

HASIL PENELITIAN

DETEKSI DAN HITUNG OTOMATIS BUAH COKELAT PADA POHON SECARA REALTIME MENGGUNAKAN DEEP LEARNING



YULIA EKAWATY

D032172015

DOSEN PEMBIMBING:

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

Dr. Eng. Intan Sari Areni, ST, MT

**PROGRAM PASCASARJANA
UNIVERSITAS HASANUDDIN
MAKASSAR
2019**

LEMBAR PENGESAHAN

Judul Thesis : Deteksi dan Hitung Otomatis Buah Cokelat Pada
Pohon Secara *Realtime* Menggunakan *Deep Learning*

Nama : Yulia Ekawaty

NIM : D032172015

Program Studi : Teknik Elektro

Konsentrasi : Teknik Informatika

Diajukan sebagai salah satu syarat akademik pada program Pascasarjana
Universitas Hasanuddin Makassar.

Menyetujui

Pembimbing I,

Pembimbing II,

Dr. Indrabayu, ST.,M.Bus.Sys.,MT
NIP: 197507162002121004

Dr. Eng. Intan Sari Areni, ST.,MT
NIP: 197502032000122002

Ketua Program Studi

Prof. Dr. Eng. Syafaruddin, ST., M.Eng
NIP: 197405301999031001

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas selesainya tesis berjudul “**Deteksi dan Hitung Otomatis Buah Cokelat Pada Pohon Secara *Realtime* Menggunakan *Deep Learning***” serta shalawat dan salam dihaturkan kepada Nabi Muhammad SAW yang mengajarkan wahyu Allah SWT dan budi pekerti yang baik dan mulia bagi manusia.

Ucapan terima kasih penulis hanturkan kepada: Dr. Indrabayu, ST.,MT.,M.Bus.Sys dan Dr. Eng. Intan Sari Areni, ST.,MT. selaku pembimbing, juga kepada para Bapak/Ibu penguji yang telah banyak memberikan masukan, serta kepada para staf Departemen Teknik Elektro Universitas Hasanuddin yang telah memberikan bantuan administrasi pada tahap penyelesaian penelitian tesis ini.

Tak lupa pula ucapan terima kasih kepada *AIMP Family* yang telah menjadi keluarga serta pemberi semangat dan ilmu luar biasa, juga kepada teman seperjuangan mahasiswa pascasarjana Teknik Informatika angkatan 2018, yang telah menjadi sahabat dalam suka dan duka dalam menyelesaikan penelitian ini.

Penulis menyadari bahwa tesis ini masih perlu masukan dan saran agar bisa menuju ke arah yang lebih sempurna sehingga bisa memberikan manfaat kepada seluruh pihak.

Makasar,2019

Penulis

ABSTRAK

YULIA EKAWATY. Deteksi dan Hitung Otomatis Buah Cokelat pada Pohonnya Secara *Realtime* Menggunakan *Deep Learning* (dibimbing oleh Indrabayu dan Intan Sari Areni).

Penelitian ini bertujuan untuk mendeteksi dan menghitung otomatis buah cokelat pada pohonnya secara *realtime* dengan metode *deep learning: You Only Look Once* (YOLO). Skenario pengambilan data dalam penelitian ini dilakukan dengan mengambil data buah cokelat pada pohon menggunakan drone dengan 3 variasi jarak (d) yaitu $d = 100$ cm, $d = 150$ cm dan $d = 200$ cm yang kemudian dilakukan proses streaming ke sebuah PC agar data dapat diolah secara *realtime*. Data dibagi menjadi 2 bagian yaitu data latih sebanyak 180 gambar dan data uji sebanyak 45 gambar. Desain sistem dimulai dengan *Preprocessing*, *Feature Extraction*, dan *Classification*, hingga menghasilkan sistem deteksi buah cokelat. Pada penelitian ini juga dilakukan perhitungan akurasi jumlah buah coklat per pohon yang menghasilkan akurasi tertinggi 87.2% di $d = 100$ cm dan terendah 73.6 di $d = 200$ cm.

Kata kunci: deteksi buah cokelat, *deep learning*, *CNN You Only Look Once* (YOLO), *drone*, *realtime*

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
ABSTRAK	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
A. LATAR BELAKANG	1
B. RUMUSAN MASALAH.....	5
C. TUJUAN PENELITIAN.....	6
D. MANFAAT PENELITIAN.....	6
E. BATASAN MASALAH.....	7
F. SISTEMATIKA PENULISAN.....	7
BAB II TINJAUAN PUSTAKA	10
A. LANDASAN TEORI.....	10
1) Tanaman Cokelat (<i>theobroma cacao</i>).....	10
2) Visi Komputer.....	12
3) Pengolahan Citra Digital	13
4) <i>Convolutional Neural Network (CNN)</i>	20
5) <i>You Only Look Once (YOLO)</i>	30

6) Darknet-53	Error! Bookmark not defined.
7) <i>Drone</i> DJI Mavic Air	36
B. PENELITIAN TERKAIT	40
C. STATE OF THE ART	41
D. KERANGKA PIKIR	44
BAB III METODE PENELITIAN	45
A. TAHAPAN PENELITIAN	45
B. WAKTU DAN LOKASI PENELITIAN	47
1) Waktu.....	47
2) Lokasi	47
C. JENIS PENELITIAN.....	47
D. PENGAMBILAN DATA	47
E. PERANCANGAN SISTEM.....	50
F. INSTRUMENTASI PENELITIAN.....	64
G. SUMBER DATA.....	65
H. TOLAK UKUR AKURASI SISTEM.....	65
BAB IV HASIL DAN PEMBAHASAN	67
BAB V KESIMPULAN DAN SARAN	75

DAFTAR GAMBAR

Gambar 1.	Buah coklat jenis Sulawesi 1	12
Gambar 2.	Contoh matriks citra <i>RGB</i> (Courtney, 2001).	17
Gambar 3.	Proses utama pada CNN.....	Error! Bookmark not defined.
Gambar 4.	Arsitektur <i>CNN</i>	Error! Bookmark not defined.
Gambar 5.	Arsitektur <i>Network YOLOv3</i> (medium.com)	31
Gambar 6.	Model <i>Layer Network Darknet-53</i> (medium.com)	33
Gambar 7.	<i>Process flow</i> dari <i>YOLOv3</i> (Murugavel, 2018, p. 3).....	36
Gambar 8.	Drone DJI Mavic Air	38
Gambar 9.	Kerangka Pikir Penelitian.....	44
Gambar 10.	Tahapan penelitian	45
Gambar 11.	Pengambilan data di lapangan	48
Gambar 12.	Desain Sistem Pengambilan Data	48
Gambar 13.	Alur data <i>video streaming</i> dari <i>drone</i>	49
Gambar 14.	Rancangan sistem	51
Gambar 15.	Contoh gambar masukan untuk 3 variasi jarak (a) $d=100\text{cm}$, (b) $d=150\text{ cm}$, (c) $d=200\text{ cm}$	52
Gambar 16.	Flowchart Labelling pada <i>Yolo Mark</i>	53
Gambar 17.	Pelabelan manual yang dilakukan pada <i>Yolo Mark</i>	54
Gambar 18.	Hasil pelabelan gambar	55
Gambar 19.	Convolution Layer	Error! Bookmark not defined.
Gambar 20.	Proses kerja <i>Max Pooling</i>	Error! Bookmark not defined.

Gambar 21. Proses <i>reshape activation map</i> dari multi dimensional array menjadi vektor.	Error! Bookmark not defined.
Gambar 22. <i>Fully connected layer</i>	61
Gambar 23. Atribut bounding box output dari FC layer	Error! Bookmark not defined.
Gambar 24. Hasil deteksi buah coklat menggunakan YOLO model.....	64
Gambar 25. Contoh hasil deteksi dan hitung otomatis jarak 100 cm.....	67
Gambar 26. Contoh hasil deteksi dan hitung otomatis jarak 150 cm.....	68
Gambar 27. Contoh hasil deteksi dan hitung otomatis jarak 200 cm.....	68
Gambar 28. Objek yang mendapat oklusi daun dan bisa terdeteksi untuk $d=100$ cm.....	72

DAFTAR TABEL

Tabel 1.	State of The Art.....	41
Tabel 2.	Akurasi sistem <i>YOLO</i>	69
Tabel 3.	Akurasi sistem $d = 100\text{cm}$, $d = 150\text{ cm}$, $d = 200\text{ cm}$	73
Tabel 4.	Uji sistem data negatif.....	74

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Dunia industri telah memasuki fase keempat dari perjalanan sejarah revolusi industri atau lebih dikenal dengan Revolusi Industri 4.0 dan telah merambah semua bidang termasuk bidang pertanian. Sejalan hal tersebut pemerintah Indonesia menyusun peta jalan dan strategi Making Indonesia 4.0, yang diluncurkan Presiden Joko Widodo pada 4 April 2018 ("Kementerian Perindustrian," n.d.)

Dalam bukunya yang berjudul "*The Fourth Industrial Revolution*", Schawab (2017) menjelaskan revolusi industri 4.0 telah mengubah hidup dan kerja manusia secara fundamental. Berbeda dengan revolusi industri sebelumnya, revolusi industri generasi ke-4 ini memiliki skala, ruang lingkup dan kompleksitas yang lebih luas. Kemajuan teknologi baru yang mengintegrasikan dunia fisik, digital dan biologis telah mempengaruhi semua disiplin ilmu. Revolusi Industri 4.0 juga mendorong sistem otomatisasi di dalam semua proses aktivitas (Rosyadi, 2018). Otomatisasi tersebut termasuk di bidang pertanian dimana salah satu komoditas pertanian yang unggul dan mendapatkan perhatian khusus adalah tanaman coklat (*theobroma cacao*) yang sampai saat ini Indonesia masih melakukan impor untuk memenuhi kebutuhan dalam negeri.

Seiring berkembangnya teknologi, begitu banyak pengembangan dan inovasi yang dilakukan untuk penelitian dalam bidang pertanian dan perkebunan cokelat khususnya pada *yield estimation* dengan hasil yang belum maksimal dan masih dipandang perlu untuk terus dikembangkan menuju hasil yang diharapkan. Dan untuk menjawab tantangan era digitalisasi saat ini, penulis berupaya melakukan pengembangan penelitian dengan menggunakan *UAV (Unmanned Aerial Vehicle)* atau biasa disebut *drone*.

Drone merupakan pesawat tanpa awak yang cukup inovatif karena mengandalkan IoT (*Internet of Things*) dan AI (*Artificial Intelligent*). Dahulu pemanfaatan drone sangat terbatas misalnya hanya digunakan untuk kepentingan militer namun saat ini penggunaannya telah merambah segala bidang termasuk bidang pertanian.

Kehadiran teknologi drone dalam sektor pertanian bisa mendatangkan banyak manfaat. Teknologi ini bisa memberikan solusi revolusioner dalam meningkatkan kualitas serta kuantitas hasil panen. Manfaat lain teknologi drone untuk pertanian antara lain: drone bisa membantu dalam melakukan persiapan sebelum musim tanam. Petani bisa menggunakan perangkat ini untuk mengetahui kondisi kebun secara menyeluruh. Drone juga memiliki kemampuan dalam melakukan pemetaan kondisi kebun secara 3 dimensi. Selanjutnya, data tersebut dapat digunakan untuk analisis kondisi lahan. Hasil analisis bisa dipakai untuk menentukan pola penanaman bibit yang maksimal . Keberadaan drone juga bisa dimanfaatkan untuk melakukan

penanaman bibit. Cara penanaman bibit menggunakan drone memungkinkan proses penanaman berlangsung dengan lebih cepat. Drone tidak hanya punya kemampuan untuk menembakkan bibit ke permukaan tanah, tetapi juga pemupukan. Di waktu yang sama, keberadaannya juga berguna untuk mengurangi biaya untuk ongkos penanaman mencapai 85 persen. Drone dapat pula digunakan untuk aktivitas penyemprotan tanaman di kebun. Drone terbaru memiliki beragam fitur yang memungkinkan penyemprotan bisa dilakukan secara efektif dan maksimal. Saat ini dapat dijumpai jenis drone untuk pertanian yang disertai dengan kemampuan mengeluarkan suara ultrasonik ataupun laser yang berguna dalam mengenali struktur kondisi lahan. Fitur ini memungkinkan drone dapat melakukan penyemprotan secara lancar tanpa terjadi tabrakan. Selain itu, fitur tersebut juga dapat dimanfaatkan dalam melakukan pemindaian permukaan tanah. Selanjutnya, hasil pemindaian bisa menjadi data awal untuk menentukan jumlah cairan yang perlu disemprotkan. Semua itu bisa dilakukan secara real-time dengan jangkauan yang luas.

Karena drone telah *familiar* penggunaannya dalam bidang pertanian, beberapa peneliti sebelumnya telah menggunakan drone dalam penelitiannya misalnya: Estimasi Hasil Panen Kubis dengan menggunakan drone (Dita Awaldi, 2018). Uktoro, Arief pada tahun 2017 juga menggunakan drone pada penelitian yang berjudul : Analisis Citra Drone Untuk Monitoring Kesehatan Tanaman Kelapa Sawit. Karena itu, dalam penelitian ini penulis juga melakukan deteksi dan hitung buah coklat secara otomatis yang

pengambilan datanya menggunakan drone. Penelitian ini sebagai langkah awal untuk *yield estimation* agar persiapan panen dan pasca panen coklat juga dapat diprediksi dengan lebih cepat dan akurat dengan bantuan teknologi drone.

Tahap awal penelitian dilakukan observasi peninjauan secara langsung pada sebuah perusahaan yang bergerak di bidang perkebunan coklat, terdapat beberapa masalah terkait proses persiapan panen, masa panen dan pasca panen. Petani coklat terkendala untuk mengestimasi hasil panennya. Terdapat pula kendala lain yaitu saat investor memerlukan data perkebunan coklat khususnya terkait hasil panen, diperlukan waktu yang cukup lama karena semuanya dilakukan secara manual. Untuk mengatasi kendala tersebut diusulkan untuk penggunaan teknologi yang tahap awalnya dimulai dengan mendeteksi dan menghitung otomatis buah coklat pada pohon menggunakan metode *deep learning* untuk mendapatkan hasil deteksi yang lebih akurat .

Sebelum memutuskan menggunakan metode *deep learning*, sebelumnya penulis juga melakukan penelitian menggunakan metode *image processing* dengan metode *in range* warna yaitu RGBtoHSV. Dalam penelitian ini penulis menemui beberapa kendala untuk menghasilkan akurasi yang maksimal disebabkan antara lain adanya iluminasi cahaya saat pengambilan data yang menyebabkan tidak seragamnya warna permukaan buah sehingga mempengaruhi pengolahan data selanjutnya.

Berdasarkan berbagai referensi yang penulis kumpulkan akhirnya diputuskan menggunakan metode *deep learning* yang dipandang cocok untuk meminimalisasi kendala pada metode yang sebelumnya. Sedangkan penggunaan drone untuk tahap pengambilan data pada penelitian ini dimaksudkan untuk efisiensi waktu dan tenaga kerja serta memperoleh data yang *realtime*. Diharapkan juga dengan bantuan teknologi ini seluruh persiapan pra panen, masa panen dan pasca panen lebih baik lagi.

Dalam penelitian ini juga dilakukan pengambilan data dengan 3 variasi jarak untuk mengetahui jarak ideal antara kamera drone dengan objek. Dilakukan pula penghitungan akurasi hasil deteksi sistem pada berbagai jarak dari masing-masing pohon yang.

B. RUMUSAN MASALAH

Berdasarkan latar belakang maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana mengimplementasikan metode *deep learning* untuk deteksi dan hitung otomatis buah cokelat pada pohon?
2. Bagaimana unjuk kerja sistem deteksi dan hitung otomatis buah cokelat menggunakan *deep learning*?

C. TUJUAN PENELITIAN

Adapun tujuan yang akan dicapai dari penelitian ini adalah:

1. Untuk membuat sistem deteksi dan hitung otomatis buah cokelat pada pohon menggunakan metode *deep learning*.
2. Untuk mengetahui unjuk kerja sistem deteksi dan hitung otomatis buah cokelat menggunakan metode *deep learning*.

D. MANFAAT PENELITIAN

Manfaat dari penelitian adalah:

1. Bagi bidang pertanian, penelitian yang dilakukankan memberikan kontribusi otomasi sehingga menghemat tenaga kerja dan waktu. Dengan sistem deteksi buah coklat secara otomatis, maka petani perkebunan coklat khususnya perkebunan skala industri bisa mendapatkan estimasi hasil perkebunan yang lebih akurat untuk digunakan sebagai data pendukung persiapan panen dan pasca panen. Penggunaan teknologi drone pada penelitian ini juga sejalan dengan kebijakan pemerintah agar menerapkan teknologi di berbagai bidang untuk mendukung Revolusi Industri 4.0.
2. Bagi peneliti, pengaplikasian dari sejumlah konsep dan pengetahuan yang berguna untuk menambah pengetahuan dan kemampuan/*skill* mengenai proses untuk mendeteksi dan menghitung buah cokelat pada pohonnya secara *realtime*.

3. Bagi institusi pendidikan magister Departemen Teknik Elektro bidang Informatika, dapat digunakan sebagai referensi ilmiah dalam penelitian untuk pengembangan sistem untuk mendeteksi dan menghitung dengan objek yang berbeda.

E. BATASAN MASALAH

Adapun batasan masalah penelitian ini yaitu:

1. Objek buah cokelat berjenis Sulawesi 1 (S1) pada pohon diambil dalam kondisi cuaca cerah dan menggunakan *drone* .
2. Rancangan model pendeteksi buah cokelat dibuat menggunakan *deep learning You Only Look Once (YOLO)*

F. SISTEMATIKA PENULISAN

Adapun sistematika penulisan pada peneltian ini sebagai berikut:

Bab I Pendahuluan

Bab I berisi penjelasan tentang latar belakang penelitian untuk mendeteksi serta menghitung otomatis buah cokelat dengan menggunakan *deep learning* dengan beberapa variasi jarak antara objek dengan kamera.

Bab II Landasan Teori dan Kerangka Pemikiran

Bab II berisi penjelasan tentang tinjauan pustaka terkait landasan teori tanaman cokelat, proses *deep learning*, metode-metode yang digunakan, penelitian terkait dengan sistem deteksi dan penghitungan buah cokelat pada pohon, dan kerangka pikir penelitian. 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 serta hitung otomatis buah cokelat pada pohon dengan menggunakan beberapa variasi jarak.

Bab III Metodologi Penelitian

Bab III ini merupakan penjelasan tentang tahapan penelitian, waktu dan lokasi penelitian, jenis penelitian yang akan diusulkan, rancangan sistem dengan metode yang diusulkan beserta ilustrasi gambar sistem, sumber data dalam menyusun tesis ini, instrumen yang digunakan, serta jadwal penelitian.

Bab IV Hasil dan Pembahasan

Bab IV berisi penjelasan tentang hasil dan pembahasan dari penelitian yang telah dilakukan serta implikasi dari penelitian yang dilakukan. Hasil merupakan penjelasan tentang pengolahan citra menggunakan data video yang telah diambil dengan kamera *drone* dan dilakukan perhitungan akurasi sistem berdasarkan jarak (d).

Bab V Kesimpulan dan Saran

Bab V peneliti menjelaskan ringkasan hasil dari penelitian dalam mendeteksi dan menghitung otomatis buah cokelat pada pohon menggunakan *deep learning* dengan metode *You Only Look Once* (YOLO). Selain itu, peneliti menjelaskan beberapa saran untuk pengembangan penelitian selanjutnya agar hasil penelitian menjadi lebih optimal.

BAB II

TINJAUAN PUSTAKA

A. LANDASAN TEORI

1) Tanaman Cokelat (*theobroma cacao*)

Kakao merupakan salah satu komoditas andalan perkebunan yang peranannya cukup penting bagi perekonomian nasional, khususnya sebagai penyedia lapangan kerja, sumber pendapatan dan devisa negara. Di samping itu kakao juga berperan dalam mendorong pengembangan wilayah dan pengembangan agroindustri. Pada tahun 2002, perkebunan kakao telah menyediakan lapangan kerja dan sumber pendapatan bagi sekitar 900 ribu kepala keluarga petani yang sebagian besar berada di Kawasan Timur Indonesia (KTI) serta memberikan sumbangan devisa terbesar ke tiga sub sektor perkebunan setelah karet dan minyak sawit dengan nilai sebesar US \$ 701 juta. Perkebunan kakao di Indonesia mengalami perkembangan pesat dalam kurun waktu 20 tahun terakhir dan pada tahun 2002 areal perkebunan kakao Indonesia tercatat seluas 914.051 ha. Perkebunan kakao tersebut sebagian besar (87,4%) dikelola oleh rakyat dan selebihnya 6,0% dikelola perkebunan besar negara serta 6,7% perkebunan besar swasta. Jenis tanaman kakao yang diusahakan sebagian besar adalah jenis kakao curah dengan sentra produksi utama adalah Sulawesi Selatan, Sulawesi Tenggara dan Sulawesi Tengah. Di

samping itu juga diusahakan jenis kakao mulia oleh perkebunan besar negara di Jawa Timur dan Jawa Tengah (“Kementerian Perindustrian,” n.d.)

Kakao Indonesia, khususnya yang dihasilkan oleh rakyat, di pasar internasional masih dihargai paling rendah karena citranya yang kurang baik yakni didominasi oleh biji-biji tanpa fermentasi, biji-biji dengan kadar kotoran tinggi serta terkontaminasi serangga, jamur dan mitotoksin. Kelompok negara Asia diperkirakan akan terus mengalami peningkatan konsumsi seiring dengan pertumbuhan ekonomi di kawasan ini, sedikit saja kenaikan tingkat konsumsi di Asia, akan meningkatkan serangan produk kakao di Asia. Kapasitas produksi kakao di beberapa negara Asia Pasifik lain seperti Papua Nugini, Vietnam dan Filipina masih jauh di bawah Indonesia baik dalam hal luas areal maupun total produksi. Oleh karena itu dibanding Negara lain, Indonesia memiliki beberapa keunggulan dalam hal pengembangan kakao, antara lain ketersediaan lahan yang cukup luas, potensi pasar domestik yang besar dan sarana transportasi yang cukup baik.

Buah cokelat (kakao) bisa dipanen apabila perubahan warna kulit dan setelah fase pembuahan sampai menjadi buah dan matang \pm usia 5 bulan. Fase buah dari kecil hingga matang juga bervariasi sesuai jenis buah cokelat tersebut. Dalam penelitian ini buah cokelat yang dipilih sebagai objek adalah jenis Sulawesi 1 (S1) yang buahnya berwarna ungu dan saat matang berubah warna menjadi orange. Buah akan masak pada waktu 5,5 bulan (di dataran rendah) atau 6 bulan (di dataran tinggi) setelah

penyerbukan. Pemetikan buah harus dilakukan pada buah yang tepat masak karena kadar gula pada buah yang kurang matang rendah sehingga hasil fermentasi kurang baik, sebaliknya pada buah yang terlalu masak, biji seringkali telah berkecambah, pulp mengering dan aroma berkurang (Siregar dkk, 2003).



Gambar 1. Buah cokelat jenis Sulawesi 1

2) Visi Komputer

Visi komputer (*computer vision*) merupakan ilmu yang mempelajari bagaimana komputer dibuat sedemikian rupa sehingga dapat mengenali objek yang diamati. Visi komputer terdiri dari metode-metode untuk memperoleh, mengolah, dan menganalisis informasi dari satu citra atau kumpulan citra. Objek yang dimaksudkan tidak terbatas pada objek asli, tetapi bisa berupa citra dan video. Pengenalan dilakukan melalui sistem sensor, yang mana kamera merupakan sistem sensor yang paling umum digunakan dalam pengaplikasian visi komputer. Seperti halnya dengan

konsep kecerdasan buatan yang meniru konsep kecerdasan manusia, konsep visi komputer juga meniru bagaimana manusia melihat sehingga komputer dapat mengambil informasi yang terdapat dalam citra tersebut.

Visi komputer terdiri dari dua bagian, yaitu:

1. Pengolahan citra (*image processing*): merupakan bagian di mana kualitas citra ditingkatkan.
2. Pengenalan pola (*pattern recognition*): merupakan proses untuk mengelompokkan data masukan baik berupa data numerik maupun citra secara otomatis berdasarkan fitur yang dimiliki.

Aplikasi visi komputer sangat fleksibel sehingga dapat diterapkan pada berbagai bidang, seperti:

1. *Augmented reality*
2. Pendeteksian, klasifikasi dan hitung otomatis buah
3. Pengenalan dan pendeteksian wajah
4. Inspeksi kualitas pada industri
5. Sistem transportasi cerdas
6. Analisis citra medis
7. Keamanan
8. dan lain sebagainya.

3) Pengolahan Citra Digital

Pengolahan citra digital merupakan metode yang digunakan untuk mengolah citra pada komputer sehingga dapat menghasilkan gambar sesuai dengan yang dibutuhkan. Misalnya terdapat sebuah citra digital

berwarna dengan ukuran 1280×720 piksel. Dengan pengolahan citra digital, gambar tersebut dapat diubah ukurannya menjadi 640×360 piksel tanpa mengurangi kualitas gambar. Citra merupakan fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Secara matematis, fungsi ini dapat dilambangkan dengan $f(x,y)$, dimana (x,y) merupakan koordinat pada bidang dua dimensi tersebut, dan $f(x,y)$ merupakan intensitas cahaya pada titik (x,y) . Ukuran terkecil dalam citra digital biasa disebut *picture element* atau *pixel* (Syawaluddin, 2016).

Citra bergerak (*moving images*) adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri dari ratusan hingga ribuan *frame*. Meskipun sebuah citra kaya informasi, namun seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau, warnanya terlalu kontras, kurang tajam, dan kabur. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasikan (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik (Syawaluddin, 2016).

Di dalam mengolah sebuah citra, terdapat berbagai algoritma yang dapat diterapkan untuk menghasilkan keluaran yang lebih baik. Keluaran

yang baik akan mempengaruhi hasil dari proses yang akan dilakukan selanjutnya.

a. Citra Digital

Citra merupakan kombinasi antara titik, garis, bidang, dan warna yang mewakili suatu objek atau benda. Sedangkan citra digital merupakan keluaran yang dihasilkan melalui perangkat pencitraan digital seperti kamera dan dapat disimpan oleh komputer digital.

Untuk menerjemahkan citra menjadi angka-angka yang dapat dipahami oleh komputer, maka citra dibagi menjadi bagian-bagian kecil yang disebut sebagai piksel.

Dalam setiap piksel, perangkat pencitraan merekam sebuah angka ataupun sekumpulan angka yang merepresentasikan beberapa properti pada piksel yang dituju, seperti intensitas cahaya maupun warna. Angka-angka ini diatur dalam sebuah baris-baris dan kolom-kolom *array* yang merujuk pada posisi vertikal dan horizontal dari piksel-piksel yang ada dalam citra (Syawaluddin, 2016). Berikut adalah persamaan 2.1 yang menunjukkan representasi citra jika dibuat dalam bentuk matriks.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & \cdots & \cdots & f(1,M-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix} \quad 2.1$$

Menurut Darmawan dan Saptiani (2010), citra digital mengandung elemen-elemen dasar yaitu:

- a. Kecerahan (*Brightness*) merupakan intensitas cahaya rata-rata dari suatu area yang melingkupinya.
- b. Kontras (*Contrast*) merupakan sebaran terang (lightness) dan gelap (darkness) di dalam sebuah citra. Citra dengan kontras rendah komposisi citranya sebagian besar terang atau sebagian besar gelap. Citra dengan kontras yang baik, komposisi gelap dan terangnya, tersebar merata.
- c. Kontur (*Contour*) merupakan keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel tetangga, sehingga dapat dideteksi tepi objek di dalam citra.
- d. Warna (*Color*) merupakan persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya () yang dipantulkan oleh objek. Warna-warna yang dapat ditangkap oleh mata manusia merupakan kombinasi cahaya dengan panjang berbeda. Kombinasi yang memberikan rentang warna paling lebar adalah red (*R*), green (*G*) dan blue (*B*).
- e. Bentuk (*Shape*) merupakan properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk visual manusia. Umumnya citra yang dibentuk oleh manusia merupakan 2D, sedangkan objek yang dilihat adalah 3D.
- f. Tekstur (*Texture*) merupakan distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga.

b. Citra Warna

Pada citra warna, setiap piksel yang terdapat di dalamnya mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Biasanya warna-warna dasar ini disebut sebagai *channel*. Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 \times 2^8 \times 2^8 = 2^{24} = 16$ juta warna lebih. Itu sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

Penyimpanan *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* yang terdiri atas 256 gradasi warna hanya diwakili oleh 1 *byte*. Sedangkan 1 piksel citra *true color* diwakili oleh 3 *byte*, di mana masing-masing byte merepresentasikan tiap *R*, *G*, dan *B*. Gambar 2 berikut merupakan contoh representasi matriks dengan *channel R*, *G*, dan *B*.

					165	187	209	58	7	
					14	125	233	201	98	159
253	144	120	251	41	147	204				
67	100	32	241	23	165	30				
209	118	124	27	59	201	79				
210	236	105	169	19	218	156				
35	178	199	197	4	14	218				
115	104	34	111	19	196					
32	69	231	203	74						

Gambar 2. Contoh matriks citra *RGB* (Courtney, 2001).

c. Citra *Grayscale*

Sesuai dengan namanya, jenis citra ini memiliki gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan 255 menyatakan putih. Citra RGB dengan matriks penyusun citra yang sebelumnya 3 matriks akan berubah menjadi 1 matriks saja.

Citra *grayscale* merupakan citra yang nilai intensitas pikselnya didasarkan pada derajat keabuan. Pada citra *grayscale* 8-bit, derajat warna hitam sampai dengan putih dibagi ke dalam 256 derajat keabuan dimana warna hitam sempurna direpresentasikan dengan nilai 0 dan putih sempurna dengan nilai 255. Citra RGB dapat dikonversi menjadi citra *grayscale* sehingga dihasilkan hanya satu kanal warna.

4) Thresholding

Thresholding merupakan salah satu teknik dasar dalam melakukan segmentasi citra. *Thresholding* digunakan untuk mengonversi citra *grayscale* menjadi citra biner. Citra biner hanya terdiri dari dua intensitas warna yaitu hitam yang memiliki representasi nilai 0 dan putih yang memiliki representasi nilai 1. Sehingga jenis citra ini hanya membutuhkan 1 bit memori untuk menyimpan kedua warna ini.

Thresholding secara umum digunakan untuk memisahkan beberapa objek target dari latar belakangnya. Objek target biasanya direpresentasikan dengan warna putih. Adapun algoritma yang digunakan pada global *thresholding* ditunjukkan pada persamaan 2.2 berikut:

Untuk semua piksel (i, j) :

$$f'(i, j) = \begin{cases} 1 & \text{if } f(i, j) \geq T \\ 0 & \text{if } f(i, j) < T \end{cases} \quad 2.2$$

Berdasarkan persamaan 2.2, jika nilai piksel pada koordinat (i, j) lebih besar atau sama dengan nilai ambang, T , yang telah ditentukan sebelumnya, maka nilai tersebut akan diubah menjadi 1 dengan warna putih. Sedangkan jika nilai piksel pada koordinat (i, j) lebih kecil dibandingkan dengan nilai ambang, T , maka nilai tersebut akan diubah menjadi 0 dengan warna hitam.

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik yang berisi nilai-nilai *real* maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Suatu citra dapat didefinisikan sebagai fungsi $f(x, y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan *amplitude* f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Nilai pada suatu irisan antara baris dan kolom (pada posisi x, y) disebut dengan *picture element*, *image element*, *pels*, atau *pixel*. Istilah terakhir (*pixel*) adalah istilah yang paling sering digunakan pada citra digital (Sutoyo et al., 2009).

- 5) **Open CV** merupakan singkatan dari Open Computer Vision. OpenCV dapat digunakan secara gratis tanpa perlu membayar untuk mendapatkan lisensi tertentu. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. OpenCv mempunyai API (*Aplication Programming Interface*) untuk *high level* maupun *low level*, terdapat fungsi2 yang siap pakai, baik untuk loading, saving, akuisisi gambar maupun video. Pustaka memiliki lebih dari 2500 algoritma yang telah dioptimisasi termasuk algoritma klasik maupun algoritma-algoritma yang sudah masuk ke tahap *state of the art* untuk *computer vision* dan algoritma untuk *machine learning*. Algoritma-algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, objek, mengklasifikasikan gerakan manusia dalam video, mengikuti pergerakan kamera, mengikuti objek yang bergerak, mengekstrak model 3D dari suatu objek, menggabungkan citra untuk mendapatkan citra yang beresolusi tinggi, mencari gambar yang mirip dalam database, menghilangkan efek mata merah dari citra hasil tangkapan kamera flash dan masih banyak lagi. Dewasa ini telah lebih dari 47,000 pengguna dari pustaka ini dan total unduhan sudah melebihi 14 juta kali. Pustaka ini memiliki interface untuk C++, C, Python, Java dan MATLAB dan juga mendukung sistem operasi windows, Linux, Android dan Mac OS. OpenCV sangatlah handal digunakan pada realtime visionapplication. *OpenCV* terbaru saat ini sudah memasuki versi 3.4.0 (opencv.org)

6) ***Convolutional Neural Network (CNN)***

Pada tahun 1960an, ketika Hubel dan Wiesel meneliti *neuron* yang digunakan untuk orientasi *sensitif* lokal-selektif dalam sistem visual kucing, mereka menemukan bahwa struktur jaringan khusus dapat secara efektif mengurangi kompleksitas jaringan saraf umpan balik dan kemudian mengusulkan *Convolutional Neural Network (CNN)*. *CNN* adalah algoritma pengenalan yang efisien dan banyak digunakan dalam pengenalan pola dan pengolahan citra. *CNN* memiliki banyak fitur seperti struktur sederhana, parameter pelatihan yang kurang dan kemampuan beradaptasi. *CNN* telah menjadi topik hangat dalam analisis suara dan pengenalan gambar. Bobot struktur jaringan bersama membuatnya lebih mirip dengan jaringan saraf biologis. Hal ini mengurangi kompleksitas model jaringan dan jumlah bobot (Liu, 2015).

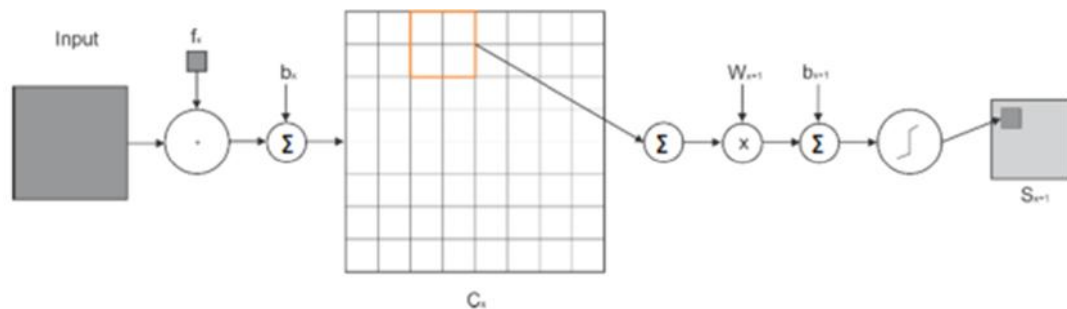
Umumnya, struktur *CNN* mencakup dua lapisan. Lapisan pertama adalah ekstraksi fitur, dimana masukan setiap *neuron* terhubung ke bidang reseptif lokal dari lapisan sebelumnya, dan mengekstrak fitur lokal. Setelah fitur lokal diekstraksi, hubungan posisi antara itu dan fitur lainnya juga akan ditentukan. Lapisan yang lainnya adalah lapisan fitur *map*, dimana setiap lapisan komputasi jaringan terdiri dari sejumlah fitur *map*. Struktur fitur *map* menggunakan fungsi *sigmoid* sebagai fungsi aktivasi dari jaringan konvolusi, yang membuat peta fitur memiliki pergeseran *invariance*. Selain itu, karena *neuron* di setiap pemetaan sama berbagi bobot, jumlah parameter bebas dari jaringan berkurang. Setiap lapisan konvolusi di

jaringan saraf konvolusi diikuti oleh lapisan komputasi yang digunakan untuk menghitung rata-rata lokal dan ekstrak kedua. Dua lapisan ekstraksi fitur unik ini mengurangi resolusi (Liu, 2015).

CNN terutama digunakan untuk mengidentifikasi perpindahan, zoom dan bentuk inversi grafis dua dimensi yang distortif. Karena lapisan deteksi fitur *CNN* belajar dengan data pelatihan, *CNN* menghindari ekstraksi fitur eksplisit dan secara implisit belajar dari data pelatihan saat kita menggunakan *CNN*. Selanjutnya, *neuron* di bidang peta fitur yang sama memiliki bobot yang sama, sehingga jaringan bisa belajar secara bersamaan. Ini adalah keuntungan besar dari jaringan konvolusi sehubungan dengan jaringan *neuron* yang terhubung satu sama lain. Karena struktur khusus dari bobot berbagi lokal *CNN* membuatnya memiliki keunggulan unik dalam pengenalan suara dan pemrosesan gambar. Tata letaknya lebih dekat ke jaringan saraf sebenarnya. Bobot bersama mengurangi kompleksitas jaringan. Secara khusus gambar vektor input multi-dimensi dapat langsung masuk ke jaringan, yang menghindari kompleksitas rekonstruksi data dalam ekstraksi fitur dan proses klasifikasi (Liu, 2015).

Algoritma *CNN* memiliki dua proses yaitu: konvolusi dan sampling. Proses konvolusi menggunakan filter yang dapat dilatih F_x , dekonvolusi gambar masukan (tahap pertama adalah gambar input, masukan dari

konvolusi setelah adalah fitur gambar masing-masing layer, yaitu feature map), lalu tambahkan bias b_x , kita bisa mendapatkan lapisan konvolusi C_x .



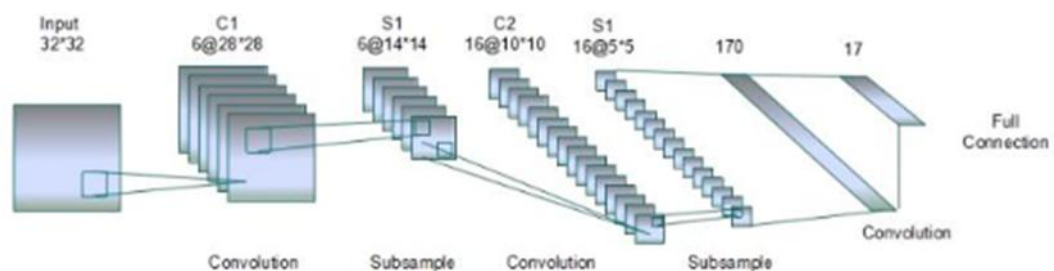
Gambar 3. Proses Utama pada CNN

Proses sampling menggunakan n piksel setiap lingkungan melalui langkah penyatuan, menjadi piksel, dan kemudian dengan bobot skalar $Wx + 1$

Teknologi kunci CNN adalah field reseptif lokal, pembagian bobot, sub sampling berdasarkan waktu atau ruang, sehingga bisa mengekstrak fitur dan mengurangi ukuran parameter pelatihan. Keuntungan algoritma CNN adalah untuk menghindari ekstraksi fitur eksplisit, dan secara implisit untuk belajar dari data pelatihan. Bobot neuron yang sama pada permukaan pemetaan fitur, sehingga jaringan dapat dipelajari secara paralel, mengurangi kompleksitas jaringan. Mengadopsi struktur sub sampling berdasarkan waktu atau ruang, dapat mencapai tingkat ketahanan tertentu, skala dan deformasi perpindahan; Input informasi dan topologi jaringan bisa menjadi pertandingan yang sangat baik, Ini memiliki keunggulan unik dalam pengenalan suara dan pengolahan gambar (Liu, 2015).

Algoritma CNN membutuhkan pengalaman dalam desain arsitektur, dan perlu melakukan debug tanpa henti dalam aplikasi praktis, agar bisa mendapatkan arsitektur aplikasi CNN yang paling sesuai. Berdasarkan

contoh pada gambar abu-abu dengan input 96×96 . Pada tahap preprocess, mengubahnya menjadi 32×32 dari ukuran gambar. Desain kedalaman lapisan 7 model konvolusi: lapisan input, lapisan konvolusi C1, lapisan sub sampling S1, lapisan konvolusi C2, lapisan sampling S2, lapisan tersembunyi H dan lapisan keluaran F (Liu,2015).

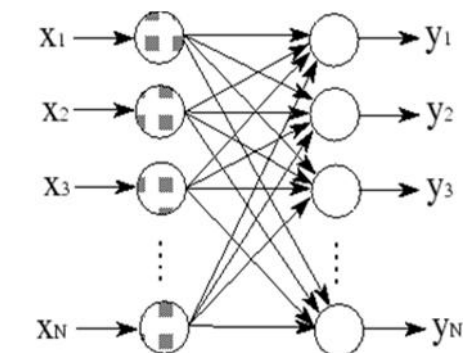


Gambar 4. Arsitektur CNN

Berdasarkan arsitekturnya, jaringan saraf tiruan terbagi atas dua, yaitu Single Layer Network dan Multi Layer Network.

1. Single Layer Network

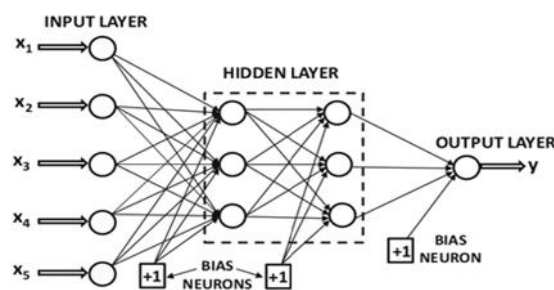
Arsitektur jaringan ini hanya terdiri dari satu input layer dan satu output layer. Setiap neuron yang terdapat di dalam input layer selalu terhubung dengan setiap neuron yang terdapat pada output layer. Jaringan ini hanya menerima masukan kemudian secara langsung akan mengolahnya menjadi suatu keluaran. Gambar 5 menunjukkan arsitektur dari single layer network.



Gambar 5. *Single Layer Network*

2. Multi Layer Network

Arsitektur jaringan ini terdiri dari lebih dari satu lapisan komputasional. Berbeda dari single layer network yang hanya terdiri dari input layer dan keluaran, pada multi layer network terdapat lapisan komputasi tambahan di antara input layer dan output layer. Arsitektur dari jaringan saraf multi layer secara spesifik dinyatakan sebagai feed forward network, karena nilai pada neuron pada suatu lapisan berturut-turut dimasukkan ke neuron pada lapisan berikutnya, mulai dari input layer hingga pada output layer. Gambar 6 menunjukkan arsitektur multi layer network.



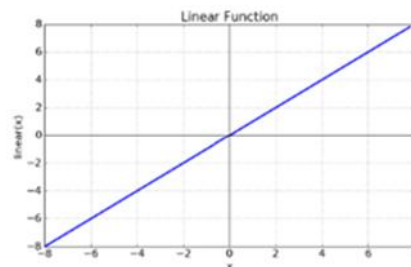
Gambar 6. *Multilayer Network*

Pada *Neural Network* dikenal istilah *Activation function* yaitu sebuah node yang ditambahkan ke akhir output dari setiap jaringan

syaraf. Hal ini juga dikenal sebagai Transfer Function. Hal ini juga dapat dilampirkan di antara dua Neural Networks. Activation function digunakan untuk menentukan output jaringan syaraf tiruan. Activation function memetakan nilai yang dihasilkan di antara 0 sampai 1 atau -1 sampai 1 dan lainlain (tergantung pada fungsinya). Activation function pada dasarnya dapat dibagi menjadi 2 tipe yaitu fungsi aktivasi linier dan fungsi aktivasi non linier. (Sharma,2017).

a. Fungsi Aktivasi *Linear*

Seperti yang dapat dilihat pada Gambar 7 fungsinya adalah garis atau linier. Oleh karena itu, output dari fungsi tidak akan dibatasi antara suatu rentang.

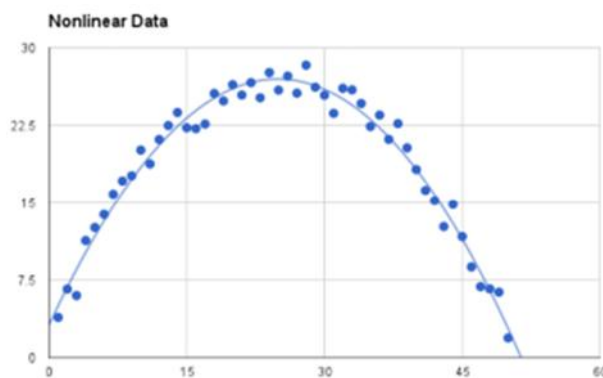


Gambar 7. Fungsi Linear

Fungsi aktivasi pada linear function memiliki persamaan : $f(x) = x$ dan rentang : -8 sampai 8. Fungsi aktivasi linier ini tidak membantu dengan kompleksitas atau berbagai parameter data biasa yang diumpankan ke jaringan syaraf tiruan. (Sharma, 2017).

b. Fungsi Aktivasi *Non Linear*

Fungsi Aktivasi *Non linear* adalah fungsi aktivasi yang paling banyak digunakan. Fungsi aktivasi *non linier* memudahkan model untuk menggeneralisasi atau menyesuaikan dengan berbagai data dan untuk membedakan antara *output*. Grafik *nonlinier* dapat dilihat seperti Gambar 8 berikut:

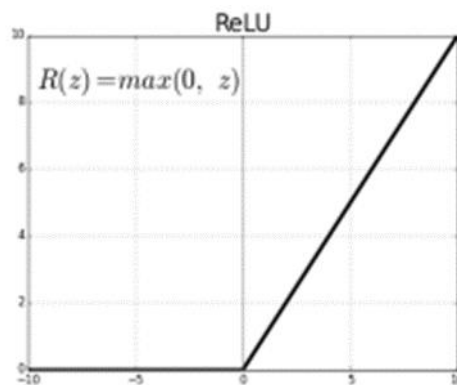


Gambar 8. Fungsi *Non Linear*

Terminologi utama yang perlu dipahami untuk fungsi nonlinier adalah Derivatif atau Diferensial dan Monotonik. Fungsi Derivatif atau Diferensial: Perubahan sumbu y w.r.t. Perubahan x-axis. Hal ini juga dikenal sebagai kemiringan dan fungsi Monotonik merupakan sebuah fungsi yang bervariasi sedemikian rupa sehingga tidak pernah berkurang atau tidak pernah meningkat. Terdapat beberapa fungsi aktivasi diantaranya adalah Sigmoid, Tanh, ReLU, dan Leaky ReLU.

c. ReLU (*Rectrified Linear Unit*)

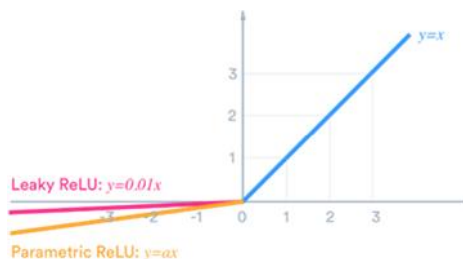
ReLU adalah fungsi aktivasi yang paling banyak digunakan saat ini. Karena, ini digunakan di hampir semua jaringan saraf konvolusi atau pembelajaran yang dalam. Fungsi ReLU diilustrasikan seperti pada Gambar 9



Gambar 9. *Rectrified Linear Unit*

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 grafik yang dihasilkan tidak dapat memetakan nilai negatif secara tepat (Sharma, 2019).

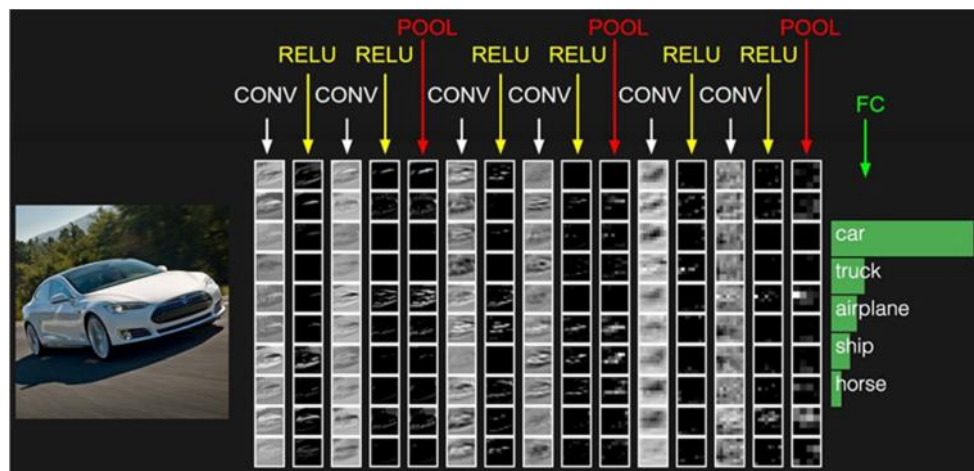
Selain itu, dengan mengubah data negatif menjadi 0, maka neuron-neuron yang ada akan mati, dan menjadi inactive apapun jenis input yang diberikan. Jika terdapat banyak neuron di dalam neural network maka performa dari sistem akan terganggu. Untuk mengatasi hal ini maka fungsi aktivasi turunan dari ReLU yaitu Leaky ReLU digunakan. Leaky ReLU mengubah nilai negatif dengan mengalikan dengan suatu faktor sehingga nilai negatif tidak menjadi 0. Gambar 10. menunjukkan grafik dari Leaky ReLU (Sharma, 2019).



Gambar 10. *Leaky ReLu* (medium.com)

d. **Fully-connected (FC) Layer**

Fully Connected 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. Tabel input yang merupakan hasil dari layer sebelumnya sesuai dengan feature map untuk fitur yang diberikan: nilai-nilai yang tinggi menunjukkan lokasi fitur pada gambar. Jika lokasi fitur pada titik tertentu dalam gambar adalah karakteristik dari kelas tertentu, maka diberikan nilai bobot yang sama dengan tabel.



Gambar 11. Ilustrasi proses lengkap CNN

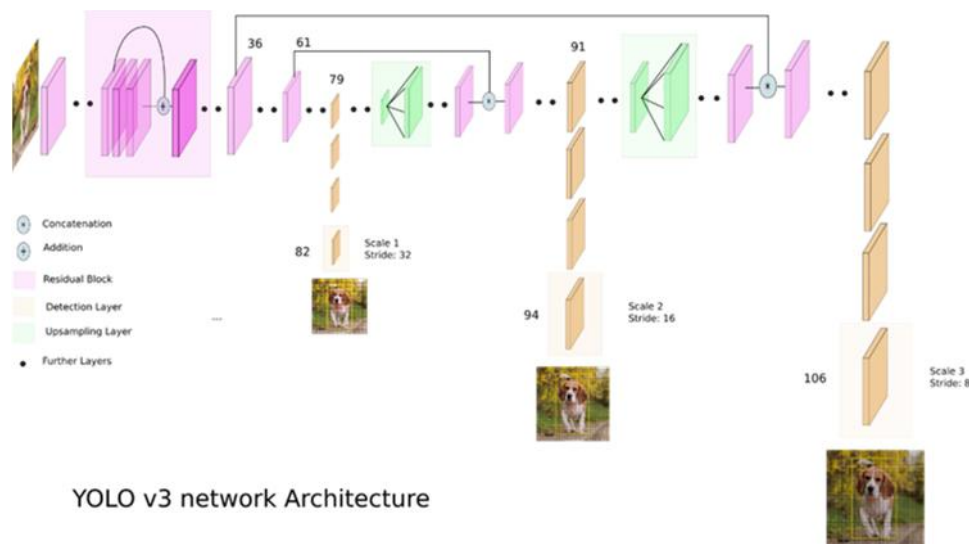
7) *You Only Look Once* (YOLO)

Berdasarkan penelitian Liu et al., terdapat tiga metode *deep learning* yang memiliki performansi tertinggi, yaitu *Faster RCNN*, *You Only Look Once* (YOLO), dan *Single Shot Multibox Detector* (SSD). Melalui eksperimen, penelitian tersebut menguji waktu dan akurasi deteksi menggunakan dataset PASCAL VOC, COCO, dan ILSVRC dengan ukuran masukan yang bervariasi. Ketiga metode tersebut dapat mendeteksi objek secara *realtime*, namun memiliki *mAP*, *FPS*, dan jumlah *bounding box* hasil deteksi untuk objek yang berbeda. *Mean Average Precision* (*mAP*) merupakan akurasi rata-rata presisi ukuran *bounding box* dengan ukuran objek pada citra.

YOLO menggunakan *convolutional neural networks* untuk mendeteksi objek. YOLO adalah salah satu algoritma deteksi objek yang tercepat saat ini dan merupakan pilihan yang baik untuk melakukan deteksi objek secara *realtime*, tanpa harus kehilangan banyak akurasi.

Jika dibandingkan dengan algoritma *recognition* (pengenalan), algoritma pendeteksi objek tidak hanya memprediksi jenis kelas objek akan tetapi juga mendeteksi lokasi dari objek tersebut. Jadi, algoritma ini tidak hanya mengklasifikasi sebuah input citra ke dalam suatu kategori, tetapi juga bisa mendeteksi berbagai objek yang berada di dalam citra. Algoritma ini mengaplikasikan *single neural network* ke sebuah citra secara keseluruhan. Artinya, *network* tersebut membagi citra ke berbagai *regions* dan memprediksi *bounding boxes* dan probabilitas tiap *region*. Setiap *bounding box* yang ada dihitung berdasarkan probabilitas prediksinya.

YOLOv3 adalah algoritma deteksi objek generasi ketiga dari YOLO. Pada versi ini, YOLO mengalami improvisasi dengan berbagai penyesuaian dan memiliki kemampuan untuk mendeteksi objek yang lebih kecil dengan lebih baik. Arsitektur dari YOLOv3 dapat dilihat Gambar 12. berikut:



Gambar 12. Arsitektur *Network* YOLOv3 (medium.com)

Pada paper YOLOv3, penulis memperkenalkan arsitektur baru yang lebih dalam yang diberi nama *Darknet-53*. Seperti yang terlihat pada namanya, arsitektur ini memiliki 53 *layer* konvolusi, dimana setiap *layer* berisi *batch normalization* dan *ReLU activation* (Redmon and Farhadi, n.d.)

Algoritma YOLOv3 sudah melengkapi beberapa kekurangan dalam sistem pendeteksian YOLOv2. YOLOv2 menggunakan *custom deep architecture darknet-19*, sebuah *neural network* dengan 19 *layer* yang ditambah dengan 11 *layer* lainnya untuk proses deteksi objek. Dengan jumlah *layer* sebanyak 30, YOLOv2 terkadang kesulitan saat mendeteksi objek-objek yang kecil. Hal ini terjadi karena dalam YOLOv2, *input* citra hanya di *downsampling* secara kasar. Untuk mengatasi hal ini, YOLOv2 menggunakan *identity mapping*, *fitur map* gabungan dari *layer* sebelumnya untuk mendapatkan *low level features*. YOLOv2 masih kekurangan berbagai element penting yang ada pada berbagai algoritma deteksi objek lain yang sudah ada saat ini, antara lain *Residual Blocks (ResNet)*, *skip connections*, dan *upsampling*. Algoritma YOLOv3 sudah menerapkan semuanya dalam sistemnya. Secara lengkap, model *Layer Network Darknet-53* dapat dilihat pada Gambar 13. di bawah ini:

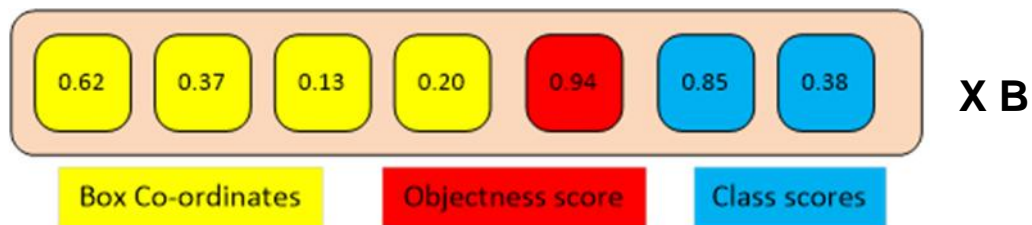
	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	128×128
	Convolutional	64	3×3	
	Residual			
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	64×64
	Convolutional	128	3×3	
	Residual			
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	32×32
	Convolutional	256	3×3	
	Residual			
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	16×16
	Convolutional	512	3×3	
	Residual			
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	8×8
	Convolutional	1024	3×3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Darknet-53 model

Gambar 13. Model *Layer Network Darknet-53* (medium.com)

Arsitektur terbaru yang digunakan pada YOLOv3 memiliki *residual skip connections*, dan *upsampling*. Fitur yang paling menonjol dari YOLOv3 adalah algoritma ini melakukan pendeteksian objek pada 3 skala yang berbeda. YOLO merupakan sebuah *fully convolutional network* dimana *output* nya digenerate dengan cara mengaplikasikan *filter* berupa kernel ke dalam *fitur map*. Dalam YOLOv3, deteksi dilakukan dengan cara mengaplikasikan kernel pendeteksi berukuran 1×1 pada *fitur map* pada 3 ukuran yang berbeda dan 3 lokasi yang berbeda pada *network*.

Ukuran dari kernel pendeteksi adalah $1 \times 1 \times (B \times (5 + C))$. Disini B adalah jumlah dari *bounding boxes* di dalam sebuah *cell* pada *fitur map* yang dapat diprediksi, “5” merupakan 4 atribut dari *bounding box* dan 1 *object confidence*, sementara C adalah jumlah kelas objek yang dilatih. *Fitur map* yang diperoleh dari kernel ini memiliki tinggi dan lebar yang identik dengan *fitur map* sebelumnya, dan memiliki atribut deteksi seperti yang disebutkan di atas.



Gambar 14. Atribut *bounding box output* dari *FC layer*

Sebagai contoh, *input* citra memiliki ukuran 416×416 piksel. YOLOv3 melakukan proses deteksi objek pada 3 skala berbeda dengan cara *down sampling* dimensi dari *input* citra dengan faktor 32, 16, dan 8 secara berurutan. Proses deteksi pertama dilakukan pada *layer* ke-82. Untuk 81 *layer* pertama, citra akan dilakukan *down sampling* oleh *network*, dengan *stride* 32. Jika *input* citra berukuran 416×416 , *fitur map* hasil dari *network* memiliki ukuran 13×13 . Proses pendeteksian pertama akan dilakukan dengan menggunakan *filter* pendeteksi berukuran 1×1 , sehingga memberikan *fitur map* dengan ukuran $13 \times 13 \times 255$.

Kemudian, *fitur map* dari *layer* 79 akan ditujukan ke beberapa *layer* konvolusi sebelum di *upsampling* ke dimensi berukuran 26×26 . *Fitur map* ini memiliki hubungan dengan *fitur map* pada *layer* 61. Kemudian *fitur map* yang ada akan diteruskan ke beberapa *convolutional layer* untuk digabungkan dengan *fitur* yang berasal dari *layer* yang ada lebih awal (*layer* 61). Kemudian, proses deteksi kedua dilakukan pada *layer* ke 94, dengan *fitur map* hasil deteksi berukuran $26 \times 26 \times 255$.

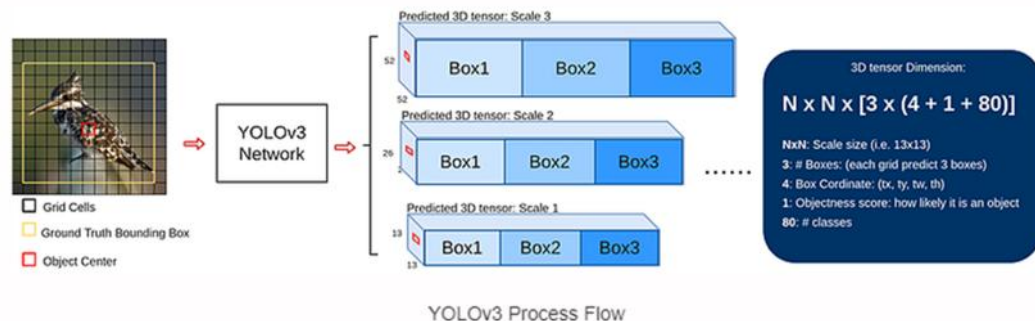
Prosedur yang sama kemudian dilakukan, dimana *fitur map* dari *layer* 91 akan diteruskan ke beberapa *convolutional layer* sebelum digabungkan dengan *fitur map* dari *layer* 36. Setelah itu proses deteksi ketiga atau *final* akan dilakukan pada *layer* ke 106, dengan *fitur* berukuran $52 \times 52 \times 255$.

Deteksi pada *layer* yang berbeda dapat mengatasi permasalahan dalam mendeteksi objek yang lebih kecil, permasalahan yang sering ditemukan pada YOLOv2. *Upsampling layer* yang berhubungan dengan *layer-layer* yang lebih awal memberikan *fitur-fitur* yang lebih detail pada *fitur map* yang membantu dalam pendeteksian objek yang lebih kecil.

Layer dengan ukuran 13×13 bertanggung jawab dalam pendeteksian objek berukuran besar, dimana *layer* 52×52 mendeteksi objek yang kecil, sementara *layer* 26×26 mendeteksi objek dengan ukuran sedang. Process

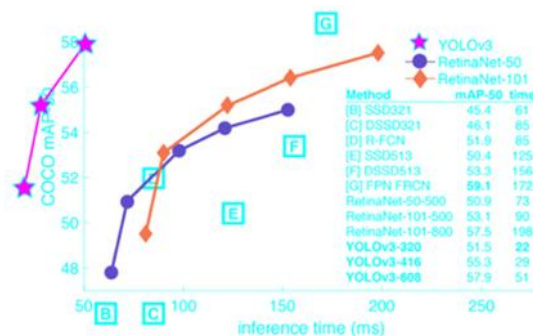
Flow dari proses deteksi objek dapat dilihat pada Gambar 15 berikut:

The algorithm



Gambar 15. Process flow dari YOLOv3 (Murugavel, 2018, p. 3)

Redmon et al. dalam tulisannya menyebutkan bahwa YOLOv3 sangat unggul dalam processing time karena proses deteksinya yang menggunakan pendekatan yang sangat berbeda sehingga memangkas waktu komputasi. Gambar 16 menunjukkan grafik perbandingan performa YOLO dengan algoritma lain :



Gambar 16. Grafik perbandingan performa YOLO (Redmon et al, 2015)

8) **Bounding Box**

Bounding box merupakan marquee khusus yang berupa kotak atau lingkaran yang memiliki titik-titik koordinat dan ukuran lebar serta tinggi (Komputer, 2008).

Untuk mencari *bounding box* gambar terlebih dahulu dibagi menggunakan kotak kotak pada seluruh bagian gambar, kotak ini

dinamakan *grid cell*. Setiap kotak dijadikan sebagai titik tengah dan diberikan 3 bentuk BBoxes. BBoxes ini digabungkan dengan *confidence*, penggabungan ini mempunyai kerja masing masing yaitu memprediksikan apakah ada objek atau tidak. Jika seandainya diprediksi adanya objek maka BBoxes akan diberikan nilai 1, jika tidak diberikan nilai 0. BBoxes yang bernilai 1 akan menghasilkan bounding box. Setiap *bounding box* memiliki 5 prediksi : x, y, w, h dan *confidence*. (x,y) sebagai titik koordinat tengah *grid cell*. Width dan height (w,h) sebagai batas ukuran keseluruhan gambar (objek). *Confidence* adalah bentuk nilai keyakinan dari prediksi (Redmon & Farhadi, 2016)

9) **Drone DJI Mavic Air**

Drone merupakan kendaraan tanpa awak berbentuk seperti pesawat atau *helicopter* dengan ukuran kecil yang dapat dioperasikan dengan *remote control*. *Drone* biasanya digunakan sebagai alat operasi bagi anggota militer, pertahanan dan melakukan aksi *intelligent*. *Drone* juga dimanfaatkan oleh perusahaan-perusahaan untuk mengantar barang. Kini dalam masyarakat umum *drone* digunakan untuk keperluan *fotografi* dan pembuatan film (Immersa Lab, 2018). Gambar 17 adalah *drone DJI Mavic Air* yang digunakan pada penelitian ini.



Gambar 17. Drone DJI Mavic Air

Drone DJI Mavic Air adalah *drone professional* keluaran DJI dan termasuk keluarga Mavic. Memiliki body yang mirip dengan DJI Spark dan fungsi lipat seperti DJI Mavic Pro.

Drone DJI Mavic Air dilengkapi dengan *GPS* dan *vision sistem* yang terdapat di bawah, depan dan belakang drone untuk sekaligus menjalankan fitur *sensing*.

Adapun baterai yang digunakan type LiCoO₂ 3s memiliki kapasitas 2375 mAh, dengan tegangan 11.55 Volt. Sedangkan kamera memiliki kualitas 2.7K hingga 4K.

Remote control drone DJI Mavic Air memiliki jangkauan transmisi hingga 4 km dalam kondisi tidak ada angin dan tidak ada interferensi sinyal, memiliki interface yang sederhana dan menggunakan *smartphone* dengan panjang maksimal 160 mm dengan ketebalan 6.5 – 8.5 mm sebagai layar RC. Kapasitas Baterai 2970 mAh, dan mendukung *USB port micro USB* (type-B), *USB-C* (Omah Drones, 2018)

Tabel 1. Spesifikasi umum dari drone DJI Mavic Air

Spesifikasi	Keterangan
Berat	430 g
Dimensi	Terlipat 168 x 83 x 49 mm (<i>p x l x t</i>) tidak terlipat 168 x 184 64 mm (<i>p x l x t</i>)
Diagonal	213 mm
Kecepatan maksimal	68.4 km/jam
Waktu terbang	21 menit (dengan kecepatan konstan 25 km/jam)
Tinggi terbang Maks.	10 km (dengan kecepatan konstan 25 km/jam)
Suhu operasi	0° – 40° Celcius
Memori Internal	8 GB
Memori Ekssternal	Class 10 Extreme
Posisi	Jarak
Depan	0.5 – 24 m
Belakang	0.5 – 20 m
Bawah	0.5 – 30 m

Tabel 2. Spesifikasi Kamera

Spesifikasi	Keterangan
ISO Range	Video 100 – 1600 Foto 100 – 3200
Ukuran Gambar	4 : 3 : 4056 x 3040 16 : 9 : 4056 x 2280
Bawah	0.5 – 30 m
Resolusi Video	4K Ultra HD: 3840x2160 24/25/30p 2.7K: 2720x1530 24/25/30/48/50/60p FHD: 1920x1080 24/25/30/48/50/60/120p HD: 1280x720 24/25/30/48/50/60/120p
Bitrate Video Maks.	100 Mbps
File System	FAT32
Format Gambar	JPEG
Format Video	MP4/MOV

B. PENELITIAN TERKAIT

Beberapa penelitian terkait deteksi objek telah dilakukan sebelumnya dengan beberapa metode dan algoritma sebagai berikut:

1. Sheryl et.al melakukan penelitian untuk menghitung ikan *fingerlings* menggunakan teknik *image processing* dan menyelidiki efektifitas *Convolutional Neural Netwok (CNN)* pada deteksi dan akurasi perhitungannya. Teknik yang di usulkan dengan mengetest pada empat ukuran yang berbeda. Nilai ambang batas ditetapkan untuk meningkatkan tingkat efisiensi deteksi ikan dan akurasi penghitungan. Metode eksperimental dan menangkap gambar dari jari itu dilakukan di Biro Perikanan dan Sumber Daya Perairan (BFAR) Laboratorium Pembenihan Ikan Nila dan Kesehatan Ikan di Kalimantan Barcenaga, Naujan, Oriental Mindoro, Filipina. 2400 gambar bibit nila digunakan dalam pelatihan dan 1.600 gambar ditahap pengujian (Lainez and Gonzales, 2019).
2. Suchet et.al pada tahun 2017 melakukan penelitian tentang deteksi buah dalam perkebunan menggunakan metode *Faster R-CNN*, Buah yang dideteksi beragam antara lain mangga , almond dan apel. Dataset diperbanyak untuk mendapatkan hasil yang maksimal. Disebutkan juga bahwa deteksi ini sangat membantu untuk pemetaan hasil panen. Dalam penelitian ini juga disajikan teknik augmentasi data dan tiap data buah diambil 1-100 gambar. Hasil dari penelitian ini

menunjukkan $F1 > 0.9$ dan menunjukkan akurasi yang lebih baik dari penelitian sebelumnya (Suchet et.al, 2017)

3. Xiao Changyi, et al pada tahun 2015 melakukan deteksi dan perhitungan buah apel menggunakan *BP Neural Network* dan *Circle Hough Transform* berdasarkan mode warna dan bentuk untuk manajemen produksi di kebun dengan menggunakan 178 gambar training dan 39 gambar untuk uji coba. Algoritma yang diusulkan menunjukkan kestabilan dalam mendeteksi dan menghitung apel dari gambar pohon apel dengan koefisien korelasi R^2 mencapai 0,985 (Changyi et al., n.d.).

C. STATE OF THE ART

State of The Art penelitian yang berhubungan dengan pendeteksian objek buah dengan menggunakan beberapa metode, antara lain metode *deep learning* dapat dilihat pada Tabel 3 berikut:

Tabel 3. State of The Art

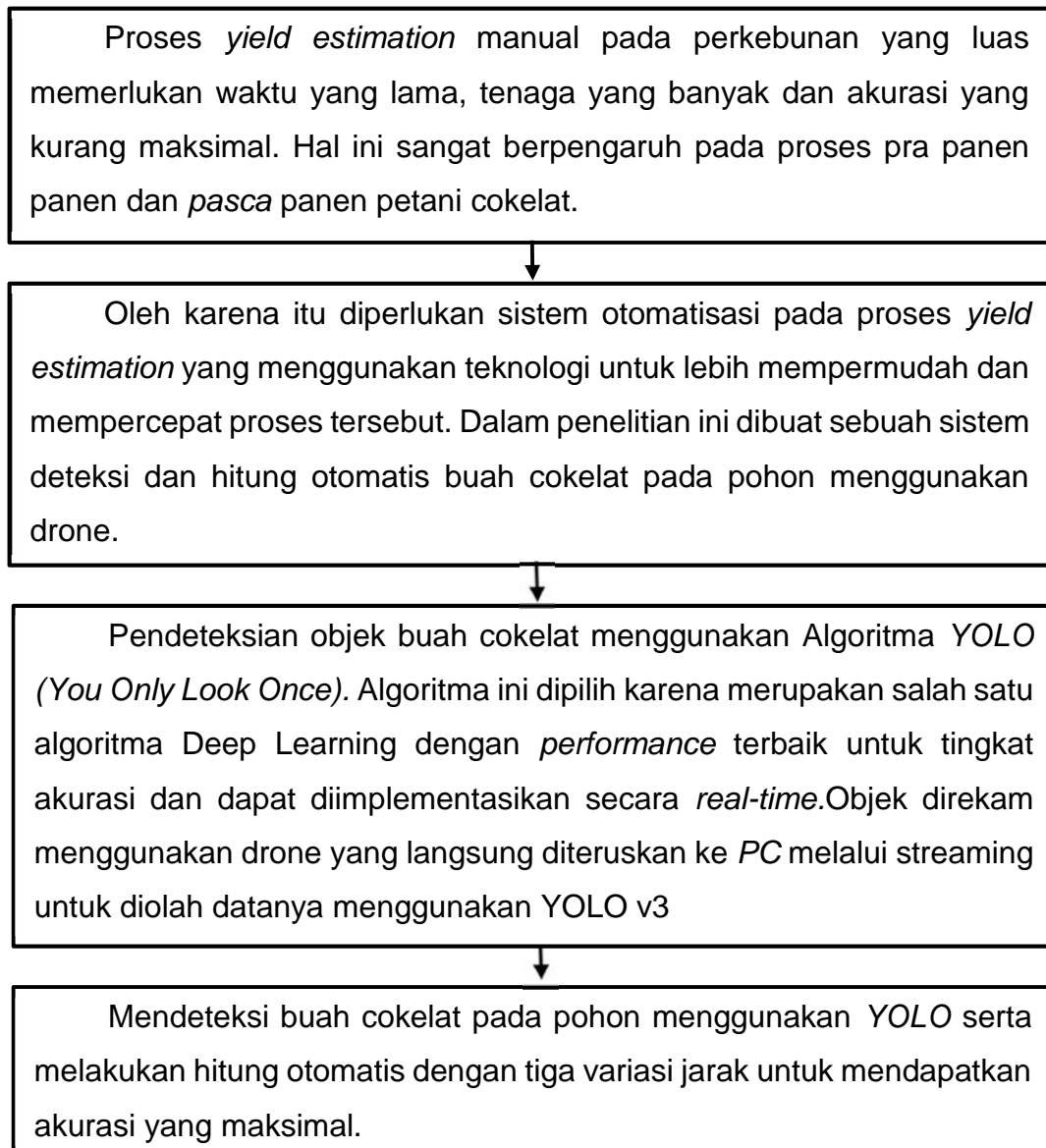
No	Judul	Penulis & Tahun	Penerbit	Metode	Hasil
1	<i>DeepFruits: A Fruit Detection System Using Deep Neural Networks</i>	Inkyu Sa *, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez and Chris McCool/ 2016	Sensor, MDPI journal	<i>Faster R-CNN</i>	$F1 \text{ score} = 0.838$

No	Judul	Penulis & Tahun	Penerbit	Metode	Hasil
2	<i>Deep Fruit Detection in Orchards</i>	Suchet Bargoti and James Underwood/ 2017	2017 IEEE International Conference on Robotics and Automation (ICRA) Singapore, May 29 - June 3, 2017	<i>Faster R-CNN</i>	$F1 > 0.9$
3	<i>A Real-Time Detection Framework for On-Tree Mango Based on SSD Network</i>	Qiaokang Liang, Yaonan Wang , Wei Zhu , Jianyong Long , Wei Sun , and Wanneng Wu/ 2018	Springer nature Zwitzerland	<i>Deep learning, CNN, Single Shot MultiBox Detector (SSD) network, SSD400 untuk deteksi realtime</i>	93%
4	<i>Apple detection from apple tree image based on BP neural network and Hough transform</i>	Xiao Changyi, Zheng Lihua , Li Minzan, Chen Yuan, Mai Chunyan/2015	Int J Agric & Biol Eng/ 2015	<i>BP Neural network, Edge Detection, Circle Hough Transform</i>	$R2 = 0,985$
5	<i>An yield estimation in citrus orchards via fruit detection and counting using image processing</i>	Ulzii-Orshikh Dorj, et al	Computers and Electronics in Agriculture / 2017	<ul style="list-style-type: none"> • RGB ke HSV • Watershed Algorithm untuk Deteksi dan Count 	94,24% $R2 = 0,93$

No	Judul	Penulis & Tahun	Penerbit	Metode	Hasil
6	<i>Automatic fruit count on coffee branches using computer vision</i>	<i>P.J. Ramos, et al / 2017</i>	<i>Computers and Electronics in Agriculture</i>	<i>Binary Image</i> •Canny edge detector untuk segmentasi •Fitzgibbon untuk klasifikasi •Bayes classifier, k-nearest neighbors classifier (KNN), Support Vector Machine (SVM)	88%

D. KERANGKA PIKIR

Kerangka penelitian ini didasarkan pada beberapa kendala saat dilakukan *yield estimation* pada perkebunan cokelat berskala besar:



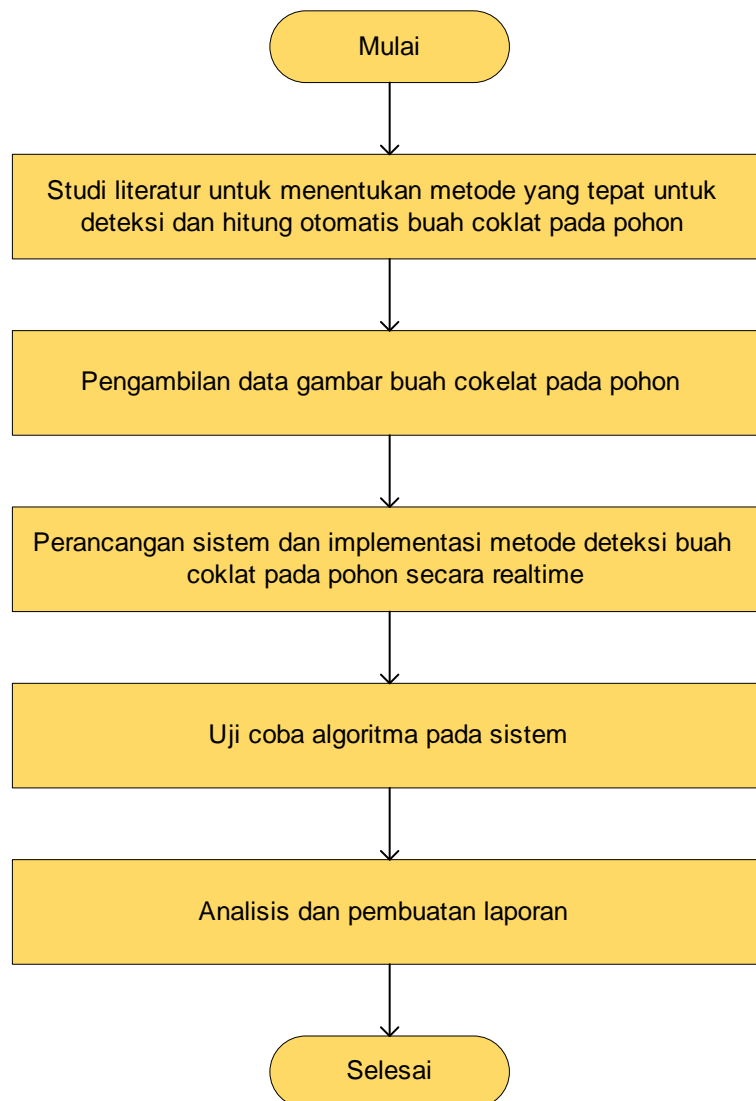
Gambar 18. Kerangka Pikir Penelitian

BAB III

METODE PENELITIAN

A. TAHAPAN PENELITIAN

Tahapan dalam penelitian ini dapat dilihat pada Gambar 19 berikut:



Gambar 19. Tahapan penelitian

Tahapan penelitian secara garis besar diuraikan sebagai berikut:

1. Studi literatur penelitian terkait metode deteksi dan hitung otomatis buah coklat diuraikan berdasarkan referensi yang bersumber dari jurnal atau *paper* penelitian, buku, artikel dan laporan penelitian. Selain itu, dilakukan juga pencarian dokumentasi terkait hasil penelitian sebelumnya yang dapat menjadi acuan dalam penelitian yang dilakukan.
2. Pengambilan data buah coklat pada pohon menggunakan *drone* yang kemudian melakukan koneksi ke android yang berfungsi sebagai *remote control* dan monitor *drone*. Android melakukan *streaming* yang kemudian terhubung ke PC agar data objek bisa diambil untuk diolah ke tahap selanjutnya.
3. Perancangan sistem dilakukan dengan membuat diagram sistem penelitian mulai dari proses input data, *preprocessing*, ekstraksi fitur pendeteksian dan hitung otomatis buah coklat pada pohonnya.
4. Uji coba algoritma pada sistem dilakukan dengan menguji algoritma yang digunakan dengan gambar buah coklat dan beberapa data negative yaitu objek selain buah coklat yang berada di sekitar perkebunan coklat untuk mengetahui akurasi dan efektifitas algoritma yang digunakan.
5. Setelah dilakukan uji coba algoritma atau metode yang digunakan maka dilakukan analisis. Tahapan ini menganalisis setiap paramater yang digunakan dengan menghitung tingkat akurasinya. Proses akhir

yang dilakukan yaitu pembuatan laporan publikasi dan laporan akhir magister.

B. WAKTU DAN LOKASI PENELITIAN

1) Waktu

Waktu penelitian dilaksanakan selama 7 (tujuh) bulan dimulai pada bulan Maret 2018 sampai Oktober 2019.

2) Lokasi

Penelitian dilaksanakan di perkebunan cokelat Desa Gantarang Keke, Kabupaten Bantaeng, BLPP Gowa dan Laboratorium Kecerdasan Buatan Departemen Teknik Informatika Universitas Hasanuddin.

C. JENIS PENELITIAN

Penelitian yang diusulkan merupakan penelitian yang bersifat eksperimental dan analisis sehingga dapat dilakukan dengan metode studi pustaka (*library research*), metode pengumpulan data lapangan (*field research*), perancangan sistem, dan analisis.

D. PENGAMBILAN DATA

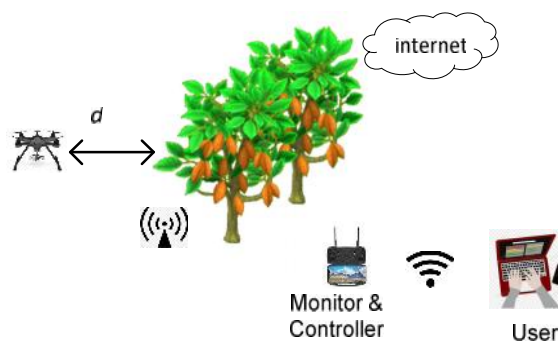
Data yang digunakan dalam penelitian ini diperoleh melalui pengambilan gambar buah cokelat pada pohon menggunakan *drone* yang akan terbang diantara baris pohon cokelat dan langsung mengambil data dengan tiga variasi jarak (d) yaitu $d=100$ cm, $d=150$ cm dan $d=200$ cm



Gambar 20. Pengambilan data di lapangan

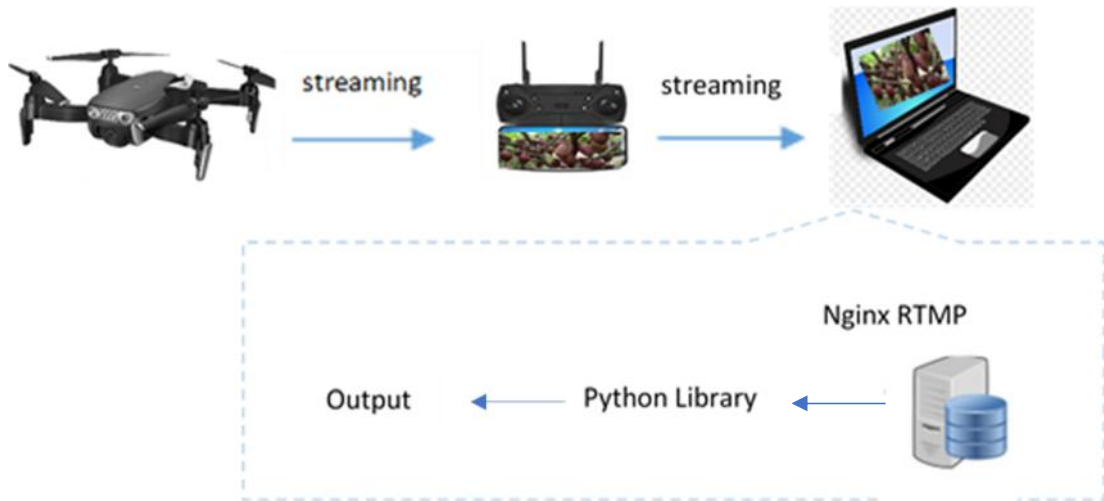
Data yang diperoleh kemudian dikirim ke PC melalui *streaming* untuk dapat diolah datanya hingga menghasilkan sistem pendeteksian buah coklat secara *realtime*.

Gambar 21 memperlihatkan desain sistem pengambilan data di lapangan:



Gambar 21. Desain Sistem Pengambilan Data

Gambar 22 berikut memperlihatkan alur data yang dikirim secara *realtime* dari *drone*:



Gambar 22. Alur data *streaming* dari *drone* ke *PC*

Untuk melakukan *streaming* dari *drone* ke *PC* langkah-langkahnya sebagai berikut :

1. Langkah pertama adalah melakukan instalasi software Nginx pada PC atau Laptop yang diperlukan untuk menjalankan protocol RTMP (*Real Time Messaging Protocol*)
2. Kemudian menghubungkan android yang sekaligus berfungsi sebagai *remote control* dengan PC atau laptop tempat data akan diolah, dengan melakukan setting pada IP address masing-masing perangkat dan melakukan koneksi dalam satu jaringan yang sama.
3. Fasilitas streaming pada drone DJI diatur ke menu custom, yaitu menu *streaming* melalui protokol RTMP://10.1.1.1/live/bird yang alamatnya disesuaikan dengan IP pada laptop. Selanjutnya *drone*

diarahkan pada objek untuk mengambil gambar, maka proses streaming dimulai.

4. Drone akan melakukan capture gambar setiap 5 detik dan selanjutnya gambar objek yang terdeteksi akan muncul pada layar PC.

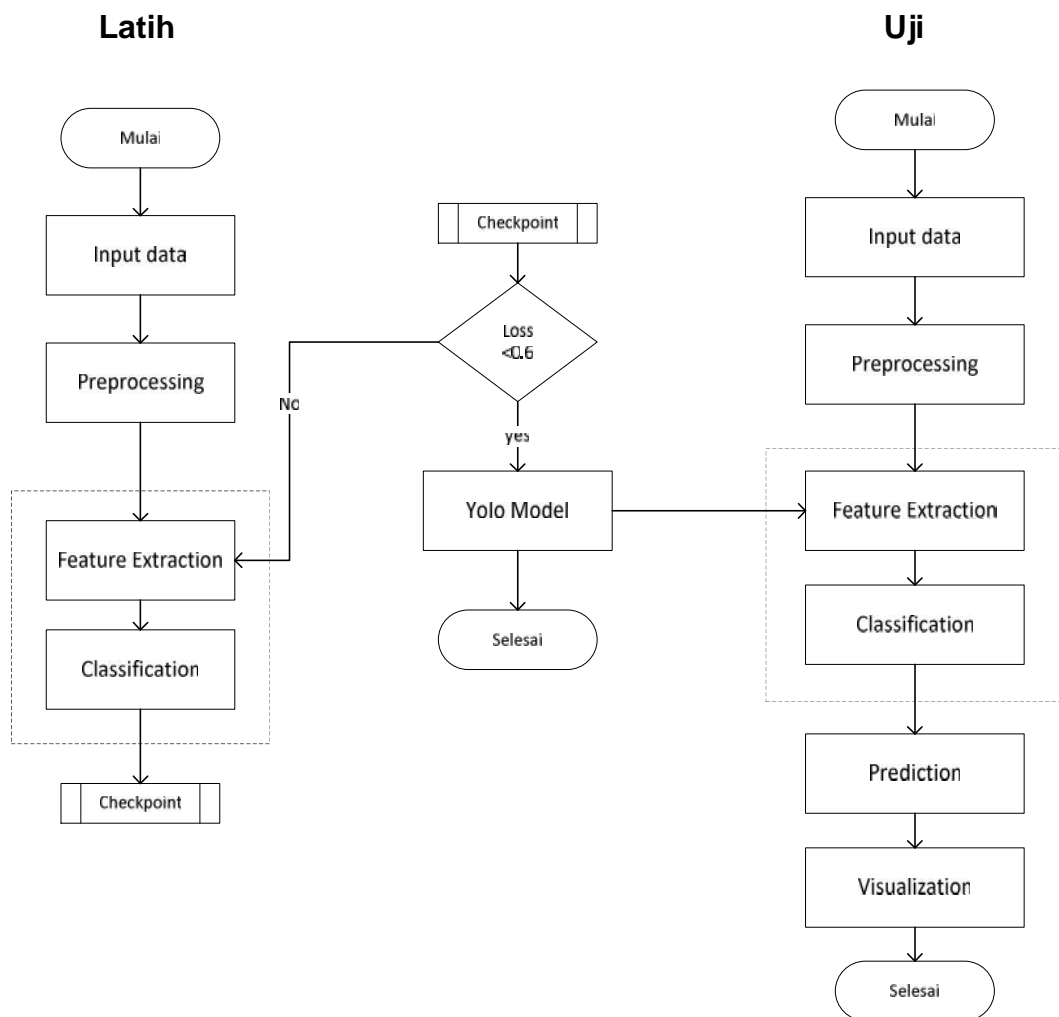
E. PERANCANGAN SISTEM

Sistem yang dibuat pada penelitian ini menggunakan algoritma *deep learning* YOLO generasi ketiga yang tahapan pelatihan datanya menggunakan *framework Darknet-53* yaitu sebuah *framework* yang prinsip kerjanya berdasarkan *neural network* dan mempunyai 30 *layer* seperti penjelasan pada Bab 2.

Sistem ini secara umum dibagi menjadi dua bagian, yaitu pelatihan dan pengujian seperti yang diperlihatkan pada gambar 23. Data latih yang digunakan sebanyak 180 gambar dan data uji sebanyak 45 gambar. Dalam proses pelatihan, fitur yang telah diekstrak dari data latih menggunakan *layer convolutional neural network* akan dijadikan input ke dalam *fully connected layer* yang akan menghasilkan klasifikasi objek : coklat dan bukan coklat. Hasil klasifikasi dari *fully connected layer* inilah yang akan digunakan untuk menentukan hasil deteksi objek selanjutnya. Model yang sudah dilatih akan disimpan untuk kemudian digunakan untuk proses pengujian. Model *deep learning* yang sudah dilatih inilah yang akan digunakan untuk proses pengujian sistem nantinya. Tahap berikutnya dilakukan training pada data latih menggunakan *Darknet* untuk memperoleh

model *YOLO* yang akan digunakan untuk proses pendeteksian objek nantinya.

Rancangan sistem *training* dan uji data pada penelitian ini dapat dilihat pada Gambar 23 berikut:



Gambar 23. Rancangan sistem

1) Input Data

Tahap awal dimulai dengan menginput data yang akan dijadikan masukan untuk melakukan *training*. Data yang disiapkan adalah data gambar berukuran 3840 x 2160. Data tersebut diambil dengan menggunakan kamera drone. Drone terbang diantara baris pohon coklat dan mengambil gambar objek dengan 3 variasi jarak (d), yaitu $d = 100$ cm , $d = 150$ cm dan $d = 200$ cm untuk masing-masing pohon coklat.

2) Preprocessing Data Training

- a. Data latih yang berjumlah 180 gambar terlebih dahulu akan melalui tahap preprocessing yaitu tahap pelabelan. Contoh data latih dengan tiga variasi jarak seperti Gambar 24 berikut:



(a)

(b)

(c)

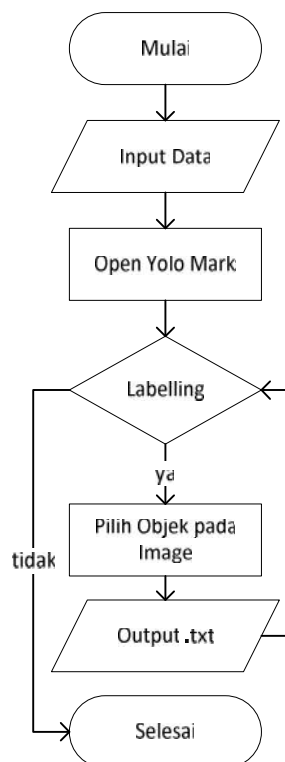
Gambar 24. Contoh gambar masukan untuk 3 variasi jarak (a) $d=100$ cm, (b) $d=150$ cm, (c) $d=200$ cm

Gambar yang akan digunakan sebagai data latih sebanyak 180 gambar berukuran 3840 x 2160 dengan 3 variasi jarak dan untuk masing-masing d disiapkan 60 data latih. Pada proses training ini

data yang diinput tidak memerlukan *resize* karena gambar otomatis akan berubah ke ukuran *default* untuk *YOLO* yaitu 416×416 piksel. Ukuran 416×416 ini juga dipilih untuk mendapatkan proses komputasi yang cepat tanpa harus kehilangan akurasi sistem (Redmon and Farhadi, n.d.)

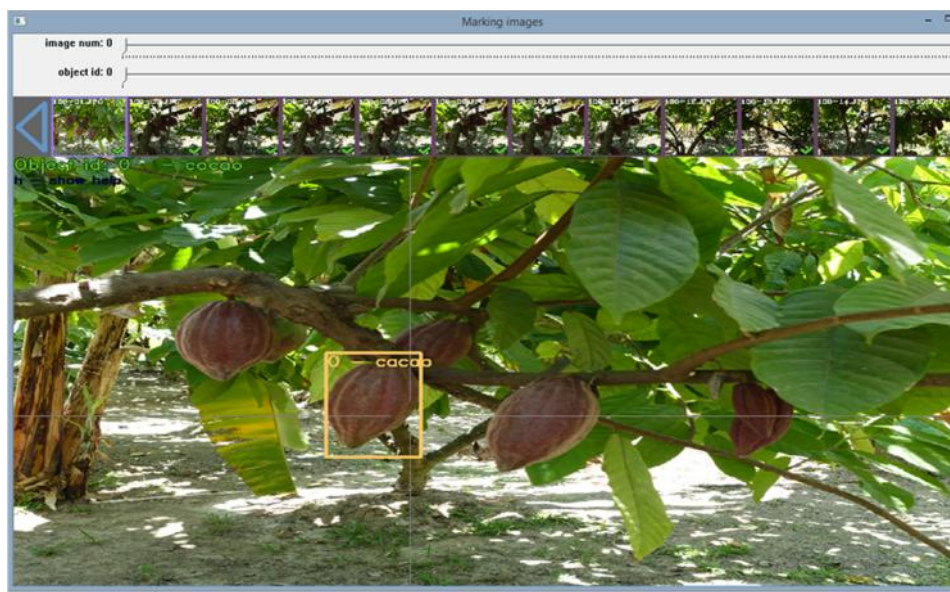
- b. Menentukan lokasi buah cokelat pada pohon dengan membuat *bounding box (bbox)* dan pelabelan kelas dengan nama “cacao”.

Gambar 25. menampilkan tahapan pelabelan dan pembuatan *bounding box* yang dilakukan menggunakan aplikasi *Yolo-mark*.



Gambar 25. Flowchart Labelling pada Yolo Mark

Pertama-tama dilakukan instalasi aplikasi *Yolo-Mark* dan memasukkan semua data latih ke dalam folder “img”. Selanjutnya adalah membuka file “*yolo-mark.cmd*”. Semua data latih yang berada pada folder “img” akan tampil pada aplikasi pelabelan dan selanjutnya dilakukan pelabelan manual pada objek yang dimaksud.



Gambar 26. Pelabelan manual yang dilakukan pada *Yolo Mark*

Pada proses ini tiap objek buah coklat pada gambar akan diberi kotak pembatas (*bounding box*) secara manual dan nama *class* yaitu “cacao” otomatis akan muncul pada sudut *bounding box*. Proses ini akan menghasilkan *output* berupa data berektensi .txt untuk masing-masing data latih yang berisi koordinat objek yang telah diberi label sebelumnya dan jumlahnya sesuai banyaknya *bounding box* dalam gambar. Contoh pelabelan gambar dan pembuatan *bounding box*

objek serta output berupa file *txt*. seperti terlihat pada Gambar 27 berikut:



(a)

File	Edit	Format	View	Help
0	0.213672	0.427778	0.100781	0.177778
0	0.276172	0.420139	0.044531	0.134722
0	0.366016	0.579167	0.089844	0.205556
0	0.425781	0.425000	0.087500	0.119444
0	0.540625	0.620139	0.114063	0.237500
0	0.753125	0.590278	0.042188	0.125000
0	0.764453	0.634722	0.063281	0.144444

(b)

Gambar 27. (a) Contoh pelabelan gambar dan pembuatan *bounding box* objek yang mendapat oklusi antara buah, (b) *Output file txt*.



(a)

File	Edit	Format	View	Help
0	0.526563	0.052083	0.064062	0.084722
0	0.852344	0.255556	0.051562	0.130556
0	0.910156	0.295139	0.093750	0.176389
0	0.374609	0.405556	0.127344	0.141667
0	0.471484	0.331944	0.110156	0.088889
0	0.591797	0.386111	0.122656	0.194444
0	0.636328	0.310417	0.038281	0.104167
0	0.435937	0.508333	0.148438	0.150000
0	0.527734	0.434028	0.164844	0.101389
0	0.130469	0.670833	0.062500	0.152778
0	0.364062	0.591667	0.093750	0.194444
0	0.323438	0.911806	0.129688	0.154167

(b)

Gambar 28. (a) Contoh pelabelan dan pembuatan *bounding box* untuk posisi buah *horizontal*, (b) *Output file txt*.

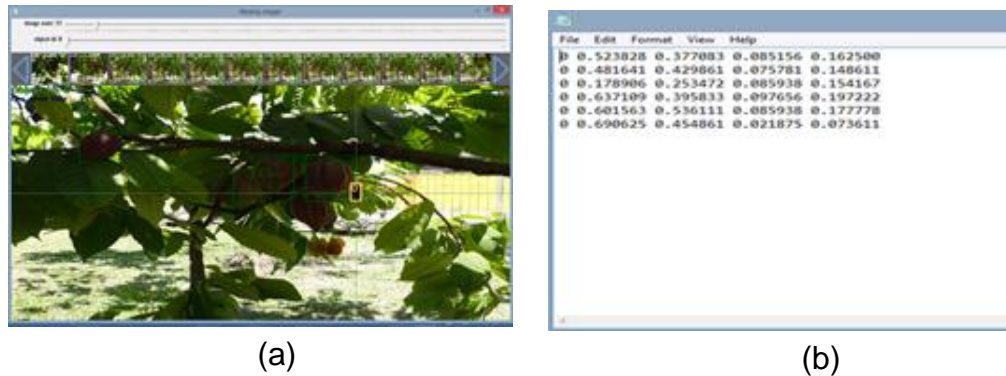


(a)

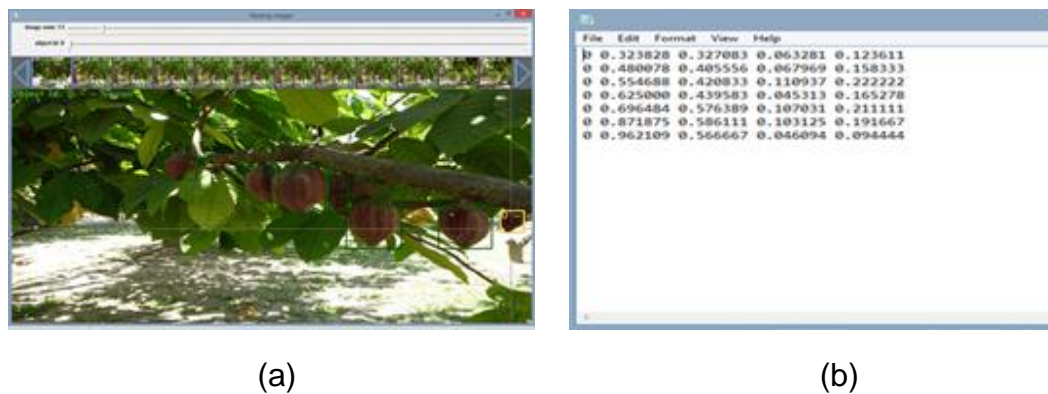
File	Edit	Format	View	Help
0	0.555469	0.436806	0.090625	0.070833
0	0.677734	0.430556	0.064844	0.108333
0	0.571875	0.545833	0.045313	0.088889
0	0.607813	0.506250	0.062500	0.062500
0	0.517578	0.731250	0.064844	0.137500
0	0.821094	0.336806	0.035937	0.087500
0	0.944141	0.350000	0.078906	0.155556
0	0.919922	0.180556	0.049219	0.094444
0	0.197266	0.770139	0.061719	0.126389
0	0.087891	0.806250	0.052344	0.095833
0	0.519922	0.679861	0.072656	0.031944
0	0.585938	0.191667	0.048437	0.158333

(b)

Gambar 29. (a) Contoh pelabelan dan pembuatan *bounding box* objek yang mendapat oklusi daun, (b) *Output file txt*.



Gambar 30. (a) Contoh pelabelan dan pembuatan *bounding box* objek berukuran kecil, (b) *Output file txt*.



Gambar 31. Contoh pelabelan dan pembuatan *bounding box* objek yang mendapat oklusi batang, (b) *Output file txt*.

Untuk tahap pelabelan ini jumlah gambar yang di proses pada aplikasi *yolo-mark* sebanyak 180 gambar yang di dalamnya terdapat objek yang telah diberi *bounding box*. Sedangkan jumlah bounding box yang ada pada data latih adalah sebanyak 1210 buah. Gambar seluruh data latih yang telah diberi label dan bounding box dapat dilihat pada lampiran gambar.

3) Tahap Training Data.

Setelah tahap pelabelan selesai, maka akan dihasilkan *file image beserta txt*, sebanyak data yang telah selesai proses pelabelannya. File tersebut akan di simpan pada folder data di dalam folder darknet.

Name	Date	Type	Size	Tags
100-01.JPG	10/5/2019 12:35 PM	JPEG image	5,397 KB	
100-01.txt	11/5/2019 11:29 PM	Text Document	1 KB	
100-05.JPG	10/5/2019 12:34 PM	JPEG image	5,815 KB	
100-05.txt	11/5/2019 11:32 PM	Text Document	1 KB	
100-06.JPG	10/5/2019 12:34 PM	JPEG image	5,782 KB	
100-06.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-07.JPG	10/5/2019 12:34 PM	JPEG image	5,828 KB	
100-07.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-08.JPG	10/5/2019 12:34 PM	JPEG image	5,852 KB	
100-08.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-09.JPG	10/5/2019 12:34 PM	JPEG image	5,833 KB	
100-09.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-10.JPG	10/5/2019 12:34 PM	JPEG image	5,894 KB	
100-10.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-11.JPG	10/5/2019 12:34 PM	JPEG image	5,977 KB	
100-11.txt	11/5/2019 11:33 PM	Text Document	1 KB	
100-12.JPG	10/5/2019 12:25 PM	JPEG image	6,219 KB	
100-12.txt	11/5/2019 11:36 PM	Text Document	1 KB	
100-13.JPG	10/5/2019 12:25 PM	JPEG image	5,974 KB	
100-13.txt	11/5/2019 11:36 PM	Text Document	1 KB	
100-14.JPG	10/5/2019 12:26 PM	JPEG image	6,146 KB	
100-14.txt	11/5/2019 11:36 PM	Text Document	1 KB	
100-15.JPG	10/5/2019 12:26 PM	JPEG image	5,401 KB	
100-15.txt	11/5/2019 11:37 PM	Text Document	1 KB	
100-16.JPG	10/5/2019 12:26 PM	JPEG image	5,582 KB	
100-16.txt	11/5/2019 11:42 PM	Text Document	1 KB	

Gambar 32. *Output* proses pelabelan gambar

Seluruh file tersebut akan diinput ke dalam *"train_obj.cmd"*.

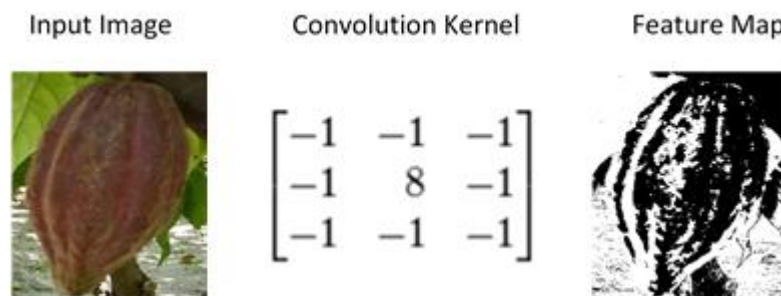
Dala proses training ini terjadi tahapan yang prosesnya berdasar pada Convolution Neural Network yaitu sebagai berikut:

4) Feature Extraction

Tahapan ekstraksi fitur ini prosesnya menggunakan *Convolutional Neural Network (CNN)*. Dalam *CNN* ada 2 proses yang akan terulang beberapa kali sampai menghasilkan *activation map* yaitu:

a) *Convolution*:

Layer pertama pada *feature extraction* adalah *convolutional layer*. *Convolutional layer* terdiri dari neuron yang membentuk sebuah filter dengan panjang dan tinggi (piksel). Filter ini akan digeser ke seluruh bagian gambar, setiap pergeseran akan dilakukan operasi dot antara input dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* yang disebut *activation map*. Ilustrasi proses convolution map menghasilkan feature map dapat dilihat pada Gambar 33 berikut:

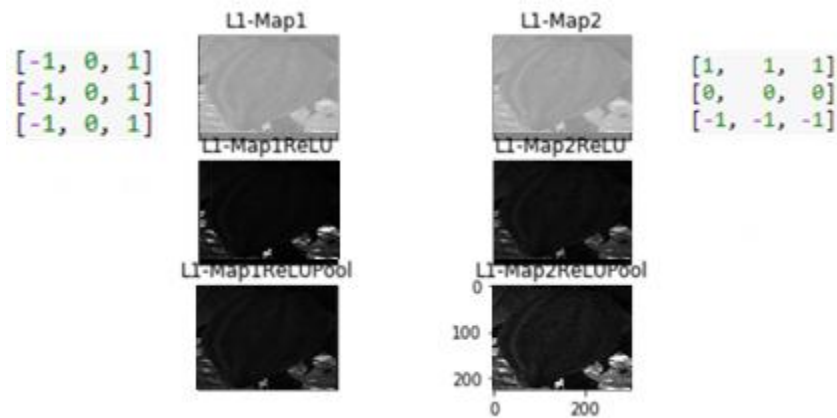


Gambar 33. Ilustrasi Output Activation Map

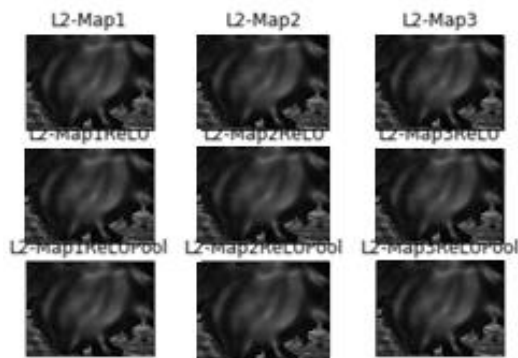
b) *Max Pooling*

Layer Max Pooling terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh *activation map* yang berasal dari proses convolution. *Pooling layer* digunakan untuk mempercepat komputasi karena parameter yang harus *update* akan menjadi semakin sedikit setelah proses *pooling* ini. Proses *convolution* dan *max pooling* akan terus berulang hingga membentuk *activation map* yaitu gambar yang sudah terbentuk

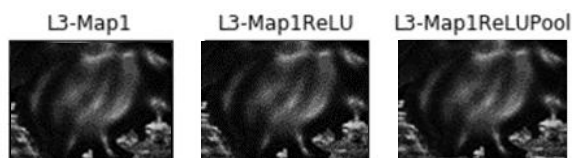
fitur-fiturnya setelah melewati filter *vertical* dan *horizontal* yang mengekstrak fitur yang berbeda dari gambar tersebut.



Gambar 34. Proses *convolution* dan *max pooling* pada citra.



Gambar 35. Proses *convolution* dan *max pooling* pada convolutional layer kedua.



Gambar 36. Activation map dari proses *convolution* dan *max pooling*

Penjelasan berikut adalah contoh proses ekstraksi fitur untuk citra berukuran 416×416 yang hasilnya berupa fitur map hasil dari network

yang memiliki ukuran 13×13 . Proses pendeteksian pertama akan dilakukan dengan menggunakan *filter* pendeteksi berukuran 1×1 , sehingga memberikan *fitur map* dengan ukuran $13 \times 13 \times 255$.

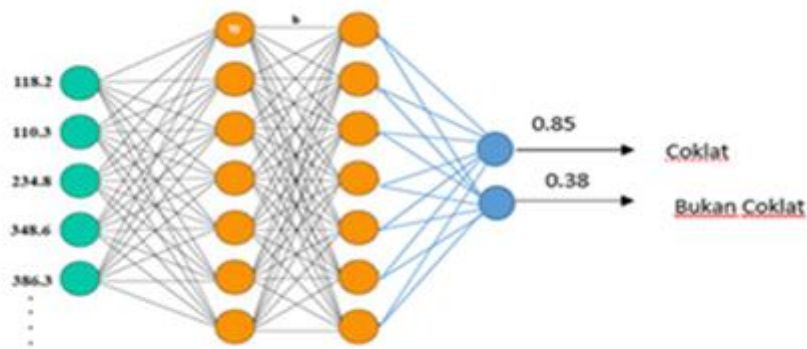
Kemudian, *fitur map* dari *layer* 79 akan ditujukan ke beberapa *layer* konvolusi sebelum di *upsampling* ke dimensi berukuran 26×26 . *Fitur map* ini memiliki hubungan dengan *fitur map* pada *layer* 61. Kemudian *fitur map* yang ada akan diteruskan ke beberapa *convolutional layer* untuk digabungkan dengan *fitur* yang berasal dari *layer* yang ada lebih awal (*layer* 61). Kemudian, proses deteksi kedua dilakukan pada *layer* ke 94, dengan *fitur map* hasil deteksi berukuran $26 \times 26 \times 255$.

Prosedur yang sama kemudian dilakukan, dimana *fitur map* dari *layer* 91 akan diteruskan ke beberapa *convolutional layer* sebelum digabungkan dengan *fitur map* dari *layer* 36. Setelah itu proses deteksi ketiga atau *final* akan dilakukan pada *layer* ke 106, dengan *fitur* berukuran $52 \times 52 \times 255$.

5) Classification

Tahap akhir dari algoritma Yolo sebelum sampai pada *output prediction* adalah proses klasifikasi. Hasil dari ekstraksi fitur akan menjadi input pada proses ini dimana proses ini menggunakan *neural network fully connected layer* yang memiliki *input layer*, *hidden layer*, *output layer*, *activation function*, dan *loss function*. Input dari *neural network* ini dalam bentuk *vector*, sehingga *activation map* yang hasilnya berbentuk *array* harus melewati proses *reshape*. Hasil dari

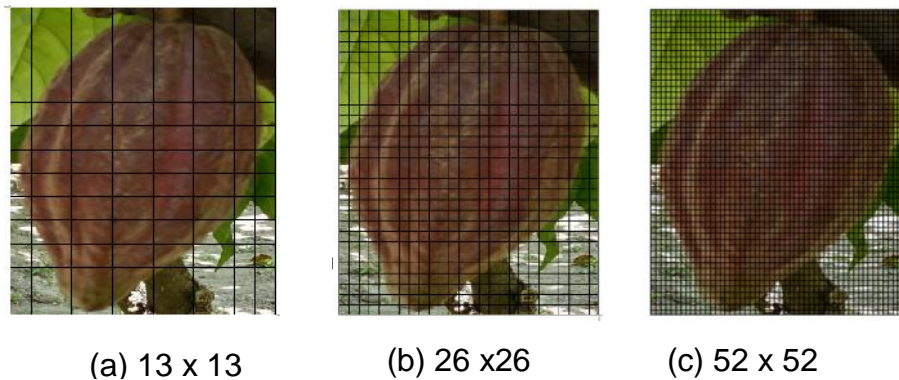
reshape ini akan menjadi input untuk *fully connected layer* pada proses klasifikasi seperti pada Gambar 37 berikut:



Gambar 37. *Fully connected layer* pada klasifikasi objek buah cokelat

6) Output Prediction:

YOLO v3 membuat prediksi di 3 skala yang berbeda. Layer deteksi digunakan untuk melakukan deteksi pada *feature map* dengan tiga ukuran berbeda, masing-masing memiliki langkah 32, 16, 8. Ini berarti, dengan input 416 x 416, maka dilakukan deteksi pada skala 13 x 13, 26 x 26 dan 52 x 52. Jaringan melakukan downsampling gambar input hingga lapisan deteksi pertama, di mana deteksi dibuat menggunakan *feature map layer* dengan stride 32. Selanjutnya, layer di-*upsampled* oleh faktor 2 dan digabungkan dengan *feature map layer* sebelumnya yang memiliki *feature map* yang ukurannya identik. Deteksi selanjutnya dilakukan pada layer dengan stride 16. Prosedur *upsampling* yang sama diulang, dan deteksi akhir dilakukan pada layer dengan stride 8.



Gambar 38. Contoh 3 skala prediksi bounding box untuk objek buah coklat

Pada setiap skala, dilakukan 3 prediksi bounding box dengan 3 anchor yang berbeda, sehingga total anchor yang digunakan adalah 9. Dengan melakukan prediksi menggunakan 3 skala tersebut YOLO v3 terbukti lebih baik dalam melakukan prediksi terhadap objek kecil dibandingkan YOLO versi sebelumnya dimana proses upsampling sangat membantu jaringan dalam mempelajari fitur dari objek kecil.

7) Deteksi Objek

Tahap selanjutnya adalah proses deteksi dimana untuk penelitian ini setiap objek yang telah diberi bounding box, sistem akan melakukan prediksi sebanyak $((52 \times 52) + (26 \times 26) + (13 \times 13)) \times 3 = 10647$ bounding box. Selanjutnya bagaimana menjadikan prediksi sebanyak 10647 bounding box sehingga akhirnya hanya terpilih 1 bounding box yang

```

18 ap.add_argument("-c", "--confidence", type=float, default=0.5,
19     help="minimum probability to filter weak detections")
20 ap.add_argument("-t", "--threshold", type=float, default=0.3,
21     help="threshold when applying non-maxima suppression")

```

akan terdeteksi sebagai objek? Pertama-tama, sistem akan melakukan filter terhadap bounding box yang memiliki *confidence score* sesuai nilai *threshold* yang telah ditetapkan dalam hal ini adalah 0.5, sehingga semua *bounding box* yang mempunyai nilai di bawah 0.5 akan diabaikan. Selanjutnya jika terjadi multi deteksi pada satu gambar maka disinilah fungsi nilai threshold dari *Non Maxima Supression* (NMS) dimana sistem akan menyimpan *bounding box* dengan nilai terbaik dan menghapus semua *bounding box* yang berada di bawah ambang batas *threshold* dari NMS. Sehingga pada akhirnya hanya terdapat 1 *bounding box* yang mempunyai nilai *confidence* tertinggi dari seluruh prediksi yang ada.

8) Tahapan Pengujian Sistem

Setelah proses training selesai dan telah di dapatkan model untuk diujikan pada sistem deteksi objek maka proses pengujian sistem dilakukan dengan tahapan sebagai berikut:

- a. Data uji sebanyak 45 gambar akan diinput ke dalam sistem dan selanjutnya akan melewati proses ekstraksi fitur dan klasifikasi memakai YOLO model hasil training data sebelumnya.
- b. Gambar yang memiliki *bounding box* dengan nilai *confidence* yang sesuai dengan nilai *threshold* yang ditetapkan sebelumnya, dalam hal ini nilainya > 0.5 akan dikenali sebagai objek buah cokelat.

9) Visualisasi

Visualisasi hasil deteksi buah cokelat dapat dilihat pada Gambar 39 berikut:



Gambar 39. Visualisasi Hasil deteksi buah cokelat menggunakan YOLO model

F. INSTRUMENTASI PENELITIAN

Instrumen yang digunakan dalam penelitian ini meliputi:

1. *Software*

- a) Windows 10 x64
- b) Jupyter Notebook Framework
- c) Google Colab
- d) Python 3.6

2. *Hardware*

- a) Laptop ASUS, RAM 8 GB, Prosesor Intel(R) Core(TM) i5-4210U CPU @2.20GHz

b) *Drone DJI Mavic Air*

d) *Smartphone*

G. SUMBER DATA

Data yang digunakan dalam penelitian ini adalah data primer yang merupakan video buah cokelat pada pohon yang diambil dengan 3 variasi jarak yang selanjutnya dilakukan ekstraksi menjadi gambar yang dibagi menjadi 2 jenis data sebagai berikut:

a. Data Latih

Data latih sebanyak 180 gambar berukuran 3840 x 2160 dan digunakan dalam proses training untuk menghasilkan sistem otomatis pendeteksian buah cokelat pada pohonnya menggunakan algoritma *You Only Look Once v3 (YOLO)*.

b. Data Uji

Data uji sebanyak 45 gambar berukuran 3840 x 2160 pixel yang di resize menjadi 900 x 900 pixel dan kemudian diujikan pada sistem untuk menentukan apakah algoritma *You Only Look Once v3 (YOLO)* cocok untuk mendeteksi buah cokelat pada pohonnya.

H. TOLAK UKUR AKURASI SISTEM

Setelah dilakukan perancangan sistem dengan metode CNN Yolo v3 untuk deteksi dan penghitungan buah cokelat, dilanjutkan dengan pengujian data pada sistem yang telah dirancang dan pengujian unjuk kerja sistem. Pada tahap ini dilakukan penghitungan nilai akurasi jumlah buah

pada 5 pohon berbeda dan nilai akurasi pada masing-masing gambar data uji dengan 3 jarak yang berbeda dengan menggunakan Persamaan 3.1 berikut:

$$A = \frac{Ns}{Nt} \times 100\% \quad 3.1$$

Nilai akurasi kinerja sistem (A_c) diperoleh dengan persentase jumlah buah cokelat hasil penghitungan sistem (N_s) terhadap jumlah total buah cokelat hasil penghitungan yang dilakukan manual (N_t).

BAB IV . HASIL DAN PEMBAHASAN

Pada penelitian ini, sistem deteksi buah cokelat pada pohonnya menggunakan metode CNN Yolo v3 dengan 180 data latih dan 45 data uji yang diambil menggunakan 3 variasi jarak (d).

Gambar 40, 41 dan 42 memperlihatkan contoh hasil deteksi pada 3 d berbeda:



Gambar 40. Contoh hasil deteksi dan hitung otomatis jarak 100 cm



Gambar 41. Contoh hasil deteksi dan hitung otomatis jarak 150 cm



Gambar 42. Contoh hasil deteksi dan hitung otomatis jarak 200 cm

Pada Gambar 40 terlihat adanya buah yang mendapat oklusi dari daun dan buah yang bertumpuk sehingga sistem tidak dapat mendeteksi objek tersebut sebagai buah. Sedangkan objek yang terdeteksi sebagai buah cokelat akan menampilkan *bounding box* dengan label 'cacao' dan

tercantum *confidence score* yaitu tingkat kepercayaan sistem terhadap objek yang diprediksi sebagai buah cokelat. Nilai *confidence score* (*cf*) ini berhubungan dengan nilai *Intersection over Union* (*IoU*) sebuah citra, yaitu perbandingan *bbox* yang dilakukan manual saat pelabelan dan *bbox* hasil deteksi seperti yang telah dijelaskan pada bab 3.

Sedangkan pada Gambar 41 terlihat buah cokelat yang ukurannya terlalu kecil sehingga tidak terdeteksi dan dianggap bukan buah cokelat.

Akurasi sistem tiap jarak akan diuji menggunakan gambar buah cokelat berukuran 900×900 untuk dilakukan deteksi dan penghitungan secara otomatis.

Tabel 4. menunjukkan tingkat akurasi sistem untuk semua data uji:

Tabel 4. Akurasi sistem YOLO

Data	Jarak (<i>d</i>)	Perhitungan Sistem (<i>Ns</i>)	Perhitungan Manual (<i>Nt</i>)	Akurasi (%)
Gbr.1	100	5	6	83
Gbr.2	100	5	6	83
Gbr.3	100	5	6	83
Gbr.4	100	5	6	83
Gbr.5	100	3	3	100
Gbr.6	100	3	3	100
Gbr.7	100	5	6	80
Gbr.8	100	5	6	80
Gbr.9	100	3	3	100

Data	Jarak (d)	Perhitungan Sistem (N_s)	Perhitungan Manual (N_t)	Akurasi (%)
Gbr.10	100	10	12	80
Gbr.11	100	10	12	80
Gbr.12	100	4	5	80
Gbr.13	100	13	15	87
Gbr.14	100	14	15	90
Gbr.15	100	13	15	87
Gbr.16	150	6	8	75
Gbr.17	150	7	9	77
Gbr.18	150	13	15	87
Gbr.19	150	12	15	80
Gbr.20	150	12	15	80
Gbr.21	150	13	15	87
Gbr.22	150	13	15	87
Gbr.23	150	13	15	87
Gbr.24	150	11	15	73
Gbr.25	150	12	15	80
Gbr.26	150	7	8	88
Gbr.27	150	7	8	88
Gbr.28	150	8	10	80
Gbr.29	150	8	10	80
Gbr.30	150	8	10	80

Data	Jarak (d)	Perhitungan Sistem (N_s)	Perhitungan Manual (N_t)	Akurasi (%)
Gbr.26	150	7	8	88
Gbr.27	150	7	8	88
Gbr.28	150	6	8	75
Gbr.29	150	6	8	75
Gbr.30	150	7	8	88
Gbr.31	200	8	10	80
Gbr.32	200	8	10	80
Gbr.33	200	16	18	88
Gbr.34	200	17	18	94
Gbr.35	200	13	16	80
Gbr.36	200	13	16	80
Gbr.37	200	15	16	93
Gbr.38	200	15	17	88
Gbr.39	200	15	19	78
Gbr.40	200	14	18	78
Gbr.41	200	8	11	72
Gbr.42	200	17	19	90
Gbr.43	200	7	8	88
Gbr.44	200	14	16	88
Gbr.45	200	14	16	88

Pada beberapa data di $d = 100$ cm, objek buah cokelat dapat terdeteksi 100 persen, namun ada pula yang terdeteksi hanya 70 persen. Akurasi penghitungan oleh sistem dipengaruhi adanya oklusi dari batang, daun dan antar buah juga oleh kestabilan terbang *drone* saat pengambilan gambar. Jarak pengambilan gambar sangat berpengaruh sehingga untuk perhitungan akurasi tiap gambar berdasarkan jarak, pada $d = 100$ cm diperoleh akurasi 100% dan akurasi terendah yaitu 72% pada $d = 200$ cm. Untuk $d = 100$ cm, buah yang mendapat oklusi dari daun masih bisa terdeteksi sebagai objek seperti yang terlihat pada Gambar 43 berikut:



Gambar 43. Objek yang mendapat oklusi daun dan bisa terdeteksi untuk $d = 100$ cm.

Tabel 5. Menunjukkan akurasi perhitungan jumlah buah dari 5 pohon berbeda menggunakan 3 variasi jarak (d):



Tabel 5. Akurasi sistem $d = 100\text{cm}$, $d = 150\text{ cm}$, $d = 200\text{ cm}$

Nama	100 cm			150 cm			200m		
	N_s	N_t	Akurasi (%)	N_s	N_t	Akurasi (%)	N_s	N_t	Akurasi (%)
Pohon 1	5	5	100	5	6	83	4	6	67
Pohon 2	13	15	87	16	18	89	10	18	56
Pohon 3	5	6	83	7	9	78	8	11	73
Pohon 4	5	6	83	7	9	78	7	9	78
Pohon 5	10	12	83	12	15	80	17	18	94
Akurasi (%)	87.2			81.6			73.6		

Akurasi rata-rata perhitungan jumlah buah pada $d = 100$ yaitu 87.2% dan terendah di $d = 200$ yaitu 73.6% untuk 5 pohon berbeda.

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. Untuk menguji unjuk kerja sistem ini, maka sistem juga diuji menggunakan objek yang berada di sekitar pohon coklat dan hasilnya dapat dilihat pada Tabel 4 berikut:

Tabel 6. Uji sistem data negatif

Gambar	Uraian	Hasil
	Daun kering pada pohon cokelat.	Tidak terdeteksi
	Daun yang berwarna kuning pada pohon cokelat.	Tidak terdeteksi

BAB V KESIMPULAN DAN SARAN

a. Kesimpulan

Setelah melakukan perancangan sistem untuk deteksi serta hitung otomatis buah cokelat secara *realtime* maka dapat disimpulkan bahwa:

1. Menggunakan metode *Deep Learning YOLO v3* yang melewati training dengan *framework* Darknet 53 dan jumlah iterasi sebanyak 4000, diperoleh nilai akurasi rata-rata jumlah buah pada 5 pohon berbeda adalah: tertinggi di jarak 100 cm yaitu 87.2 % dan terendah yaitu 73.6 % di jarak 200 cm. Sehingga disimpulkan bahwa metode ini adalah salah satu metode yang tepat untuk mendeteksi dan menghitung otomatis buah cokelat pada pohon secara *realtime*.
2. Untuk $d = 100$ cm pada beberapa gambar, sistem dapat mendeteksi objek walaupun mendapat oklusi. Selain faktor oklusi pada objek, proses pengambilan data menggunakan drone yang tidak stabil menyebabkan hasil gambar menjadi blur sehingga objek tidak terdeteksi.

b. Saran

Untuk meningkatkan akurasi atau mengembangkan sistem yang telah dibuat, terdapat beberapa hal yang dapat dilakukan, yaitu:

1. Menambah jumlah data training dan iterasi untuk mendapatkan hasil yang maksimal.

2. Merancang sistem pengiriman data yang lebih baik dari *drone* ke PC sehingga lebih menghemat waktu proses dan mengurangi *delay time*
3. Mencoba metode lain untuk mengurangi komputasi pada saat *training*, dapat mendeteksi dan menghitung jumlah objek lebih akurat.

DAFTAR PUSTAKA

- Choudhary, P., Khandekar, R., Borkar , A., & Chotaliya, P. (2017). Image Processing Algorithm For Fruit Identification . *International Research Journal of Engineering and Technology (IRJET)*, Vol. 04 No.03.
- Lainez, S.M.D., Gonzales, D.B., 2019a. Automated Fingerlings Counting Using Convolutional Neural Network, in: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). Presented at the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), IEEE, Singapore, pp. 67–72. <https://doi.org/10.1109/CCOMS.2019.8821746>
- KEMENPERIN, S. (2007). *Gambaran Sekilas Industri Kakao*. Jakarta: Departemen Perindustrian.
- Kusuma S, F., Pawening, R, E., & Dijaya, R. (2017). Automatic Fruit Count On Coffee Branches Using Computer Vision. *Jurnal Ilmu Teknologi Sistem Informasi*.
- Nisar, H., Yang, H. Z., & Ho, Y. K. (2015). Predicting Yield of Fruit and Flowers Using Digital Image Analysis. *Indian Journal and Science and Technology*, Vol 8(32).
- Putra, D. (2009). *Pengolahan Citra Digital*. Yogyakarta: Andi Yogyakarta.
- Rosyadi, S. (2018). *Tantangan Peluang Bagi Revolusi Industri 4.0*.
- Syal, A., Garg, D., & Sharma, S. (2014). Apple Fruit Detection and Counting Using Computer Vision Techniques. *IEEE* .
- Zheng Lihua, X., Chen Yuan, L., & Chunyan, M. (2015). Apple Detection From Apple Tree Image Based on BP Neural Network and Hough Transform . *International Journal Agricultural & Biology Engineering*, Vol. 8 No.6.

- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J., Farhadi, A., n.d. YOLOv3: An Incremental Improvement 6.
- Romano, N., Zeng, C., 2017. Cannibalism of Decapod Crustaceans and Implications for Their Aquaculture: A Review of its Prevalence, Influencing Factors, and Mitigating Methods. *Reviews in Fisheries Science & Aquaculture* 25, 42–69. <https://doi.org/10.1080/23308249.2016.1221379>
- Yang, W., Jiachun, Z., 2018. Real-time face detection based on YOLO, in: 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII). Presented at the 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), IEEE, Jeju, pp. 221–224. <https://doi.org/10.1109/ICKII.2018.8569109>