dihasilkan dari operasi *max pooling* P5.

#### b. RPN (*Region Proposal Network*)

Setiap *feature map* yang dihasilkan pada proses ekstraksi fitur akan melalui lapisan konvolusi 3 x 3. Tetapi sebelum itu *feature map* tersebut dipindai terlebih dahulu menggunakan anchor box dengan berbagai skala dan rasio. Output yang dihasilkan kemudian diteruskan ke dua cabang, satu berkaitan dengan skor objektivitas atau *confidence score* dan lainnya untuk *bounding box regressor* seperti terlihat pada Gambar 25. *Confidence score* dapat diperoleh dengan menghitung IoU (*intersection over union*) antara *bounding box* dengan *ground-truth* seperti terlihat pada Gambar 26 di mana *ground-truth* ditandai dengan garis solid sedangkan *bounding box* ditandai dengan garis putus-putus. IoU diperoleh dengan membagi luas area yang saling berhimpit dengan luas area gabungan *ground truth* dan *bounding box*. *Confidence score* berkisar antara 0 hingga 1. Semakin besar *Confidence score* berarti semakin tinggi keyakinan sistem bahwa objek yang terdapat di dalam *bounding box* merupakan objek yang hendak dideteksi.



Gambar 25. Proses RPN



Gambar 26. Menghitung IoU

**c. Instance detection dan semantic segmentation**

Proses *instance segmentation* dilakukan dengan menggunakan *fully-connected network* yang mengambil RoI sebagai inputan untuk mendeteksi keberadaan objek, *bounding box*, *class label* dan nilai *confidence*. *Fully Convolutional Network* (FCN) digunakan untuk melakukan *semantic segmentation* pada citra dengan memprediksi *semantic class* dari masing-masing piksel pada *bounding box*. Inilah yang menyebabkan setiap *instance* dapat menampilkan warna yang berbeda sesuai dengan *bounding box*nya.

**B. Penelitian Terkait**

Beberapa penelitian terkait identifikasi pemangkasan cabang di antaranya sebagai berikut:

- a. Penelitian sebelumnya oleh Amean *et al.*, melakukan segmentasi dan perhitungan otomatis pada cabang tanaman *Hibiscus* berdasarkan ukuran cabang dan ranting. Algoritma yang digunakan ialah Hough Transform untuk mendeteksi dan menggambar garis pada ranting dan cabang, sedangkan Frangi 2D *filter* digunakan untuk klasifikasi piksel

pada gambar. Hasil penelitian menunjukkan metode yang digunakan dapat mendeteksi dan menghitung cabang tanaman secara otomatis (Amean *et al.*, 2013).

- b. Karkee *et al.* melakukan identifikasi pemangkasan cabang pada pohon apel menggunakan kamera tiga dimensi untuk pengambilan gambar dan *medial axis thinning algorithm* untuk menghasilkan kerangka 3D pohon apel. Hasil penelitian menunjukkan perhitungan algoritma yang digunakan dan pemangkasan oleh ahli menghasilkan nilai rata-rata yang serupa pada jarak antar cabang tanaman apel (Karkee *et al.*, 2014).
- c. Akbar *et al.* melakukan rekonstruksi 3D pada pohon apel menggunakan informasi kedalaman memanfaatkan sensor kamera Kinect 2. Algoritma Levenberg-Marquardt digunakan untuk membagi titik awan pohon yang diperoleh dari gambar kedalaman menjadi beberapa penampang untuk menghitung diameter cabang utama pada beragam nilai kedalaman. Hasil penelitian menunjukkan bahwa skema yang diusulkan memberikan kinerja sebesar 89% untuk memperkirakan dengan tepat diameter cabang utama (Akbar *et al.*, 2016).
- d. El Mesalami *et al.* melakukan deteksi batang dan cabang utama pada tanaman tomat yang ditanam menggunakan *high-wire cultivation system* dengan mengkombinasikan algoritma segmentasi berbasis Frangi dan *Distance Regularized Level Set Evolution* (DRLSE). Algoritma segmentasi yang digunakan secara efisien mendeteksi

batang dan cabang utama tanaman yang ditanam di rumah kaca dan mampu mengenali tanaman target dengan latar belakang berantakan. Pengurangan signifikan dalam waktu eksekusi dicapai melalui pelacakan objek target pada gambar yang dinormalisasi untuk menghilangkan efek cahaya yang tidak seimbang di dalam rumah kaca.(El Mesalami *et al.*, 2018).

- e. Elfiky *et al.* melakukan penelitian tentang penggunaan fitur *Geometric Skeleton-based* (SbG) dalam skema rekonstruksi 3D. Hasil yang diperoleh menunjukkan keefektifan fitur SbG untuk rekonstruksi otomatis hanya dengan menggunakan dua tampilan, depan dan belakang. Hasil kami menunjukkan bahwa skema yang diusulkan menempatkan titik pemangkasan pada cabang pohon dengan akurasi 96.0% (Elfiky *et al.*, 2015).
- f. Chattopadhyay *et al.* menyajikan skema rekonstruksi 3D berbasis setengah lingkaran untuk pemodelan batang dan cabang utama pohon apel. Metode ini melibatkan estimasi kesalahan diameter, membuat model pohon setengah lingkaran dari gambar kedalaman tunggal, dan merekonstruksi model 3D akhir pohon dengan menyelaraskan urutan gambar kedalaman (Chattopadhyay *et al.*, 2016).
- g. Amatya *et al.* meneliti sistem visi mesin untuk segmentasi dan deteksi cabang-cabang pohon ceri dengan dedaunan penuh, di mana ranting pohon terlihat terputus-putus. Metode segmentasi gambar

dikembangkan untuk mengidentifikasi segmen cabang yang terlihat (Amatya *et al.*, 2016).

### C. State of The Art

Penelitian sebelumnya oleh Amean *et al.*, melakukan segmentasi dan perhitungan otomatis pada cabang tanaman *Hibiscus*. Karkee *et al.* melakukan identifikasi pemangkasan cabang pada pohon apel dengan struktur tinggi dan ramping tanpa daun yang banyak (Karkee *et al.*, 2014). Mirip dengan penelitian sebelumnya, Akbar *et al.*, Elfiky *et al.* dan Chattopadhyay *et al.* melakukan rekonstruksi 3D pada pohon apel dengan struktur tinggi dan ramping tanpa daun yang banyak (Akbar *et al.*, 2016; Chattopadhyay *et al.*, 2016; Elfiky *et al.*, 2015). El Mesalami *et al.* melakukan deteksi batang dan cabang utama pada tanaman tomat yang ditanam di dalam rumah kaca (El Mesalami *et al.*, 2018). Sedangkan Amatya *et al.* melakukan deteksi cabang pada tanaman ceri menggunakan metode Bayesian *classifier* (Amatya *et al.*, 2016).

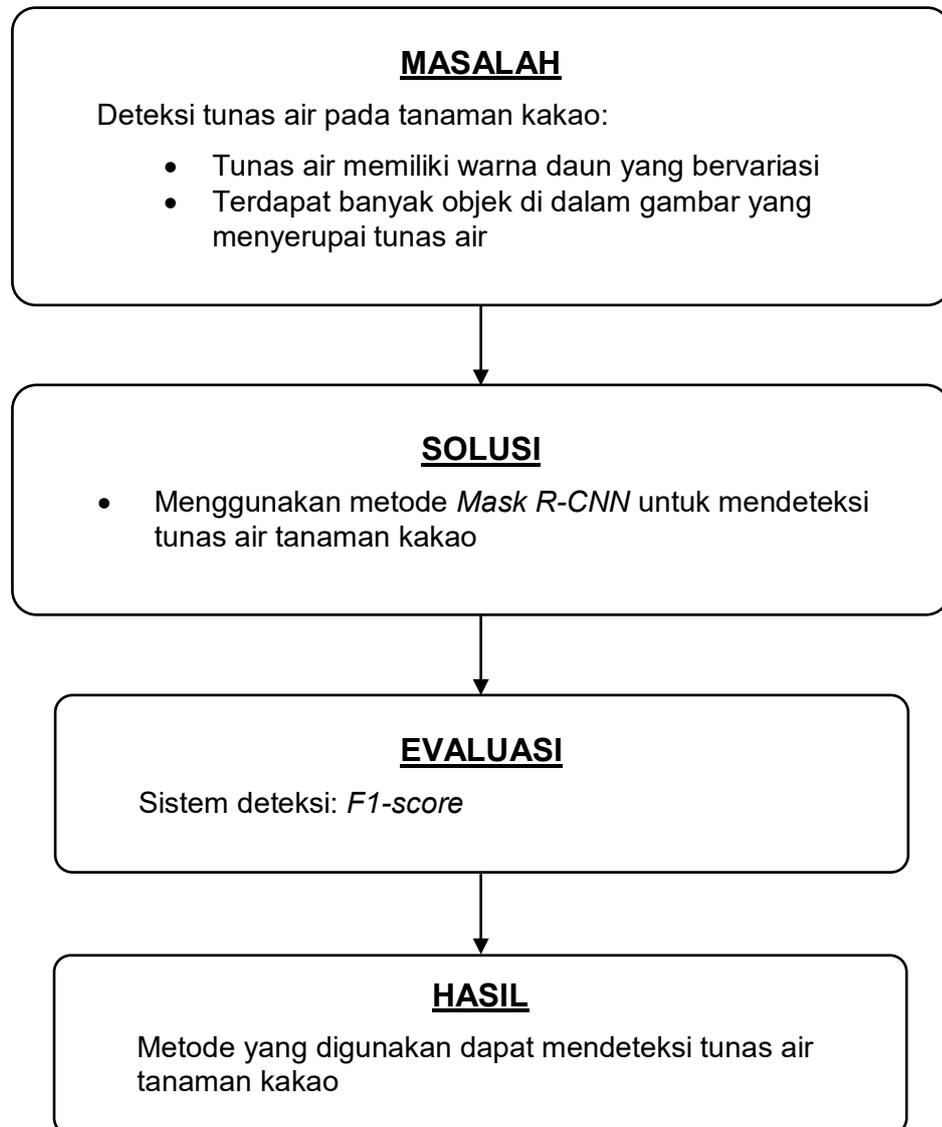
Pada penelitian ini menggunakan objek tanaman kakao yang ditanam di perkebunan di mana daun dapat ditemukan di sepanjang cabang yang baru tumbuh sehingga penampakan cabang terputus-putus. Oleh sebab itu pada penelitian ini akan digunakan metode Bayesian *classifier*.

Tabel 1. *State of the art*

No	Judul	Penerbit, Penulis & Tahun	Metode	Hasil	Jumlah Data
1.	<i>Automatic plant branch segmentation and classification using vesselness measure</i>	<i>Proceedings of Australasian Conference on Robotics and Automation.</i> Amean <i>et al.</i> 2013	<ul style="list-style-type: none"> <li>• Hough Transform techniques</li> </ul> Frangi 2D filter	Sistem menunjukkan persentase yang tinggi dalam segmentasi cabang tanaman	12 gambar
2.	<i>Identification of pruning branches in tall spindle apple trees for automated pruning.</i> (10.1016/j.compag.2014.02.013)	<i>Computer Electron. Agric.</i> Karkee <i>et al.</i> 2014	<ul style="list-style-type: none"> <li>• Time-of-flight (TOF) 3D camera</li> </ul> Thinning algorithm	algoritma yang digunakan dan pemangkasan manual menghasilkan jarak cabang rata-rata yang serupa.	400 gambar
3.	<i>A novel framework for modeling dormant apple trees using single depth image for robotic pruning application</i> (10.1109/ICRA.2016.7487718)	<i>IEEE International Conference on Robotics and Automation (ICRA).</i> Akbar <i>et al.</i> 2016	<ul style="list-style-type: none"> <li>• Algoritma Levenberg-Marquardt</li> <li>• 3D reconstruction model</li> </ul>	Ketepatan mencapai 89%	Data diambil dari 6 pohon
4.	<i>Automatic Detection of the Main Vine and Branches of Tomato Plants Grown in Greenhouses</i> (10.1109/EIT.2018.8500146)	<i>IEEE International Conference on Electro/Information Technology (EIT).</i> El Mesalami <i>et al.</i> 2018	<ul style="list-style-type: none"> <li>• Distance Regularized Level Set Evolution</li> <li>• (DRLSE)</li> <li>• Frangi filter</li> </ul>	System mampu mendeteksi batang dan cabang tanaman secara efisien	-

5.	<i>Automation of dormant pruning in specialty crop production: An adaptive framework for automatic reconstruction and modeling of apple trees</i>	2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Elfiky et al. 2015	<ul style="list-style-type: none"> <li>• <i>Skeleton based geometric</i></li> <li>• <i>ICP algorithm</i></li> </ul>	Akurasi 96 % dalam menempatkan titik pemangkasan pada cabang pohon	-
6.	<i>Measuring and modeling apple trees using time-of-flight data for automation of dormant pruning applications</i>	2016 IEEE Winter Conference on Applications of Computer Vision (WACV). Chattopadhyay et al. 2016	<ul style="list-style-type: none"> <li>• <i>3D point cloud</i></li> <li>• <i>Voronoi diagram technique</i></li> </ul>	Kinerja rata-rata, 89.4% untuk memperkirakan diameter cabang utama dengan toleransi 5 mm, dan 100% untuk mengidentifikasi cabang dengan benar.	175 <i>depth image</i>
7.	<i>Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting</i>	<i>Biosystems engineering xxx</i> (2015) 1 e1 3. Amatya et al. 2015	<ul style="list-style-type: none"> <li>• <i>Bayesian classifier</i></li> </ul>	Akurasi deteksi 89.2%	141 citra

#### D. Kerangka Pikir



Gambar 27. Kerangka pikir

Gambar 27 menunjukkan kerangka pikir pada penelitian ini. Pada tahap pertama menjelaskan permasalahan dalam identifikasi tunas air pada tanaman kakao yaitu, tunas air kakao memiliki warna yang bervariasi, ada yang menyerupai daun muda (*flush*) tanaman kakao dan ada juga yang mirip warna daun kering. Kemudian, solusi yang ditawarkan ialah dengan

mengurangi objek yang menyerupai tunas air ialah dengan memotong area gambar yang banyak ditemukan objek menyerupai tunas air seperti bagian atas tanaman yang didominasi daun hijau dan bagian bawah yang banyak ditemukan daun kering, sedangkan untuk deteksi tunas air itu sendiri digunakan metode *deep learning* yaitu *Mask R-CNN*. Evaluasi dilakukan dengan menghitung persentase gambar yang tetap menampilkan tunas air meskipun telah dilakukan pemotongan gambar. Sedangkan untuk sistem deteksi dievaluasi dengan menghitung nilai *F1-score* sistem.