

SKRIPSI

**DETEKSI DAN HITUNG BUAH COKELAT MATANG DAN
NON-MATANG PADA POHONNYA SECARA *REALTIME***

Disusun dan diajukan oleh:

MUHAMMAD RAEDI RADIFAN

D42116304



**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2022**

LEMBAR PENGESAHAN SKRIPSI
DETEKSI DAN HITUNG BUAH COKELAT MATANG DAN
NON-MATANG PADA POHONNYA SECARA REALTIME

Disusun dan diajukan oleh

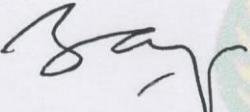
MUHAMMAD RAEDI RADIFAN

D42116304

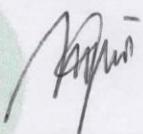
Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 01 November 2022 dan dinyatakan
telah memenuhi syarat kelulusan.

Menyetujui,

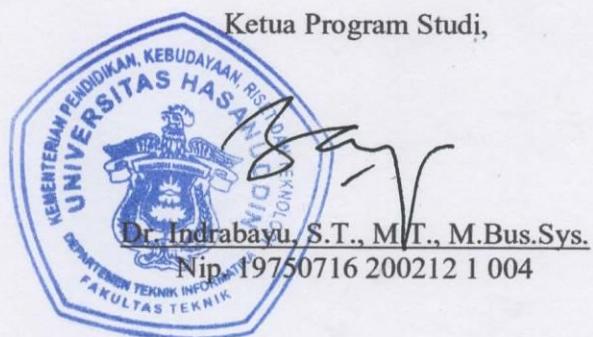
Pembimbing Utama,


Dr. Indrabayu, ST., MT., MBusSys.
Nip. 197507162002121004

Pembimbing Pendamping,


Dr. Ir. Ingrid Nurtanio, MT.
Nip. 196108131988112001

Ketua Program Studi,



PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Raedi Radifan

NIM : D42116304

Departemen : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

DETEKSI DAN HITUNG BUAH COKELAT MATANG DAN NON-MATANG PADA POHONNYA SECARA REALTIME

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 24 Oktober 2022

Yang menyatakan,



Muhammad Raedi Radifan

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan karuniaNya kepada penulis, sehingga penulis dapat menyelesaikan skripsi berjudul **“Deteksi Dan Hitung Buah Cokelat Matang Dan Non-Matang Pada Pohonnya Secara *Realtime*”** dengan baik. Shalawat dan salam senantiasa tercurah kepada Rasulullah SAW yang mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang ini. Penyusunan skripsi ini sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak baik morel maupun materil. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini terutama kepada:

1. Kedua orang tua, yang telah memberikan dukungan dan motivasi serta doa yang tiada henti-hentinya kepada penulis.
2. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., selaku dosen pembimbing I dan Ibu Dr. Ir. Ingrid Nurtanio, M.T., selaku dosen pembimbing II, yang selalu menyediakan waktu, tenaga dan pikirannya yang luar biasa untuk mengarahkan penulis dalam penyusunan Tugas Akhir ini.

3. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin, yang telah banyak membantu penulis selama masa perkuliahan.
4. AIMP *Family*, yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data, dan diskusi terkait penyusunan skripsi.
5. Saudara-saudari IGNITER 16, yang selalu menyemangati dan membantu penyelesaian skripsi ini serta mengisi hari-hari menjadi sangat menyenangkan.
6. Semua pihak atas dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga skripsi ini dapat bermanfaat bagi para pembaca dan semua pihak.

Makassar, Oktober 2022

Penulis

ABSTRAK

Estimasi jumlah dan waktu panen kakao (cokelat) sangat penting bagi petani untuk mengestimasi usaha yang dibutuhkan untuk memanen tanamannya. Hal ini dapat dilakukan dengan cara menghitung jumlah buah matang dan non matang pada pohon cokelat. Namun jika dilakukan manual akan sulit terutama jika terdapat banyak pohon dan lahan yang luas. Untuk itu pada penelitian ini diajukan solusi deteksi dan perhitungan jumlah buah cokelat langsung pada pohonnya menggunakan *deep learning*. Pengambilan data menggunakan *drone* dengan tiga variasi jarak yaitu 100 cm, 150 cm, dan 200 cm. Data yang digunakan dalam penelitian ini sebanyak 674 citra gambar dan dibagi menjadi 542 data *training* serta 132 data *testing*. Sedangkan algoritma yang digunakan adalah algoritma *You Only Look Once* (YOLO) versi ke-4 untuk pendekalian objek. Pada penelitian ini dilakukan perhitungan akurasi klasifikasi menggunakan *confusion matrix* yang menghasilkan rata-rata akurasi sebesar 97% serta perhitungan jumlah buah cokelat per pohon yang diperoleh dengan persentase jumlah buah prediksi yang benar terhadap jumlah buah sebenarnya yang menghasilkan rata-rata akurasi tertinggi 90% pada jarak 100 cm dan terendah 79% pada jarak 200 cm.

Kata Kunci: Deteksi Buah Cokelat, *Deep Learning*, CNN, *You Only Look Once* (YOLO), *Drone*, *Realtime*.

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN SKRIPSI.....	ii
PERNYATAAN KEASLIAN.....	iii
KATA PENGANTAR	iv
ABSTRAK.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4

BAB II	6
TINJAUAN PUSTAKA.....	6
2.1 Tanaman Kakao	6
2.2 Pengolahan Citra Digital	7
2.3 Visi Komputer.....	8
2.4 <i>Deep Learning</i>	9
2.5 <i>Convolutional Neural Network (CNN)</i>	11
2.6 YOLO (You Only Look Once).....	21
BAB III	25
METODOLOGI PENELITIAN	25
3.1 Tahap Penelitian.....	25
3.2 Waktu dan Lokasi Penelitian.....	26
3.3 Instrumen Penelitian.....	26
3.4 Teknik Pengambilan Data	27
3.5 Perancangan Sistem.....	29
3.6 Analisis Kerja Sistem	38
BAB IV	42
HASIL DAN PEMBAHASAN	42
4.1 Hasil Penelitian	42

4.2 Pembahasan.....	51
BAB V	56
PENUTUP	56
5.1 Kesimpulan	56
5.2 Saran	56
DAFTAR PUSTAKA	58
LAMPIRAN.....	60

DAFTAR GAMBAR

Gambar 2.1 Contoh buah kakao (a) matang (b) non matang	7
Gambar 2.2 Proses <i>Convolution Neural Network</i>	11
Gambar 2.3 Contoh RGB $4 \times 4 \times 3$	12
Gambar 2.4 Proses Konvolusi.....	13
Gambar 2.5 Fungsi Linear.....	14
Gambar 2.6 Fungsi Sigmoid	15
Gambar 2.7 Fungsi Tanh.....	16
Gambar 2.8 Fungsi ReLU	17
Gambar 2.9 Fungsi Leaky ReLU	18
Gambar 2.10 Fungsi Mish.....	19
Gambar 2.11 Contoh <i>max pooling</i>	20
Gambar 2.12 Contoh <i>max pooling</i> dan <i>mean pooling</i>	20
Gambar 2.13 <i>Fully Connected Layer</i>	21
Gambar 2.14 Perbandingan YOLOv4 dan objek detector lain	22
Gambar 2.15 Arsitektur sederhana YOLOv4.....	24
Gambar 3.1 Tahapan Penelitian	25

Gambar 3.2 Skema Pengambilan Data.....	28
Gambar 3.3 Contoh citra untuk tiga variasi jarak (a) 100 cm, (b) 150 cm, (c) 200 cm.....	29
Gambar 3.4 Rancangan Sistem	30
Gambar 3.5 <i>File predefined_classes.txt</i>	31
Gambar 3.6 (a) Contoh pelabelan citra & pembuatan <i>bounding box</i> (b) <i>Output file .txt</i>	32
Gambar 3.7 Contoh Proses Konvolusi	33
Gambar 3.8 Contoh proses <i>max pooling</i>	34
Gambar 3.9 Contoh <i>Fully Connected Layer</i>	35
Gambar 3.10 Contoh <i>Output Fully Connected Layer</i>	36
Gambar 3.11 Visualisasi <i>output</i> dari sistem.....	37
Gambar 3.12 Gambaran perhitungan IoU	38
Gambar 3.13 <i>Confussion Matrix</i>	39
Gambar 4.1 Contoh kesalahan klasifikasi <i>false positive</i> (a) dan <i>false negative</i> (b)	54
Gambar 4.2 Buah teroklusi batang.....	54
Gambar 4.3 Buah terkena cahaya matahari.....	55

DAFTAR TABEL

Tabel 4.1 <i>Confusion matrix</i>	43
Tabel 4.2 <i>Confusion matrix</i> setiap jarak	43
Tabel 4.3 Hasil Uji Deteksi dan Hitung Buah pada Jarak 100 cm	43
Tabel 4.3 (Lanjutan).....	44
Tabel 4.3 (Lanjutan).....	45
Tabel 4.3 (Lanjutan).....	46
Tabel 4.4 Hasil Uji Deteksi dan Hitung Buah pada Jarak 150 cm	46
Tabel 4.4 (Lanjutan).....	47
Tabel 4.4 (Lanjutan).....	48
Tabel 4.5 Hasil Uji Deteksi dan Hitung Buah pada Jarak 200 cm	49
Tabel 4.5 (Lanjutan).....	50
Tabel 4.5 (Lanjutan).....	51

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia saat ini telah memasuki era revolusi industri yang ke-empat atau disebut juga Industri 4.0, ditandai dengan penggunaan mesin-mesin otomasi yang terintegrasi dengan jaringan internet. Revolusi Industri 4.0 mendorong sistem otomatisasi di dalam semua proses aktivitas. Di sektor pertanian juga perlu beradaptasi untuk menjawab tantangan Industri 4.0. “Ke depan, olah lahan, tanam, panen, hingga pengolahan dilakukan menggunakan *remote control* dari rumah” kata Mantan Menteri Pertanian Andi Amran Sulaiman.

Amran menjelaskan, ada lima teknologi utama yang menopang implementasi Industri 4.0, yaitu: *Internet of Things*, *Artificial Intelligence*, *Human-Machine Interface*, teknologi robotik dan sensor, serta teknologi 3D Printing. Kesemuanya itu mentransformasi cara manusia berinteraksi hingga pada level yang paling mendasar, juga diarahkan untuk efisiensi dan daya saing industri.

“Melalui implementasi Industri 4.0 disektor pertanian, diharapkan proses usahatani menjadi semakin efisien, sehingga terjadi efisiensi, peningkatan produktivitas, dan daya saing,” jelas Amran menyambut peta jalan Making Indonesia 4.0 yang sebelumnya telah diluncurkan Presiden Joko Widodo (KEMENtan, 2018).

Salah satu komoditas pertanian yang unggul dan mendapat perhatian khusus adalah kakao yang sampai saat ini masih melakukan impor untuk memenuhi kebutuhan dalam negeri. Kepala Badan Penelitian dan Pengembangan Pertanian (Balitbangtan) Fadjry Djufry menyampaikan bahwa sektor kakao di Indonesia sedang mengalami pertumbuhan signifikan, yang didukung oleh perkembangan partisipasi petani kecil. Sejak 25 tahun terakhir, biji kakao merupakan komoditas ekspor yang penting bagi Indonesia. Para petani di Indonesia memiliki kontribusi paling besar dalam produksi kakao Nasional. Saat ini, luas perkebunan kakao di Indonesia mencapai sekitar 1,5 juta hektar. Produksi kakao di Indonesia terletak di Sulawesi, Sumatera Utara, Jawa Barat, Papua, Kalimantan Timur, dan Sumatera Barat. Dari wilayah-wilayah tersebut, 75% produksi kakao Indonesia terletak di Sulawesi (BALITBANGTAN, 2019). Adapun, impor biji kakao per 2019 mencapai sekitar 234.000 ton. Dengan kata lain, butuh setidaknya 292.000 hektar kebun kakao atau lebih dari 190 juta bibit kakao dengan produktivitas saat ini untuk menghilangkan seluruh impor biji kakao (Arief, 2020).

Seiring berkembangnya teknologi, begitu banyak pengembangan dan teknologi dan inovasi yang dilakukan untuk penelitian dalam bidang pertanian dan perkebunan cokelat salah satunya estimasi hasil panen (Ekawaty, 2019). Untuk mengestimasi jumlah panen buah kakao, perlu dilakukan perhitungan buah yang terdapat dalam pohon. Namun hal ini akan sulit dilakukan secara manual apalagi ketika jumlah pohon yang banyak dan ukuran lahan yang luas. Dan juga pada saat proses panen, tidak semua buah pada pohon kakao matang.

Oleh sebab itu perlu dibangun sebuah sistem yang mendeteksi dan menghitung buah cokelat matang dan non-matang menggunakan *drone*. Penggunaan *drone* dimaksudkan untuk efisiensi waktu dan tenaga kerja serta memperoleh data yang *realtime*. Sistem tersebut dibuat untuk mengestimasi hasil panen agar persiapan panen dan pasca panen dapat diprediksi lebih cepat dan akurat.

1.2 Rumusan Masalah

- a. Bagaimana sistem dapat mendeteksi dan menghitung buah cokelat matang dan non-matang pada pohonnya secara *realtime*?
- b. Bagaimana unjuk kerja sistem dalam mendeteksi dan menghitung buah cokelat matang dan non-matang pada pohonnya secara *realtime*?

1.3 Tujuan Penelitian

- a. Membuat sistem deteksi dan hitung buah cokelat matang dan non-matang pada pohonnya secara *realtime*.
- b. Mengetahui kinerja sistem deteksi dan hitung buah cokelat matang dan non-matang pada pohonnya secara *realtime*.

1.4 Batasan Masalah

Yang menjadi batasan masalah dalam penelitian ini adalah:

1. Data yang diambil merupakan dalam kondisi cerah.
2. Data yang diolah berupa data video yang diambil menggunakan *drone* dengan resolusi 1920×1080 piksel.
3. Pengambilan data dilakukan pada satu sisi pohon sejajar dengan buah.

1.5 Manfaat Penelitian

Menghemat tenaga kerja dan waktu serta mendapatkan estimasi hasil panen yang lebih akurat untuk digunakan sebagai data pendukung persiapan panen dan pasca panen.

1.6 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat dan sistematika penulis.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori terkait hal-hal yang mendasari dan yang berhubungan dengan visi komputer dan metode-metode yang digunakan pada penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini memaparkan tahapan penelitian, waktu dan lokasi penelitian, instrumen penelitian, teknik pengambilan data dan rancangan sistem serta penerapan algoritma dan metode-metode dalam pengolahan data, mulai dari preprocessing hingga menghasilkan prediksi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil penelitian dan pembahasan terkait pengolahan data yang telah dilakukan yang disertai dengan hasil penelitian.

BAB V PENUTUP

Bab ini berisi kesimpulan yang diperolah dari hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan sistem lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Tanaman Kakao

Kakao (*Theobroma cacao L.*) merupakan salah satu komoditas perkebunan yang sesuai untuk perkebunan rakyat, karena tanaman ini dapat berbunga dan berbuah sepanjang tahun, sehingga dapat menjadi sumber pendapatan harian atau mingguan bagi pekebun. Tanaman kakao berasal dari daerah hutan hujan tropis di Amerika Selatan. Di daerah asalnya, kakao merupakan tanaman kecil di bagian bawah hutan hujan tropis dan tumbuh terlindung pohon-pohon yang besar (Syahputra, 2021).

Tanaman Kakao secara umum mulai berproduksi pada umur 2,5 - 3 tahun setelah tanam. Perolehan produksi buah kakao ditahun pertama biasanya sedikit, namun seiring pertambahnya umur tanaman kakao maka produksi buah kakao yang dihasilkan juga akan terus meningkat. Buah kakao dihasilkan melalui proses penyerbukan antara bunga jantan dan bunga betina yang tumbuh menempel pada semua bagian batang tanaman sehingga menghasilkan buah kakao. Bunga-bunga yang tumbuh pada batang tanaman pokok ini umumnya akan menghasilkan buah yang besar dan berkualitas. Buah kakao akan siap dipanen setelah umur buah mencapai 4-5 bulan sejak terjadinya proses penyerbukan, atau dengan kata lain untuk mendapat buah kakao yang matang dan siap petik dibutuhkan kurang lebih 6 bulan /setengah tahun dari proses penyerbukan. Panen dilakukan pada saat buah matang, buah

muda berwarna merah berubah menjadi orange, dan buah muda berwarna hijau berubah menjadi warna kuning ketika matang.



(a) (b)

Gambar 2.1 Contoh buah kakao matang (a) non matang (b)

Apabila panen ditunda bisa mengakibatkan biji berkecambah saat masih didalam buah dan mengurangi kadar lemak pada biji kakao. Pemanenan buah kakao dilakukan sesering mungkin setiap 7-10 hari sekali agar dapat memutus siklus hidup hama dan mencegah penularan penyakit busuk pada buah kakao (Nas, 2021).

2.2 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (video). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi kontinyu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital,

maka suatu citra harus dipresentasikan secara numerik dengan nilai diskrit. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra. Tujuan dari pengolahan citra ini adalah memperbaiki kualitas suatu citra sehingga dapat diinterpretasi dengan mudah oleh manusia atau sebuah mesin (komputer). Proses perubahan citra menjadi citra digital dinamakan dengan digitasi. Digitasi merupakan proses sebuah gambar, teks, atau suara dari benda yang dapat dilihat ke dalam data elektronik dan dapat disimpan serta diproses untuk keperluan lainnya. tujuan dari pengolahan citra terbagi atas 3 bagian, yaitu:

- 1) Memodifikasi sebuah citra untuk meningkatkan kualitas maupun menekankan pada sejumlah aspek informasi yang terkandung didalam citra.
- 2) Mengklasifikasikan, mencocokan dan mengukur bagian-bagian tertentu dari sebuah citra.
- 3) Membagi bagian-bagian citra yang ingin dihilangkan atau digabungkan dengan bagian citra lain (Dompeipen, Sompei, & Najoan, 2021).

2.3 Visi Komputer

Visi komputer dapat didefinisikan dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan, klasifikasi, pengangan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasi citra. Inti dari teknologi visi komputer adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik

sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan.

Sebagai suatu disiplin ilmu, visi komputer berkaitan dengan teori di balik sistem buatan bahwa ekstrak informasi dari gambar. Data gambar dapat mengambil banyak bentuk, seperti urutan video, pandangan dari beberapa kamera, atau data multi-dimensi dari scanner medis. Sebagai disiplin teknologi, visi komputer berusaha untuk menerapkan teori dan model untuk pembangunan sistem visi komputer. Contoh aplikasi dari visi komputer mencakup sistem untuk:

- 1) Pengendalian proses (misalnya, sebuah robot industri atau kendaraan otonom).
- 2) Mendeteksi peristiwa (misalnya, untuk pengawasan visual atau orang menghitung).
- 3) Mengorganisir informasi (misalnya, untuk pengindeksan database foto dan gambar urutan).
- 4) Modeling benda atau lingkungan (misalnya, inspeksi industri, analisis citra medis).
- 5) Interaksi (misalnya, sebagai input keperangkat untuk interaksi komputer-manusia) (Prabowo, Abdullah, & Manik, 2018).

2.4 Deep Learning

Deep Learning adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi

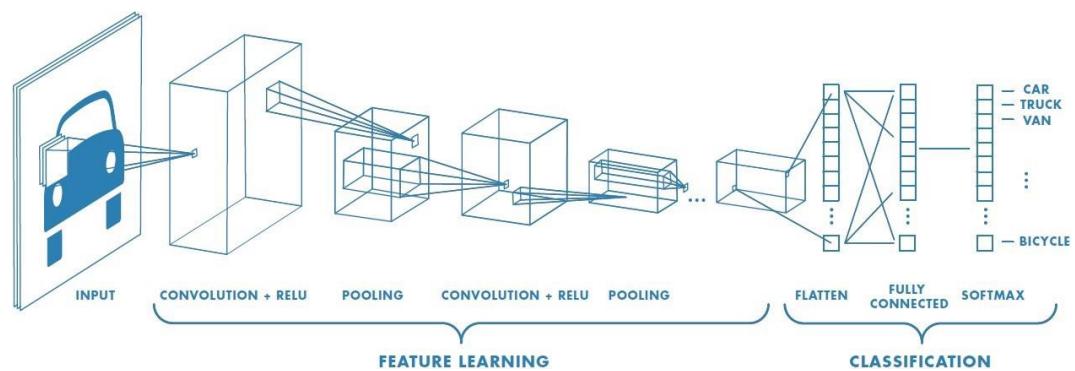
pada data menggunakan sekumpulan fungsi transformasi non-linier yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. *deep learning* disebut sebagai *deep* (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan. *deep learning* adalah salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai input dan mengolahnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi non-linier dari data masukan untuk menghitung nilai output. Algoritma pada *deep learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*).

Dalam jaringan saraf tiruan tipe *deep learning* setiap lapisan tersembunyi bertanggung jawab untuk melatih serangkaian fitur unik berdasarkan output dari jaringan sebelumnya. Algoritma ini akan menjadi

semakin kompleks dan bersifat abstrak ketika jumlah lapisan tersembunyi (*hidden layer*) semakin bertambah banyak. Jaringan saraf yang dimiliki oleh *deep learning* terbentuk dari hierarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi layer*). Berdasarkan hal itulah *deep learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi non-linier (Putri, 2020).

2.5 Convolutional Neural Network (CNN)

CNN adalah algoritma *deep learning* yang digunakan untuk memproses inputan data gambar, menentukan kepentingan (bobot dan bias yang dapat dipelajari) ke berbagai aspek dalam gambar dan berfungsi untuk membedakan objek satu dengan objek lainnya.

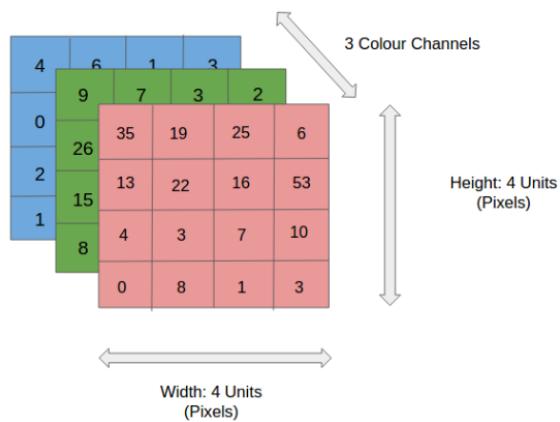


Gambar 2.2 Proses *Convolutional Neural Network*

Berikut adalah layer pada tahapan CNN yang terdapat pada **Gambar 2.2**. Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

1. Input Layer

Lapisan ini berguna untuk menampung pixel value dari gambar yang diinputkan. Semisal gambar yang diinputkan adalah gambar dengan ukuran piksel 4×4 dengan 3 channel warna RGB (*Red, Green, Blue*), seperti pada **Gambar 2.3** maka akan dimasukkan menjadi piksel array berukuran $4 \times 4 \times 3$.



Gambar 2.3 Contoh RGB $4 \times 4 \times 3$

2. Convolution Layer

Convolution Layer adalah *core* dari *Convolutional Neural Network*.

Lapisan ini menghasilkan gambar baru yang berisi *features* dari gambar yang telah diinputkan. Proses yang terjadi pada lapisan ini adalah konvolusi mengaplikasikan *filter* pada gambar. *Filter* yang diaplikasikan tersebut adalah berupa matriks bisa ukuran 1×1 , 3×3 , atau 5×5 . Proses konvolusi ini menghasilkan *feature map* yang kemudian akan dipergunakan pada saat lapisan aktivasi (*Activation Layer*). **Gambar 2.4** menggambarkan proses konvolusi bekerja.

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

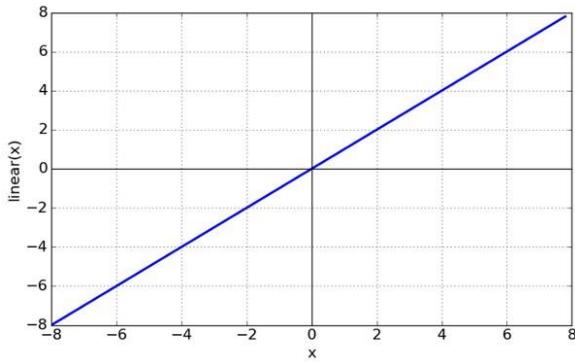
Gambar 2.4 Proses Konvolusi

3. Activation Layer

Activation Layer terletak sebelum melakukan *Pooling Layer* dan setelah melakukan *Convolution Layer*. Pada lapisan ini, terjadi proses pengubahan nilai nilai *feature map* pada jarak tertentu tergantung pada *Activation Function* yang dipakai. Tujuan utamanya yaitu untuk meneruskan nilai yang memperlihatkan *feature* dari gambar inputan. *Activation function* pada dasarnya dapat dibagi menjadi 2 tipe yaitu fungsi aktivasi linear dan fungsi aktivasi non-linear:

- Fungsi Aktivasi Linear

Fungsi aktivasi linear merupakan fungsi yang memiliki nilai *output* yang sama dengan nilai *inputnya*. Hal ini berkaitan dengan, jika sebuah neuron menggunakan *linear activation*, maka keluaran dari neuron tersebut adalah *weighted sum* dari *input* + bias. Grafik fungsi linear ditunjukan oleh **Gambar 2.5**.



Gambar 2.5 Fungsi Linear

Adapun rumus matematis dari fungsi aktivasi linear dapat dilihat pada **Persamaan 2.1**.

$$f(x) = x \quad (2.1)$$

- Fungsi Aktivasi Non-linier

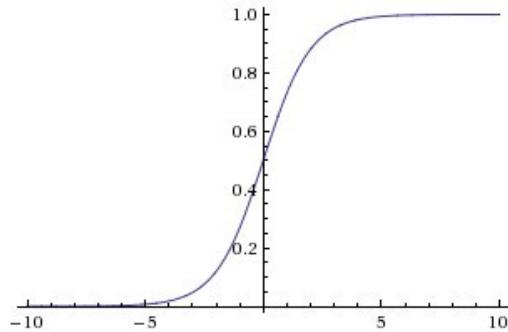
Fungsi aktivasi nonlinear adalah fungsi aktivasi yang paling banyak digunakan. Fungsi aktivasi non-linier memudahkan model untuk mengeneralisasi atau menyesuaikan dengan berbagai data dan untuk membedakan antara output. Terdapat beberapa fungsi aktivasi non-linier, diantaranya adalah:

- Sigmoid

Fungsi aktivasi sigmoid merupakan fungsi non-linear.

Masukan untuk fungsi aktivasi ini berupa bilangan riil dan output dari fungsi tersebut memiliki nilai antara 0 sampai 1.

Gambar 2.6 menunjukkan grafik fungsi aktivasi sigmoid:



Gambar 2.6 Fungsi Sigmoid

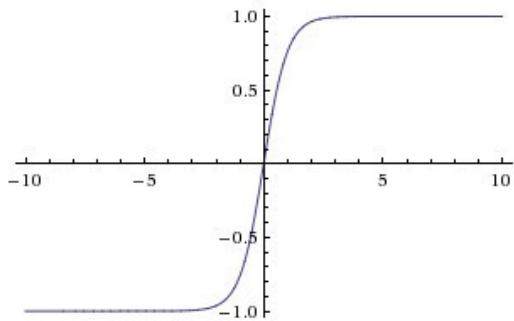
Sigmoid memiliki persamaan sebagai berikut:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2.2)$$

Jika input dari suatu node pada neural network bernilai negatif maka keluaran yang didapatkan adalah 0, sedangkan jika masukannya bernilai positif maka keluaran nilainya adalah 1. Kelemahan dari fungsi ini adalah kemungkinan besar sigmoid mematikan gradient (gradient bernilai 0). Hal ini dapat terjadi ketika neuron mengeluarkan nilai antara 0 dan 1. Selain itu, output dari sigmoid tidak berpusat pada 0 (*zero-centered*).

- Tanh

Tanh merupakan Sigmoid yang dikembangkan. Tanh akan mengubah nilai input x nya menjadi sebuah nilai yang memiliki range mulai dari -1 sampai 1. **Gambar 2.7** menunjukkan grafik fungsi Tanh:



Gambar 2.7 Fungsi Tanh

Sama seperti Sigmoid, Tanh memiliki kekurangan yaitu bisa mematikan gradient, tetapi kelebihannya adalah output yang dimiliki Tanh merupakan zero-centered. Dalam praktiknya Tanh lebih menjadi pilihan jika dibandingkan dengan Sigmoid.

Tanh memiliki persamaan sebagai berikut:

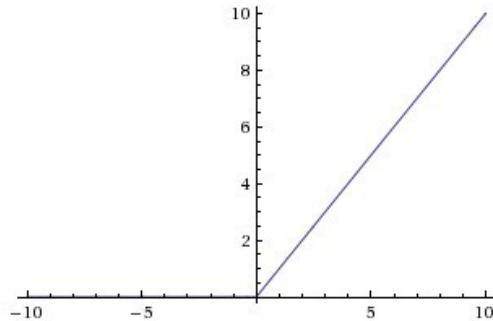
$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.3)$$

- ReLU (*Rectified Linear Unit*)

Pada dasarnya fungsi ReLU melakukan threshold dari 0 hingga infinity. ReLU memiliki persamaan sebagai berikut:

$$f(x) = \max(0, x) \quad (2.4)$$

Fungsi ini menjadi salah satu fungsi yang populer saat ini. **Gambar 2.8** menunjukkan grafik fungsi aktivasi ReLU:

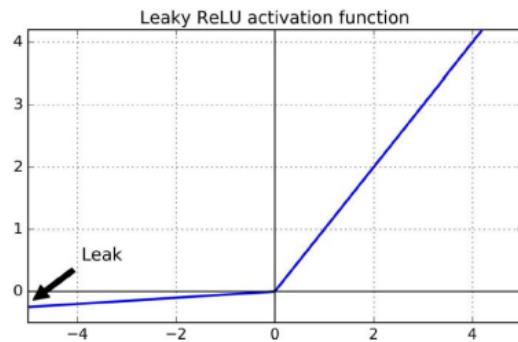


Gambar 2.8 Fungsi ReLU

Seperti yang dapat dilihat, ReLU pada intinya hanya membuat batasan pada bilangan nol, artinya apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Permasalahan yang mungkin timbul jika menggunakan ReLU adalah semua nilai negatif menjadi nol sehingga menurunkan kemampuan model dalam proses pelatihan data, karena setiap masukan data negatif yang diberikan pada fungsi aktivasi ReLU nilainya akan menjadi nol dalam grafik, sehingga mengakibatkan yang dihasilkan tidak dapat memetakan nilai negatif secara tepat.

- Leaky ReLU

Fungsi Aktivasi Leaky ReLU merupakan pengembangan dari fungsi aktivasi ReLU. Leaky ReLU menambahkan sebuah Leak dan memberlebar nilai output dari neuron. Adapun grafik dari fungsi aktivasi Leaky ReLU dapat dilihat pada **Gambar 2.9.**



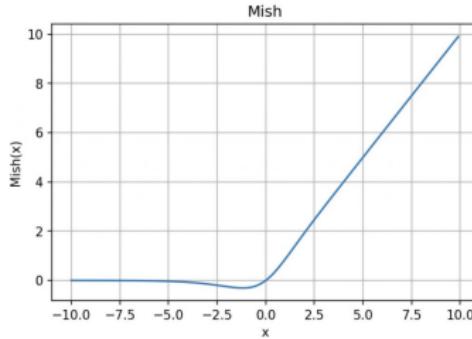
Gambar 2.9 Fungsi Leaky ReLU

Bisa dilihat pada **Gambar 2.9** Leaky ReLU membuat limit jika nilai inputan dibawah 0 dimana nilai input dikalikan dengan 0,01 dan jika nilai input ≤ 0 maka nilai input sama dengan nilai output. Adapun persamaan dari Leaky ReLU dapat dilihat pada **Persamaan 2.5** (Hatta K, 2017).

$$f(x) = (0.1x, x) \quad (2.5)$$

- Mish

Fungsi Aktivasi Mish adalah salah satu fungsi aktivasi yang dikembangkan oleh Diganta Misra dalam penelitiannya yang berjudul “*Mish: A Self Regularized Non-Monotonic Activation Function*”, *Mish function* memiliki performa yang lebih bagus dibandingkan dengan ReLU tetapi memerlukan komputasi yang lebih banyak. Adapun grafik dari fungsi aktivasi mish dapat dilihat pada **Gambar 2.10**.



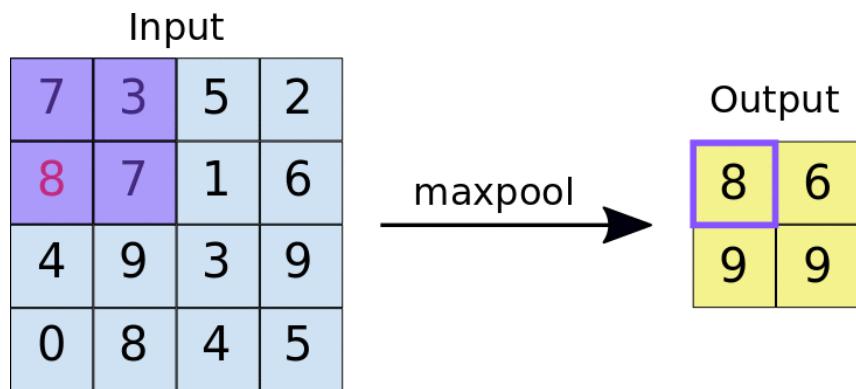
Gambar 2.10 Fungsi Mish

Pada **Gambar 2.10** bisa dilihat grafik input dan output dari fungsi aktivasi mish adapun rumus matematis dari Fungsi aktivasi Mish dapat dilihat pada **Persamaan 2.6** (Misra, 2020).

$$f(x) = x \tanh(\ln(1 + e^x)) \quad (2.6)$$

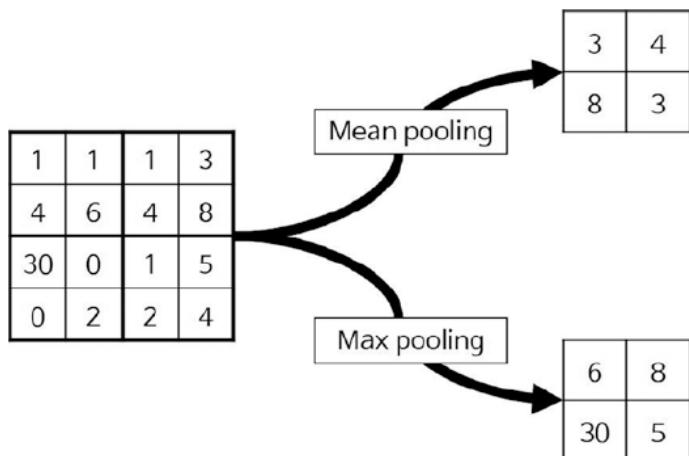
4. Pooling Layer

Pada *Neural Network*, pada umumnya setelah proses konvolusi selesai, maka akan dilakukan proses *pooling*. *Pooling* adalah proses mengurangi ukuran/resolusi pada gambar akan tetapi tetap mempertahankan informasi penting pada gambar tersebut. *Max pooling* merupakan salah satu contoh *pooling*. *Max pooling* ini bekerja dengan cara mengambil satu piksel pada area dengan luas tertentu pada gambar. Hasil *max pooling* adalah menciptakan gambar baru. **Gambar 2.11** adalah contoh *max pooling* dengan ukuran 2×2 pixel dan gambar yang dilakukan *max pooling* berukuran 4×4 pixel. Hasilnya akan membuat sebuah gambar baru dengan ukuran 2×2 .



Gambar 2.11 Contoh *max pooling*

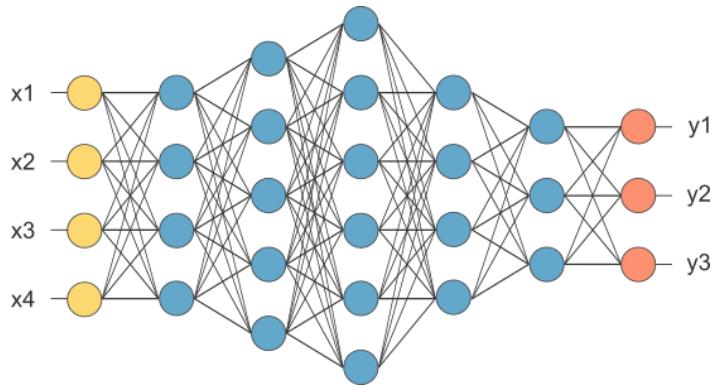
Terdapat beberapa macam pooling lain seperti *max pooling*, *min pooling*, *mean pooling*, *sum pooling*, dll **Gambar 2.12** menunjukan contoh *max pooling* dan *mean pooling* pada gambar yang sama.



Gambar 2.12 Contoh *max pooling* dan *mean pooling* pada gambar yang sama

5. Fully Connected Layer

Fully Connected Layer adalah lapisan dimana semua *neuron* aktivitas dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti hal nya jaringan syaraf tiruan bisa. Setiap aktivitas dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua neuron di lapisan *Fully-Connected*.



Gambar 2.13 Fully Connected Layer

Dari **Gambar 2.13** bisa dilihat bahwa sebelumnya hasil dari pooling dipakai sebagai masukan, hasil pooling dikonversikan dahulu menjadi vektor x_1 , x_2 , x_3 , dan seterusnya. Selanjutnya, vektor tersebut dimasukan kedalam *Fully Connected Layer*. Pada lapisan terakhir di dalam *fully connected layer* akan dipergunakan *activation function sigmoid* atau *softmax* yang berguna untuk menentukan klasifikasi gambar inputan (Irawan, 2020).

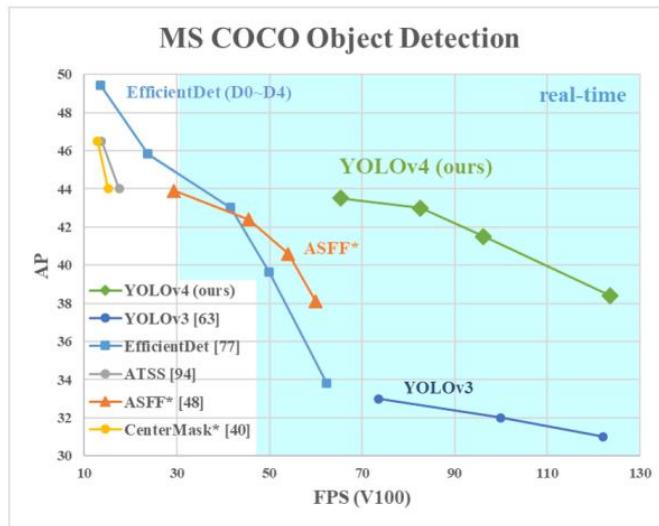
2.6 YOLO (You Only Look Once)

YOLO atau *You Only Look Once* merupakan sebuah algoritma yang menggunakan *convolutional neural networks* untuk pendekripsi objek. YOLO merupakan salah satu algoritma pendekripsi objek tercepat bila dibandingkan dengan algoritma lain (Kelvin, 2020).

YOLO merupakan salah satu metode yang memprediksi *bounding box* dan probabilitas pada kelas secara langsung dalam sekali evaluasi. Sistem deteksi pada YOLO sangat mudah yaitu apabila telah mendapatkan input citra, sistem akan melakukan perubahan terhadap citra menjadi 416 x 416.

Setelah itu diproses dengan *single convolutional neural network* dan dilakukan *non-max suppression* agar menghasilkan *bounding box* yang menentukan kelas dari tiap objeknya. *Non-max suppression* memiliki peran penting dalam memilih *bounding box* dengan nilai *confidence* yang lebih tinggi karena dalam algoritma deteksi objek pasti ada kemungkinan terdapat lebih dari satu *bounding box* (Kusuma, Usman, & Saidah, 2021).

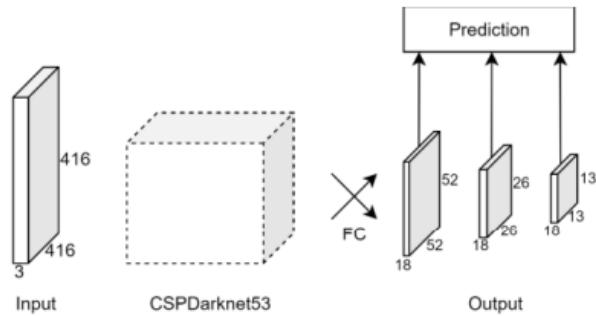
YoloV4 adalah pengembangan dari versi sebelumnya, yaitu YoloV3. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao mengusulkan algoritma YoloV4 dengan perubahan signifikan dari versi sebelumnya dan akurasi yang jauh lebih baik yaitu dengan peningkatan AP (*Average Precision*) dan FPS (*Frame Per Second*) pada YoloV4 dengan masing-masing 10% dan 12% (Bochkovskiy, Wang, & Liao, 2020). Perbandingannya dapat dilihat pada **Gambar 2.14**.



Gambar 2.14 Perbandingan YOLOv4 dan objek detektor lain

YOLOv4 menambahkan sejumlah implementasi tambahan yang meningkatkan akurasi dan efisiensi pendekripsi seperti *Weighted Residual Connection* untuk mengkombinasikan residu dari lapisan *neural network* secara efektif dan efisien. *Cross Stage Partial Connections* untuk memisahkan *feature map* dari lapisan dasar dengan menyalinnya dan mengirim satu salinan melalui *dense block* dan mengirimkan yang lain langsung ke tahap berikutnya, sehingga mengurangi komputasi di *DenseNet* dan meningkatkan pembelajaran dengan meneruskan versi *feature map* yang sebelumnya. *Cross Mini-Batch Normalization* untuk memungkinkan *training* pada satu GPU, karena sebagian besar teknik normalisasi batch memanfaatkan beberapa kekuatan GPU. *Self Adversarial Training*, mengaburkan wilayah gambar yang paling bergantung pada jaringan untuk memaksa jaringan mempelajari fitur-fitur baru. *Mosaic Data Augmentation*, menggabungkan empat gambar *training* bersama-sama, yang membantu model dalam belajar menemukan objek yang lebih kecil dan kurang fokus pada lingkungan yang tidak langsung berada di sebelah objek. *Mish Activation*, Fungsi aktivasi ini digunakan untuk mendorong sinyal ke kiri & kanan, yang tidak mungkin dilakukan dengan fungsi seperti *ReLU*. *Mish* memberikan hasil empiris yang lebih baik jika dibandingkan dengan *ReLU*, *Swish* & *Leaky ReLU*. Juga berbagai teknik Augmentasi Data berperilaku konsisten dengan penggunaan *Mish*. *DropBlock regularization* adalah teknik regularisasi untuk mengatasi over-fitting. Ini menjatuhkan blok piksel. Ia bekerja pada lapisan *Convolution*, tidak seperti putusnya pixel yang tidak

bekerja pada lapisan ini. *CIoU loss*, sebuah fungsi loss yang mencapai konvergensi lebih cepat dan akurasi yang lebih baik pada permasalahan regresi *bounding box* (Solawetz, 2020).



Gambar 2.15 Arsitektur sederhana YOLOv4

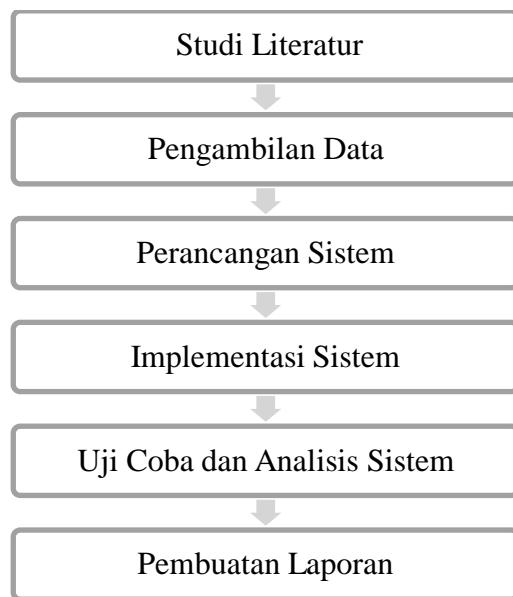
Pada **Gambar 2.15**, dapat dilihat arsitektur yang sangat sederhana dari YOLOv4. Arsitektur ini menggunakan arsitektur backbone CNN CSPDarknet53. Arsitektur ini berisi 162 lapisan. Masukan atau input merupakan citra yang sudah di *resize* sesuai ukuran yang diatur pada konfigurasi. Selanjutnya, lapisan input dimasukan kedalam arsitektur backbone CSPDarknet53 (Dhiaegana & Munir, 2020).

BAB III

METODOLOGI PENELITIAN

3.1 Tahap Penelitian

Tahap pada penelitian ini sebagaimana diperlihatkan pada **Gambar 3.1**



Gambar 3.1 Tahapan Penelitian

Adapun uraian tahap penelitian pada **Gambar 3.1** sebagai berikut.

1. Studi Literatur merupakan tahap awal yang dilakukan untuk mencari referensi penelitian-penelitian terkait topik yang dilakukan yaitu metode-metode *deep learning* untuk mendeteksi objek.
2. Pengambilan data video dilakukan selama penelitian, dengan menggunakan *drone*. Data video yang diambil akan digunakan sebagai data latih dan data uji pada sistem yang akan dibuat.

3. Perancangan sistem dilakukan dengan membuat alur kerja sistem serta diagram sistem penelitian mulai dari proses *input* data, *preprocessing*, ekstraksi fitur serta deteksi dan hitung buah kakao pada pohonnya.
4. Implementasi sistem dilakukan sesuai dengan alur kerja yang telah dibuat pada tahap sebelumnya.
5. Uji coba sistem dilakukan untuk memastikan sistem berjalan sebagaimana mestinya serta mengetahui seberapa akurat sistem yang dibuat.
6. Tahap akhir dari penelitian ini melakukan penulisan laporan dalam bentuk skripsi sebagai bahan publikasi.

3.2 Waktu dan Lokasi Penelitian

Waktu penelitian dilakukan dimulai sejak disetujuinya proposal penelitian ini pada bulan September 2020 hingga proses pelaporan pada bulan Maret 2022. Penelitian ini dilakukan di Laboratorium Kecerdasan Buatan Departemen Teknik Informatika, Fakultas Teknik Universitas Hasanuddin. Lokasi pengambilan data dilakukan di Kabupaten Polewali Mandar dan Soppeng.

3.3 Instrumen Penelitian

Pada penelitian ini instrumen yang dilakukan meliputi dua komponen, yaitu perangkat keras dan perangkat lunak, sebagai berikut.

1. Perangkat Keras (*Hardware*)

- a. Laptop Vaio VJP132, Processor Intel® Core™ i7-5500U CPU @ 2.40GHz, RAM 8 GB.
- b. *Drone DJI Mavic Air.*
- c. *Smartphone*

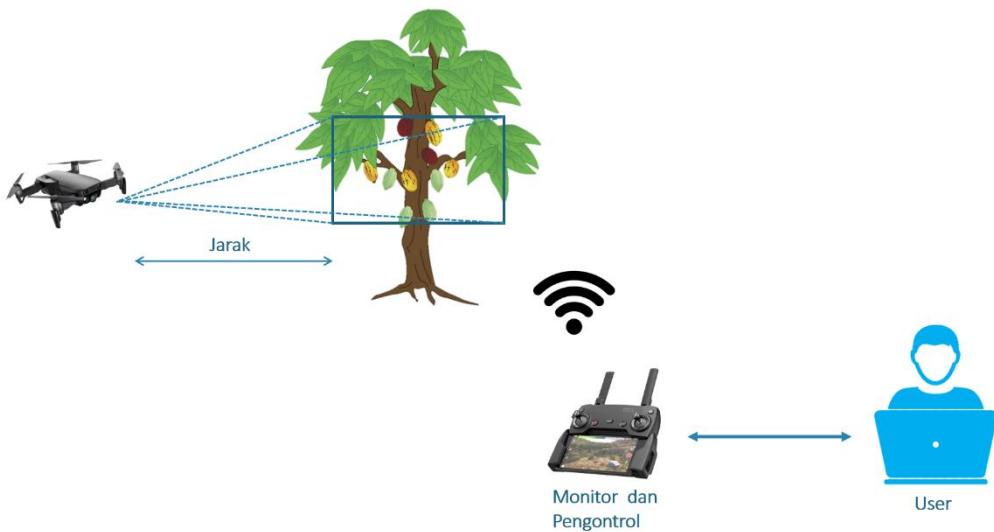
2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membangun aplikasi ini memiliki kemampuan untuk memproses citra digital, yaitu:

- a. Sistem operasi *Windows 10 x64*
- b. Aplikasi *Anaconda*
- c. *Visual Studio Code*
- d. *LabelImg*

3.4 Teknik Pengambilan Data

Pengambilan data dilakukan dengan menghubungkan *smartphone* dengan *remote control drone*, kemudian *drone* diterbangkan diantara baris pohon cokelat dengan tiga variasi jarak yaitu, 100 cm, 150 cm, dan 200 cm. pengambilan data dilakukan dengan posisi kamera drone mengarah ke buah serta ketinggian menyesuaikan terhadap buah yang berada pada pohon. Skema pengambilan data dapat diilustrasikan pada **Gambar 3.2**. Contoh hasil dari tiap jarak dapat dilihat pada **Gambar 3.3**

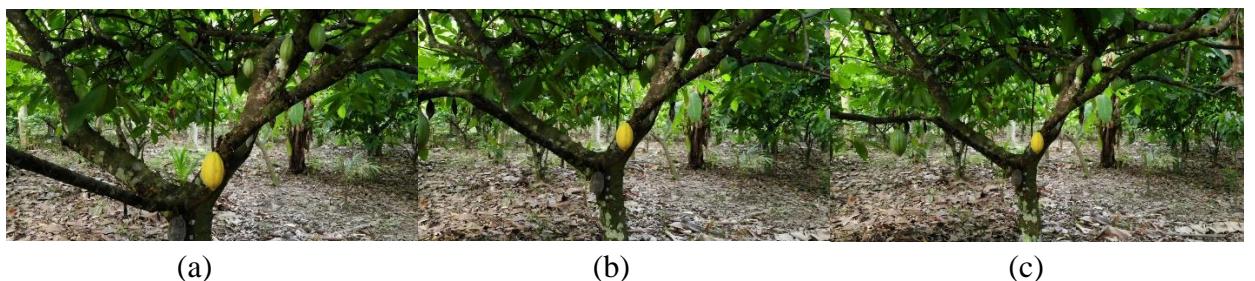


Gambar 3.2 Skema Pengambilan Data

Data yang diperoleh kemudian diproses ke PC melalui *streaming* untuk dapat diolah datanya hingga menghasilkan sistem pendekripsi buah cokelat secara *realtime*. Penerbangan *drone* masih terbatas pada jangkauan *wifi smartphone* yang digunakan. Untuk melakukan *streaming* dari *drone* ke PC langkah-langkahnya sebagai berikut:

1. Melakukan instalasi MonaServer pada PC atau laptop untuk menjalankan protokol RTMP (*Real Time Messaging Protocol*).
2. Menghubungkan android yang sekaligus berfungsi sebagai *remote control* dengan laptop tempat data akan diolah, dengan melakukan pengaturan pada *IP address* masing-masing perangkat dan melakukan koneksi dalam satu jaringan sama.
3. *Streaming* pada drone DJI diatur ke menu *custom*, yaitu menu *streaming* melalui protokol rtmp://10.1.1.1:1935 yang alamatnya disesuaikan dengan IP pada laptop. Selanjutnya *drone* diarahkan pada objek untuk mengambil gambar, maka proses *streaming* dimulai.

Hasil rekaman berupa video yang berekstensi .mp4, kemudian diekstrak menjadi *frame* dan akan diambil beberapa *frame*. Ekstraksi dilakukan dengan mengambil satu *frame* tiap tiga detik, *frame* yang terpilih merupakan *frame* yang pertama pada detik tersebut. Data yang dihasilkan kemudian dibagi menjadi data latih dan data uji.



Gambar 3.3 Contoh citra untuk tiga variasi jarak (a) 100 cm,
(b) 150 cm, (c) 200 cm

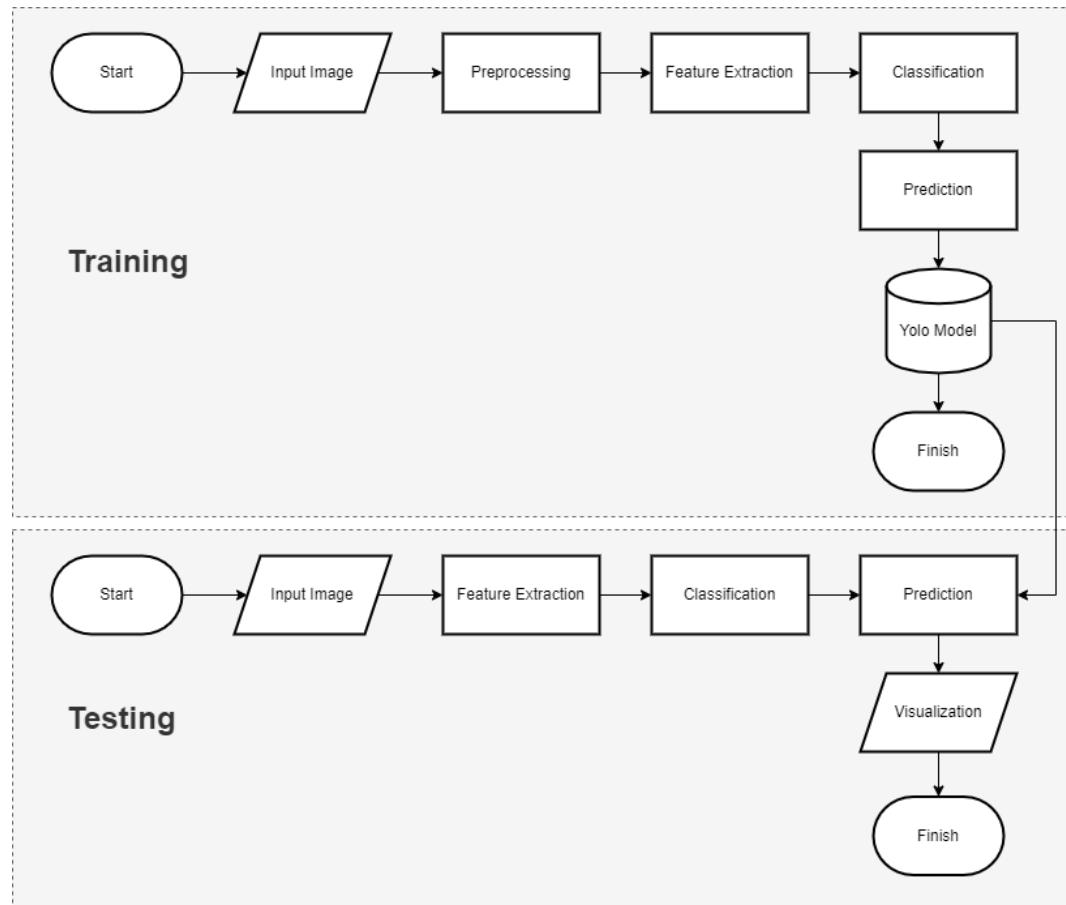
3.5 Perancangan Sistem

Sistem yang dibuat pada penelitian ini menggunakan algoritma *deep learning* YOLO versi keempat yang menggunakan *framework CSPDarknet53* yaitu prinsip kerjanya berdasarkan CNN dan *Cross Stage Partial (CSP)*.

Secara umum sistem ini terbagi menjadi dua bagian, yaitu proses *training* dan *testing*. Data latih yang digunakan sebanyak 542 gambar dan data uji sebanyak 132 gambar. Dalam proses *training*, fitur yang telah diekstrak dari data latih menggunakan *layer CNN* akan dijadikan *input* ke dalam *fully connected layer* yang akan menghasilkan klasifikasi objek buah matang dan non matang. Hasil klasifikasi dari *fully connected layer* inilah yang akan digunakan untuk menentukan hasil deteksi objek. Tahap *training*

menghasilkan model yang akan digunakan untuk proses *testing* sistem.

Rancangan sistem pada penelitian ini dapat dilihat sebagai berikut:



Gambar 3.4 Rancangan Sistem

3.5.1 *Input Image*

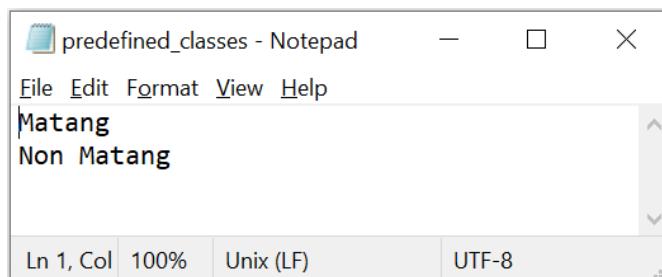
Tahap awal yang harus dilakukan adalah mempersiapkan data masukan untuk proses pelatihan dan pengujian. Data masukan berupa beberapa video berukuran 1920×1080 yang diambil menggunakan kamera *drone*. Kemudian dilakukan proses ekstraksi *frame* pada *video* agar mendapatkan citra.

3.5.2 *Preprocessing*

Preprocessing yang dilakukan pada tahap ini adalah *labelling*.

Labelling merupakan proses pemberian informasi objek yang terdapat pada citra dengan cara menandai objek-objek dan mengklasifikasikan kelas objek tersebut. Citra-citra hasil ekstraksi akan diberi label menggunakan LabelImg. Langkah-langkah memberi label sebagai berikut:

1. Menentukan daftar kelas yang akan digunakan untuk *training* dengan mengubah *predefined_classes.txt* pada folder *data*. kelas yang digunakan ada dua yaitu Matang dan Non Matang.



Gambar 3.5 File *predefined_classes.txt*

2. Menjalankan *labelimg.exe* dan klik tombol "PascalVOC" yang berada Tepat di bawah tombol "Save" di toolbar untuk beralih ke format YOLO.
3. Membuka folder yang berisi citra hasil ekstraksi untuk memproses banyak citra dan mengganti folder penyimpanan ke folder yang berisi citra itu sendiri.
4. Melakukan pelabelan manual pada objek. Pada proses ini tiap objek buah kakao pada gambar akan diberi kotak pembatas

(*bounding box*), kemudian memilih nama kelas yaitu “Matang” atau “Non Matang” dari objek tersebut.

5. Setiap gambar yang telah diberi label, secara otomatis menghasilkan sebuah file anotasi dalam format text yang berisi koordinat dan *id class* dengan format $<Class><X><Y><Width>$ $<Height>$.

File anotasi adalah hasil dari *labelling* yang merupakan persyaratan dasar untuk melatih sebuah model, anotasi ini juga untuk membantu mengenali data yang relevan. Contoh pelabelan citra dan pembuatan *bounding box* objek serta output file .txt dapat dilihat pada

Gambar 3.6.



(a)

(b)

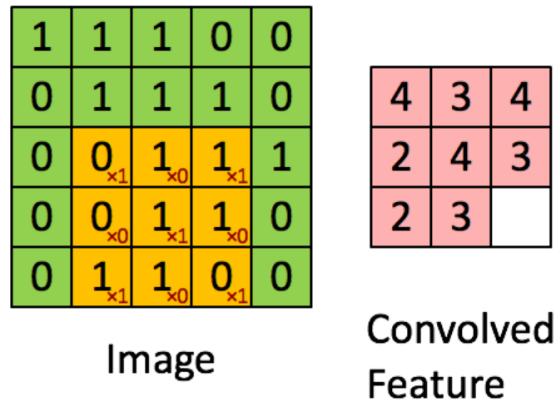
Gambar 3.6 (a) Contoh pelabelan citra dan pembuatan *bounding box*, (b) Output file .txt

3.5.3 Feature Extraction

Ekstraksi fitur bertujuan untuk mendapatkan informasi-informasi penting dari sebuah citra digital. Pada sistem ini, proses ekstraksi fitur menggunakan CNN, dimana didalamnya terdapat proses yang akan terulang beberapa kali.

- **Convolution Layer**

Convolution Layer merupakan layer pertama dalam proses ekstraksi fitur dari citra masukan. Konvolusi menjaga hubungan antara piksel dengan mempelajari fitur gambar menggunakan kotak kecil dari data masukan. Proses konvolusi merupakan proses operasi matematik dari dua masukan, yaitu matriks gambar dan matriks *filter* atau *kernel*. *Filter* kemudian bergeser sesuai dengan nilai *stride* yang ditentukan. Contoh proses konvolusi dapat dilihat pada **Gambar 3.7**.

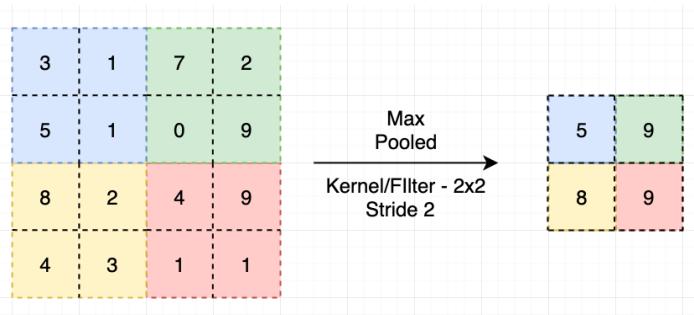


Gambar 3.7 Contoh Proses Konvolusi

- ***Pooling Layer***

Pooling Layer berfungsi untuk mengurangi jumlah parameter yang digunakan ketika dimensi citra masukan terlalu besar sehingga proses komputasi menjadi lebih cepat. Tipe *pooling* yang dipakai YOLOv4 adalah *Max Pooling*. *Max Pooling* bekerja dengan cara memilih nilai terbesar dari sebuah area citra dimana besaran area ditentukan oleh besar dimensi *filter* dan *filter* bergeser sesuai dengan nilai *stride* yang ditentukan. Contoh proses *pooling* dapat dilihat pada

Gambar 3.8.



Gambar 3.8 Contoh proses *max pooling*

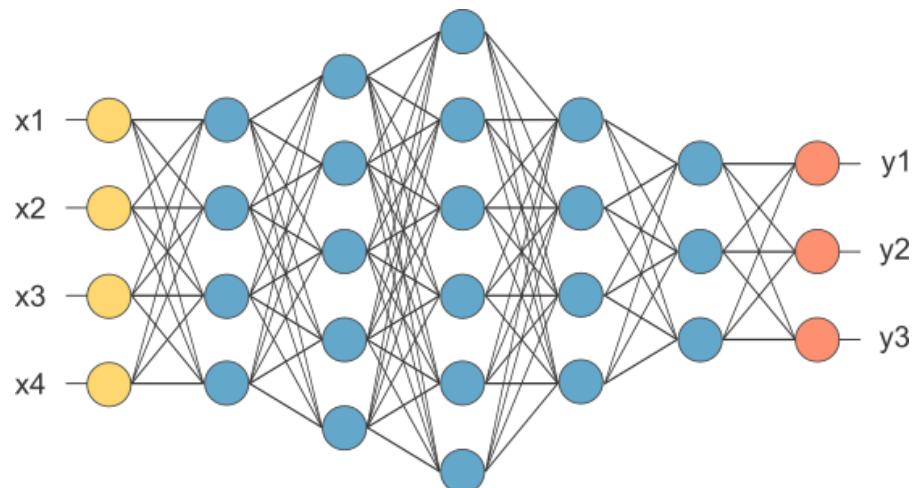
Kedua proses tersebut akan dilakukan berulang hingga membentuk *activation map* yaitu gambar yang sudah terbentuk setelah

melewati *filter* yang mengekstrak fitur yang berbeda dari gambar tersebut.

3.5.4 Classification

Data hasil proses ekstraksi fitur berupa *activation map* akan diteruskan ke *Fully Connected Layer*. Pada proses klasifikasi, model mengenali apakah ada objek yang terdeteksi pada suatu gambar dan apakah objek tersebut merupakan buah cokelat matang atau non matang. *Activation map* akan melewati proses *reshape* menjadi vektor terlebih dahulu, dikarenakan *FC Layer* membutuhkan masukan berupa vektor sedangkan *activation map* merupakan *array* multidimensi. Matriks hasil reshape tersebut kemudian menjadi masukan untuk *FC Layer*. Ilustrasi beserta hasil dari *FC Layer* dapat dilihat pada

Gambar 3.9.

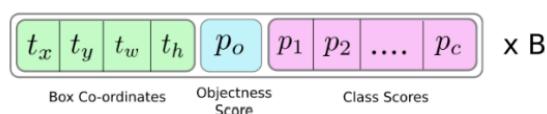


Gambar 3.9 Contoh *Fully Connected Layer*

3.5.5 Prediction

YOLOv4 menggunakan 2 pendekatan untuk menggambar *bounding box*, yaitu *Confidence Level* dan *Class Probability Map*. Pada pendekatan *confidence level* sistem akan membagi gambar menjadi beberapa *grid cell*, kemudian *grid cell* tersebut menjadi acuan untuk mendeteksi objek. Setiap *grid cell* akan memprediksi dan menggambarkan *bounding box* yang memiliki *confidence score* untuk setiap kotaknya. *Confidence score* merepresentasikan seberapa tinggi keyakinan sistem bahwa kotak tersebut mengandung objek yang ingin dideteksi dan seberapa akurat sistem memprediksi objeknya. Pada pendekatan *class probability map*, setiap *bounding box* terdiri dari 5 nilai: x, y, w, h, dan *confidence*. Nilai x dan y merepresentasikan kordinat titik tengah dari *bounding box* terhadap *grid cell*. Nilai w dan h merepresentasikan lebar dan tinggi dari keseluruhan gambar. Sedangkan nilai *confidence* merepresentasikan *Intersection over Union* (IoU) antara *bounding box* objek hasil prediksi dengan *bounding box* objek yang asli.

Keluaran dari FC Layer berupa matriks berisikan *grid cell*, dengan parameter untuk setiap objek yang terdeteksi dalam sebuah *grid cell* seperti **Gambar 3.10**.



Gambar 3.10 Contoh *Output Fully Connected Layer*

Parameter keluaran terdiri dari 3 bagian, yaitu koordinat box, *objectness score* atau nilai *confidence* atau persentase kemungkinan yang dideteksi merupakan objek, dan *class scores* atau persentase kemungkinan objek merupakan anggota dari kelas untuk masing-masing kelas.

3.5.6 Visualization

Visualisasi hasil deteksi sistem dapat ditampilkan dalam bentuk citra atau video sesuai dengan keperluan. Proses visualisasi menggunakan OpenCV yang mengefisiensikan proses komputasi data sehingga penggambaran *bounding box* menjadi lebih cepat.

Bounding box yang ditampilkan hanyalah *bounding box* yang memiliki *id* yang sama dengan *id* kelas yang ingin dihitung. Jumlah *bounding box* yang memiliki kelas yang sama dengan objek di gambar kemudian dihitung dan ditampilkan pada bagian bawah kiri citra atau video. Contoh visualisasi dari *output* dapat dilihat pada **Gambar 3.11**.

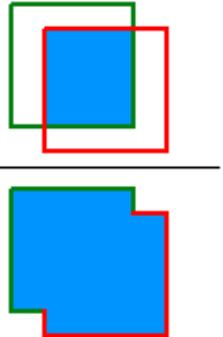


Gambar 3.11 Visualisasi *output* dari sistem

3.6 Analisis Kerja Sistem

3.6.1 *Intersection Over Union (IoU)*

IoU secara sederhana merupakan salah satu ukuran evaluasi untuk algoritma pendekripsi objek. Seluruh algoritma yang menampilkan bounding box dapat diuji menggunakan IoU. Prediksi dengan nilai IoU diatas 0.5 dianggap sebagai prediksi yang baik. **Gambar 3.12** menampilkan gambaran perhitungan IoU.

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Area of intersection}}{\text{Area of union}}$$


Gambar 3.12 Gambaran perhitungan IoU

3.6.2 *Confussion Matrix*

Confussion matrix digunakan sebagai basis untuk mengukur performa pengujian model terhadap dataset. *Confussion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Aturan perhitungan dapat dilihat pada **Gambar 3.13.**

		ACTUAL	
		Class 1 (Positive)	Class 2 (Negative)
PREDICTION	Class 1 (Positive)	TP	FP
	Class 2 (Negative)	FN	TN

Gambar 3.13 Confussion Matrix

Pada penelitian ini, *confussion matrix* dapat dijelaskan sebagai berikut:

- **True Positive (TP)**, yaitu jumlah buah matang yang terprediksi buah matang oleh sistem.
- **False Positive (FP)**, yaitu jumlah buah non matang yang terprediksi buah matang oleh sistem.
- **False Negative (FN)**, yaitu jumlah buah matang yang terprediksi buah non matang oleh sistem.
- **True Negative (TN)**, yaitu jumlah buah non matang yang terprediksi buah non matang oleh sistem.

Berdasarkan *confussion matrix*, kita dapat menghitung nilai *precision*, *recall*, *accuracy*, dan *f1 score* serta perhitungan buah pada pohon.

- **Precision (P)**, yaitu rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Untuk menghitung *precision* sistem digunakan persamaan berikut:

$$P = \left(\frac{TP}{TP+FP} \right) \times 100\% \quad (3.1)$$

- **Recall (R)**, yaitu rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Untuk menghitung nilai *recall* sistem digunakan persamaan berikut:

$$R = \left(\frac{TP}{TP+FN} \right) \times 100\% \quad (3.2)$$

- **Accuracy (A)**, yaitu rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Untuk menghitung nilai *accuracy* sistem digunakan persamaan berikut

$$A = \left(\frac{TP+TN}{TP+FP+FN+TN} \right) \times 100\% \quad (3.3)$$

- **F1 Score (F1)**, yaitu perbandingan rata-rata precision dan recall yang dibobotkan. Untuk menghitung nilai *f1 score* sistem digunakan persamaan berikut:

$$F1 = 2 \left(\frac{P \times R}{P+R} \right) \times 100\% \quad (3.4)$$

Setelah dilakukan perancangan sistem untuk deteksi dan perhitungan buah pada pohonnya, dilanjutkan dengan pengujian data pada sistem yang telah dirancang dan pengujian unjuk kerja sistem. Pada tahap ini dilakukan perhitungan nilai akurasi jumlah buah pada 11 pohon berbeda dengan tiga jarak yang berbeda dengan menggunakan persamaan berikut:

$$Ac = \frac{TP+TN}{Jumlah\ Buah\ Sebenarnya} \times 100\% \quad (3.5)$$

Nilai akurasi kinerja sistem (Ac) diperoleh dengan persentase jumlah buah prediksi yang benar (TP+TN) atau (Ns) terhadap jumlah buah sebenarnya (Nt).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bab ini menyajikan hasil dari pengujian sistem deteksi dan perhitungan buah cokelat matang dan non matang pada pohonnya dengan menggunakan metode *deep learning* YOLO. Langkah utama dalam pembentukan model ialah dengan proses *training* data. Proses ini bertujuan agar model dapat mengenali dataset yang akan digunakan. Kemudian setelah training nantinya model yang telah terbentuk akan diuji menggunakan data *testing*. Adapun jumlah data yang digunakan adalah sebanyak 674 data, dengan perbandingan 80% untuk *training* dan 20% yang digunakan menjadi data *testing* terhadap sistem yang diambil menggunakan tiga variasi jarak.

Pada penelitian ini, dilakukan percobaan dengan menggunakan parameter *batch* = 64, *subdivision* = 16, *width* = 416, *height* = 416, *momentum* = 0.949, *channels* = 3, *learning rate* = 0.001, dan *max batch* = 6000. Untuk pengukuran kinerja klasifikasi digunakan *confussion matrix*. Perhitungan jumlah TP, FP, FN, dan TN pada citra keluaran dilakukan secara manual oleh peneliti sebagai berikut.

		Aktual	
		Matang	Non Matang
Prediksi	Matang	220	5
	Non Matang	16	377

Tabel 4.1 Confusion Matrix

TP	FP	FN	TN	P	R	A	F1
220	5	16	377	97%	96%	97%	96%

Tabel 4.2 Confusion Matrix Setiap Jarak

Jarak	TP	FP	FN	TN	P	R	A	F1
100 cm	69	0	7	128	97%	95%	97%	96%
150 cm	74	0	4	129	98%	97%	98%	98%
200 cm	77	5	5	120	95%	95%	95%	95%

Adapun akurasi perhitungan jumlah buah dari 11 pohon berbeda menggunakan tiga variasi jarak adalah sebagai berikut.

Jarak 100 cm

Hasil pengujian deteksi dan perhitungan buah pada jarak 100 cm dapat dilihat pada **Tabel 4.3**.

Tabel 4.3 Hasil Uji Deteksi dan Hitung Buah pada Jarak 100 cm

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 1	Matang	1	1	100	
	Non	5	7	71	

Tabel 4.3 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 2	Matang	1	1	100	 Buah Non Matang Terdeteksi: 0 Buah Matang Terdeteksi: 1
	Non	2	3	67	 Buah Non Matang Terdeteksi: 3 Buah Matang Terdeteksi: 1
Pohon 3	Matang	0	0	-	
	Non	7	7	100	 Buah Non Matang Terdeteksi: 7 Buah Matang Terdeteksi: 0
Pohon 4	Matang	1	1	100	 Buah Matang Terdeteksi: 1 Buah Non Matang Terdeteksi: 1
	Non	1	1	100	 Buah Matang Terdeteksi: 1 Buah Non Matang Terdeteksi: 1
Pohon 5	Matang	0	0	-	
	Non	4	4	100	 Buah Non Matang Terdeteksi: 4 Buah Matang Terdeteksi: 0
Pohon 6	Matang	0	2	0	 Buah Non Matang Terdeteksi: 1 Buah Matang Terdeteksi: 2
	Non	7	7	100	 Buah Non Matang Terdeteksi: 7 Buah Matang Terdeteksi: 0

Tabel 4.3 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 7	Matang	1	1	100	
	Non	3	3	100	
Pohon 8	Matang	2	2	100	
	Non	1	1	100	
Pohon 9	Matang	5	6	83	
	Non	1	1	100	
Pohon 10	Matang	5	5	100	
	Non	2	2	100	
Pohon 11	Matang	1	1	100	
	Non	2	2	100	

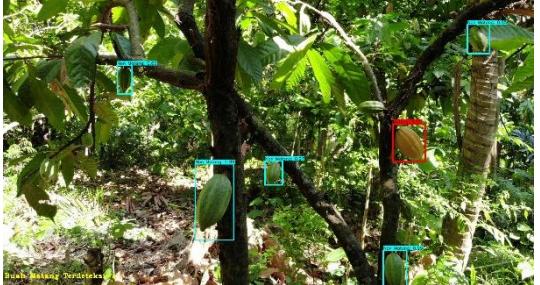
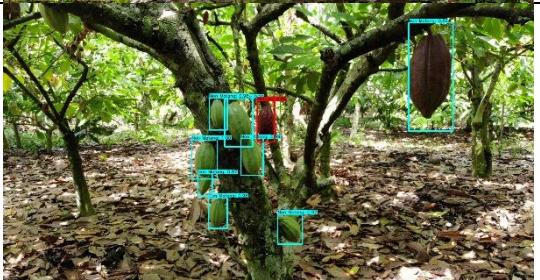
Tabel 4.3 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Total	Buah	52	58	90	
	Matang	17	20	85	
	Non	35	38	92	

Jarak 150 cm

Hasil pengujian deteksi dan perhitungan buah pada jarak 150 cm dapat dilihat pada **Tabel 4.4**.

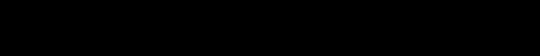
Tabel 4.4 Hasil Uji Deteksi dan Hitung Buah pada Jarak 150 cm

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 1	Matang	1	1	100	
	Non	5	8	63	
Pohon 2	Matang	1	1	100	
	Non	2	3	67	
Pohon 3	Matang	0	0	-	
	Non	7	7	100	

Tabel 4.4 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 4	Matang	1	1	100	
	Non	3	3	100	
Pohon 5	Matang	0	0	-	
	Non	4	4	100	
Pohon 6	Matang	1	2	50	
	Non	3	7	43	
Pohon 7	Matang	1	1	100	
	Non	3	4	75	
Pohon 8	Matang	2	2	100	
	Non	2	2	100	

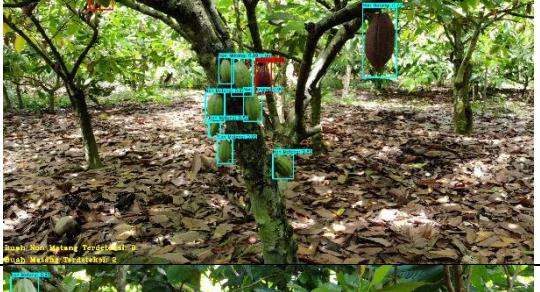
Tabel 4.4 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 9	Matang	6	6	100	
	Non	1	1	100	
Pohon 10	Matang	5	5	100	
	Non	2	2	100	
Pohon 11	Matang	1	1	100	
	Non	2	2	100	
Total	Buah	53	63	84	
	Matang	19	20	95	
	Non	34	43	79	

Jarak 200 cm

Hasil pengujian deteksi dan perhitungan buah pada jarak 200 cm dapat dilihat pada **Tabel 4.5**.

Tabel 4.5 Hasil Uji Deteksi dan Hitung Buah pada Jarak 200 cm

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 1	Matang	1	1	100	
	Non	4	9	44	
Pohon 2	Matang	0	1	0	
	Non	2	3	67	
Pohon 3	Matang	1	1	100	
	Non	7	7	100	
Pohon 4	Matang	1	1	100	
	Non	4	4	100	
Pohon 5	Matang	0	0	-	
	Non	4	4	100	

Tabel 4.5 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 6	Matang	0	2	0	
	Non	4	8	50	
Pohon 7	Matang	1	1	100	
	Non	3	4	75	
Pohon 8	Matang	2	2	100	
	Non	2	2	100	
Pohon 9	Matang	7	7	100	
	Non	1	1	100	
Pohon 10	Matang	5	5	100	
	Non	2	2	100	

Tabel 4.5 (Lanjutan)

Nama		Ns	Nt	Akurasi (%)	Visual
Pohon 11	Matang	1	1	100	
	Non	2	2	100	
Total	Buah	54	68	79	
	Matang	19	22	86	
	Non	35	46	76	

4.2 Pembahasan

Sistem yang dibuat menggunakan beberapa parameter dalam proses pelatihan yaitu sebagai berikut:

- *Batch*

Parameter ini menunjukkan ukuran batch yang digunakan dalam proses training. *Batch* merupakan berapa banyak data yang akan di proses dalam satu waktu dan disebar ke *neural network*. Dalam sistem yang dibuat ini, nilai *batch size* adalah 64 yang berarti satu iterasi terdapat 64 citra yang digunakan untuk mengupdate bobot dari *neural network*.

- *Subdivision*

Parameter ini menunjukkan nilai yang digunakan untuk membagi nilai *batch*. Dengan memberikan nilai *subdivision* menjadi 16, maka data yang akan diproses adalah 64/16 atau 4 citra dalam satu waktu. Hal ini

dilakukan untuk menghindari error akibat GPU kekurangan memori.

Saat pengujian, nilai batch dan subdivision akan di atur menjadi 1.

- *Width, Height, dan Channel*

Ketiga parameter ini menunjukkan ukuran citra yang akan menjadi masukan. Nilai untuk $width \times height$ yang digunakan adalah 416×416 piksel (bisa berupa kelipatan 32, 416 adalah standar). Hasil deteksi bisa saja lebih baik jika ukuran input diubah dengan membuat lebih besar misalnya menjadi 608×608 piksel, akan tetapi akan membutuhkan waktu *training* yang jauh lebih lama dan sistem pendeksi akan menjadi lebih lambat. Sementara nilai *channel* = 3 untuk menunjukkan bahwa citra yang akan diproses adalah citra dengan 3 *channel*, yaitu RGB (*Red, Green, Blue*).

- *Momentum*

Nilai bobot yang terdapat pada sebuah *neural network* di perbaharui dengan 1 batch data citra dan tidak dengan seluruh data citra secara bersamaan. Hal ini akan menyebabkan fluktuasi pada proses perbaruan weight. Nilai momentum 0.9 diberikan sebagai pembatas bahwa perubahan nilai bobot tidak boleh melebihi 0.9 dari weight sebelum proses update.

- *Learning Rate, Steps, dan Scale*

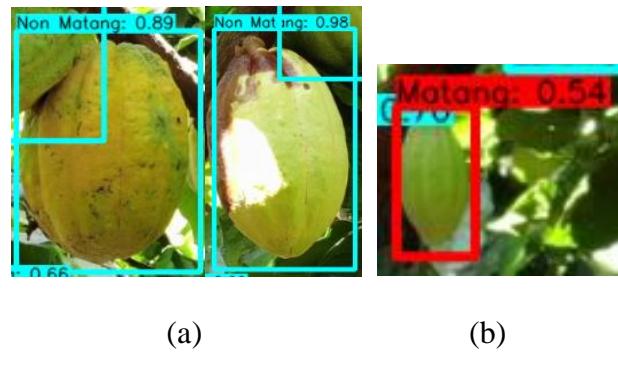
Learning rate mengatur seberapa agresif sistem akan belajar berdasarkan batch data masukannya. Nilai *learning rate* pada umumnya berkisar antara 0.01 sampai 0.0001. Pada sistem ini, nilai

learning rate yang digunakan adalah 0.001. *Steps* adalah parameter yang digunakan sebagai *threshold* untuk *learning rate*, nilai *steps* pada sistem ini adalah 4800, 5400, yang berarti nilai *learning rate* akan diperkecil dengan cara mengalikannya dengan parameter *scale* yang memiliki nilai 0.1, apabila iterasi sudah mencapai *steps* yang ditentukan.

- *Max Batches (Iterations)*

Max batches menspesifikasikan seberapa banyak pengulangan yang harus dilakukan oleh sistem untuk menyelesaikan proses training. Dalam sistem ini nilai *max batches* diatur pada 6000 iterasi.

Dengan menggunakan parameter-parameter di atas, sistem yang telah dibuat dilakukan *training* dengan jumlah data *training* sebanyak 542 data, kemudian dilakukan proses *testing* yang memberikan hasil klasifikasi rata-rata sebesar 97%. Sistem yang dibuat memiliki nilai *false positive* sebanyak 5 buah dan *false negative* sebanyak 16 buah, menandakan bahwa ada 19 buah yang salah prediksi dari data *testing*, yang seharusnya kelas matang tetapi diprediksi oleh sistem menjadi kelas non matang dan begitupun sebaliknya. Hal ini disebabkan karena kedua kelas tersebut memiliki kemiripan ciri-ciri yaitu bentuk dan warna buahnya. Adapun kesalahan klasifikasi dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Contoh kesalahan klasifikasi *false positive* (a) dan *false negative* (b)

Hasil akurasi rata-rata perhitungan buah untuk setiap jarak masing-masing sebesar 90%, 84%, 79%. Penyebab sistem tidak dapat mendeteksi objek sebagai buah adalah adanya buah yang mendapat oklusi dari batang dapat dilihat pada **Gambar 4.2** dan juga buah yang terkena cahaya dari matahari dapat dilihat pada **Gambar 4.3**.



Gambar 4.2 Buah teroklusi batang



Gambar 4.3 Buah terkena cahaya matahari

Untuk mencegah terjadinya kesalahan dalam klasifikasi objek, sistem yang dibuat diberi *threshold* 0.5 artinya objek yang dideteksi hanya akan divisualisasikan *bounding box* nya apabila *score* kelasnya diatas 0.5. Akan tetapi dalam beberapa kasus, buah kakao matang memiliki *score* kelas diatas 0.5 bukan pada kelas sebenarnya, hal ini dipengaruhi oleh warna pada buah yang memiliki kemiripan dengan warna buah kakao non matang begitupun sebaliknya sehingga terprediksi dengan *score* yang cukup tinggi.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada sistem deteksi dan hitung buah cokelat matang dan non matang pada pohonnya dengan tiga variasi jarak, maka dapat disimpulkan bahwa:

1. Sistem deteksi dan hitung buah cokelat matang dan non matang pada pohonnya dibuat dengan menggunakan algoritma YOLOv4 sebagai algoritma untuk mendeteksi objek. Sistem ini mampu mendeteksi buah sesuai kelasnya yaitu matang dan non matang serta menghitung buah langsung pada pohonnya.
2. Hasil akurasi klasifikasi sistem dengan jarak 100 cm, 150 cm, dan 200 cm masing-masing yaitu, 97%, 98%, dan 95% serta menghasilkan rata-rata akurasi sebesar 97%. sedangkan hasil rata-rata akurasi perhitungan buah perpohon sistem dengan jarak 100 cm, 150 cm, dan 200 cm masing-masing yaitu, 90%, 84%, dan 79%. Sehingga disimpulkan bahwa akurasi klasifikasi sistem tidak berpengaruh pada jarak. Sedangkan akurasi perhitungan buah perpohon sistem berpengaruh, dimana semakin jauh sebuah objek maka semakin rendah akurasinya.

5.2 Saran

Sehubungan dengan selesainya penulisan skripsi ini, penulis bermaksud untuk menyampaikan beberapa saran terhadap pengembangan sistem, antara lain:

1. Diharapkan pada penelitian selanjutnya dapat menambahkan lebih banyak jumlah data agar sistem lebih akurat dalam mendeteksi buah.
2. Sistem yang telah dibuat dapat dimodifikasi untuk mendeteksi objek lain selain buah cokelat dengan menggantikan data latih untuk mengatasi permasalahan-permasalahan lainnya.

DAFTAR PUSTAKA

- Arief, A. M. (2020). *Ironis, Indonesia Subur Tapi Pengimpor Kakao*. Retrieved from Bisnis.com: <https://ekonomi.bisnis.com/read/20201008/99/1302242/ironis-indonesia-subur-tapi-pengimpor-kakao>
- BALITBANGTAN. (2019). *Teknologi Balitbangtan: Pembuahan Kakao di Luar Musim*. Retrieved from Badan Penelitian dan Pengembangan Pertanian: <https://new.litbang.pertanian.go.id/info-teknologi/3771/>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv:2004.10934v1 [cs.CV].
- Dhiaegana, R. N., & Munir, R. (2020). *Penerapan Convolutional Neural Network untuk Deteksi Pedestrian pada Sistem Autonomous Vehicle*. Bandung: Institut Teknologi Bandung.
- Dompeipen, T., Sompei, S., & Najoan, M. (2021). Computer Vision Implementation for Detection and Counting the Number of Humans. *Jurnal Teknik Informatika* vol.16 no.1, 65-76.
- Ekawaty, Y. (2019). *DETEKSI DAN HITUNG OTOMATIS BUAH COKELAT PADA POHON SECARA REALTIME MENGGUNAKAN DEEP LEARNING*. Makassar: Universitas Hasanuddin.
- Hatta K, P. (2017, Mei 16). *Menilik Activation Function*. Retrieved from Medium: <https://medium.com/@pramestihattak/menilik-activation-functions-7710177a54c9>
- Irawan, F. S. (2020, November 25). *Convolutional Neural Network (CNN)*. Retrieved from Kotakode: [https://kotakode.com/blogs/2707/Convolutional-Neural-Network-\(CNN\)](https://kotakode.com/blogs/2707/Convolutional-Neural-Network-(CNN))
- Kelvin. (2020). *APLIKASI PENGHITUNG JUMLAH PERENANG PADA KOLAM RENANG*. Makassar: Universitas Hasanuddin.
- KEMENTAN. (2018). *Kementan Dorong Pemanfaatan Industri 4.0 Sektor Pertanian*. Retrieved from Kementerian Pertanian Republik Indonesia: <https://www.pertanian.go.id/home/?show=news&act=view&id=3399>
- Kusuma, T., Usman, K., & Saidah, S. (2021). PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD. *Jurnal Teknik Informatika (JUTIF)*, 57-66.
- Misra, D. (2020). *Mish: A Self Regularized Non-Monotonic Activation Function*. India: arXiv:1908.08681v3 [cs.LG].

- Nas, M. (2021, Juni 27). *KRITERIA BUAH KAKAO SIAP PANEN*. Retrieved from CYBEXT: <http://cybex.pertanian.go.id/mobile/artikel/98228/KRITERIA-BUAH-KAKAO-SIAP-PANEN/>
- Prabowo, D., Abdullah, D., & Manik, A. (2018). DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN COLOR OBJECT TRACKING. *Pseudocode, Volume V Nomor 2*.
- Putri, O. N. (2020). *IMPLEMENTASI METODE CNN DALAM KLASIFIKASI GAMBAR JAMUR PADA ANALISIS IMAGE PROCESSING*. YOGYAKARTA: UNIVERSITAS ISLAM INDONESIA.
- Solawetz, J. (2020, Juni 4). *YOLOv4 Explained*. Retrieved from roboflow: <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>
- Syahputra, M. (2021). *Pertumbuhan Bibit Tanaman Kakao (*Theobroma cacao L.*) dengan Pemberian Pupuk Trichokompos Tandan Kosong Kelapa Sawit*. Pekanbaru: UIN Suska Riau.
- Widyawati, W., & Febriani, R. (2021). Real-time detection of fruit ripeness using the YOLOv4 algorithm. *TEKNIKA: JURNAL SAINS DAN TEKNOLOGI VOL 17 NO 02, 205-210*.

LAMPIRAN

1. *Source Code*

Source Code yang digunakan pada penelitian ini dapat dilihat pada pranala berikut <https://github.com/muhraedi/skripsi>

2. *Dataset*

Dataset yang digunakan pada penelitian ini dapat dilihat pada pranala berikut <https://www.kaggle.com/datasets/muhraedi/kakao-tree>

LEMBAR PERBAIKAN SKRIPSI

**“DETEKSI DAN HITUNG BUAH COKELAT MATANG DAN
NON-MATANG PADA POHONNYA SECARA REALTIME”**

OLEH:

**MUHAMMAD RAEDI RADIFAN
D42116304**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 19 Oktober 2022.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

Jabatan	Nama	Tanda Tangan
Ketua	Dr. Indrabayu, S.T., M.T., M.Bus.sys., IPM	
Sekretaris	Dr. Ir. Ingrid Nurtanio, M.T.	
Anggota	Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.	
	Iqra Aswad, ST., M.T.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Indrabayu, S.T., M.T., M.Bus.sys., IPM	
II	Dr. Ir. Ingrid Nurtanio, M.T.	