

**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**

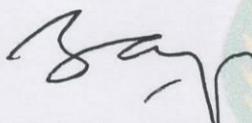
**D42116304**

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.

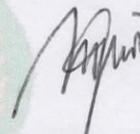
Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,



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



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

Ketua Program Studi,



Dr. Indrabayu, S.T., M.T., M.Bus.Sys.  
Nip. 19750716 200212 1 004

## 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 pendeteksian 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 otomatis 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 coklat 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  $1920 \times 1080$  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 diperoleh 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, mencocokkan 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, penganan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian 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 perangkat 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

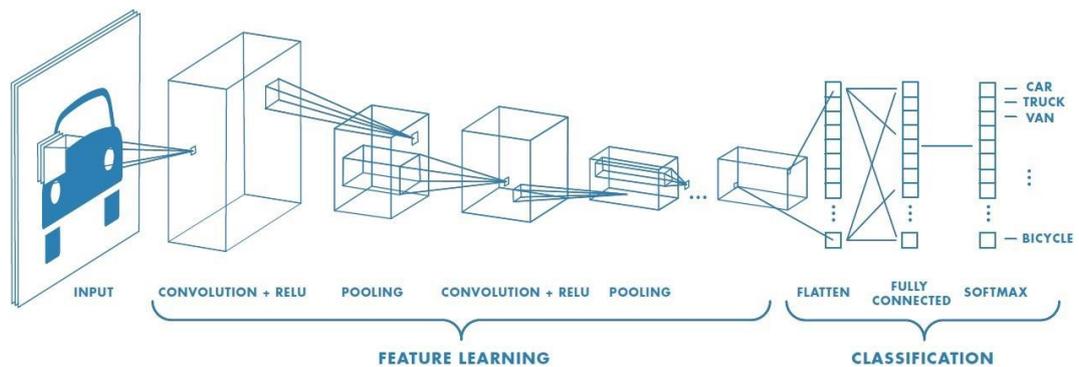
pada data menggunakan sekumpulan fungsi transformasi non-linier yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised 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 hirarki 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.

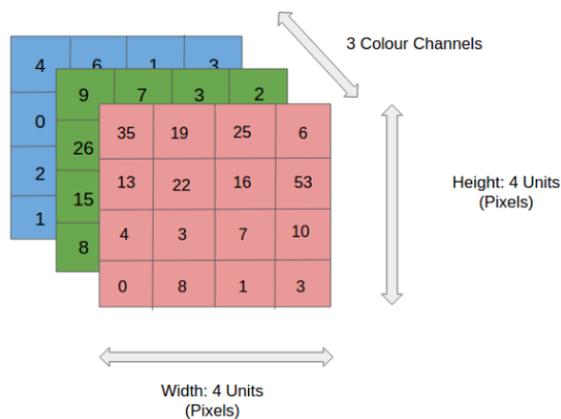


**Gambar 2.2** Proses *Convolutional Neural Network*

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  $4 \times 4$  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  $1 \times 1$ ,  $3 \times 3$ , atau  $5 \times 5$ . 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

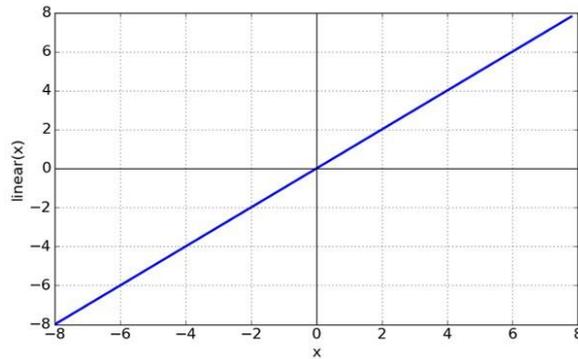
**Gambar 2.4** Proses Konvolusi

### 3. *Activation Layer*

*Activation Layer* terletak sebelum melakukan *Pooling Layer* dan setelah melakukan *Convolution Layer*. Pada lapisan ini, terjadi proses perubahan 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 ditunjukkan oleh **Gambar 2.5**.



**Gambar 2.5** Fungsi Linear

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

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

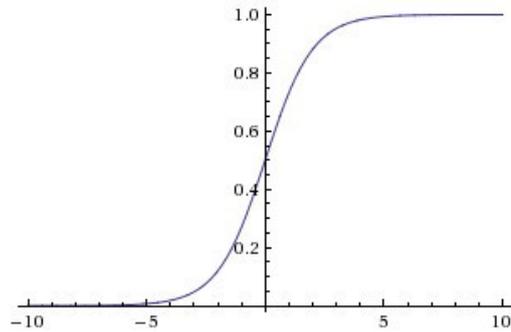
- Fungsi Aktivasi Non-linier

Fungsi aktivasi nonlinear adalah fungsi aktivasi yang paling banyak digunakan. Fungsi aktivasi non-linier memudahkan model untuk menggeneralisasi 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

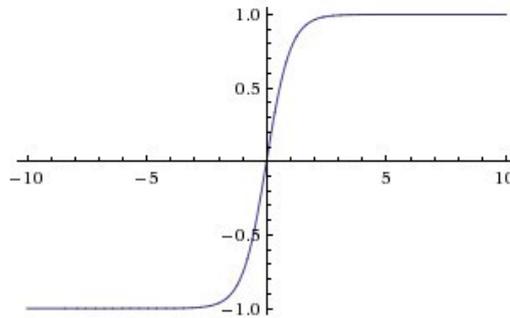
Sigmoid memiliki persamaan sebagai berikut:

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

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*).

- o 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 kelebihan adalah output yang dimiliki Tanh merupakan zero-centered. Dalam praktiknya Tanh lebih menjadi pilihan jika dibandingkan dengan Sigmoid.

Tanh memiliki persamaan sebagai berikut:

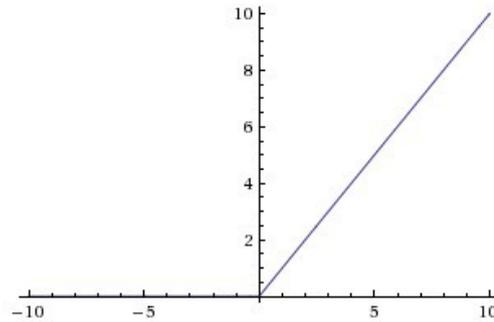
$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.3)$$

- 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:

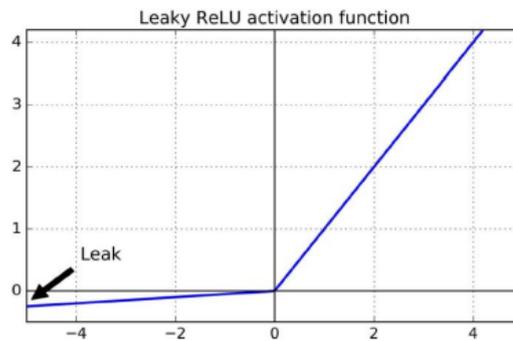


**Gambar 2.8** Fungsi ReLU

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**.



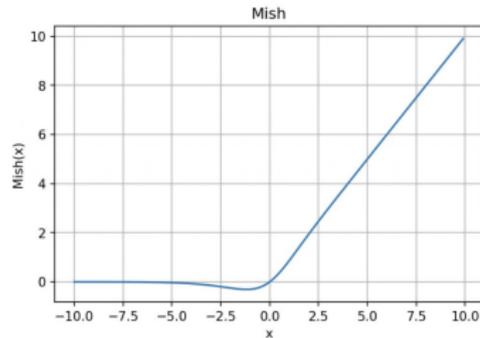
**Gambar 2.9** Fungsi Leaky ReLU

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  $\leq 0$  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**.



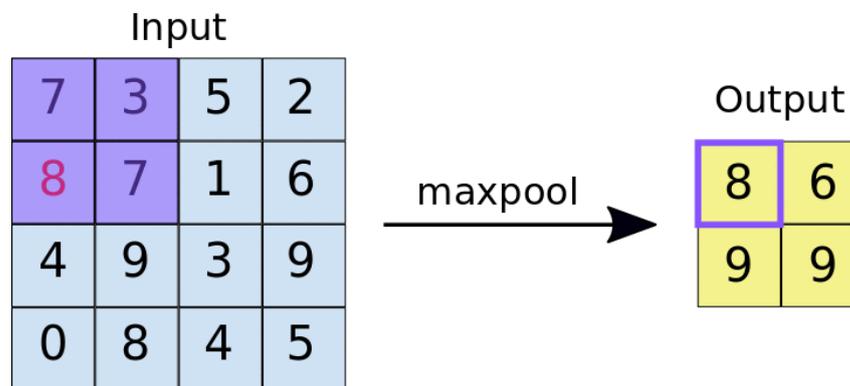
**Gambar 2.10** Fungsi Mish

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).

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

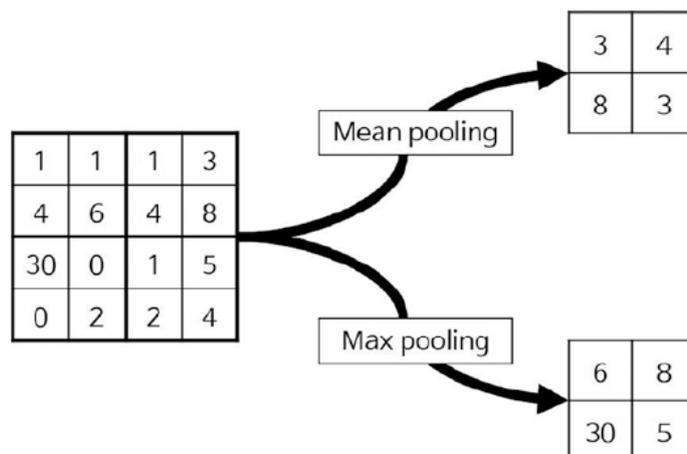
#### 4. Pooling Layer

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  $2 \times 2$  pixel dan gambar yang dilakukan *max pooling* berukuran  $4 \times 4$  pixel. Hasilnya akan membuat sebuah gambar baru dengan ukuran  $2 \times 2$ .



**Gambar 2.11** Contoh *max pooling*

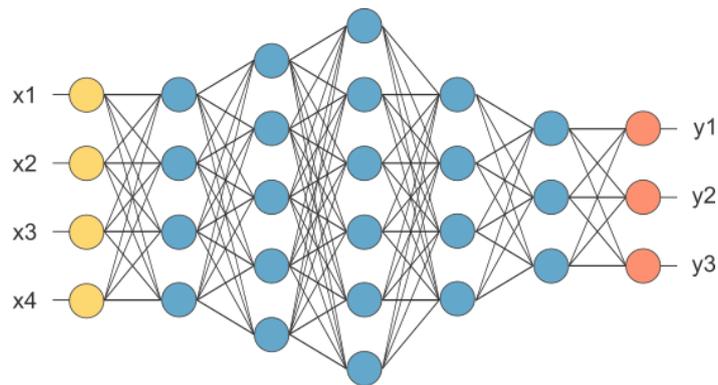
Terdapat beberapa macam pooling lain seperti *max pooling*, *min pooling*, *mean pooling*, *sum pooling*, dll **Gambar 2.12** menunjukkan contoh *max pooling* dan *mean pooling* pada gambar yang sama.



**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 halnya 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).

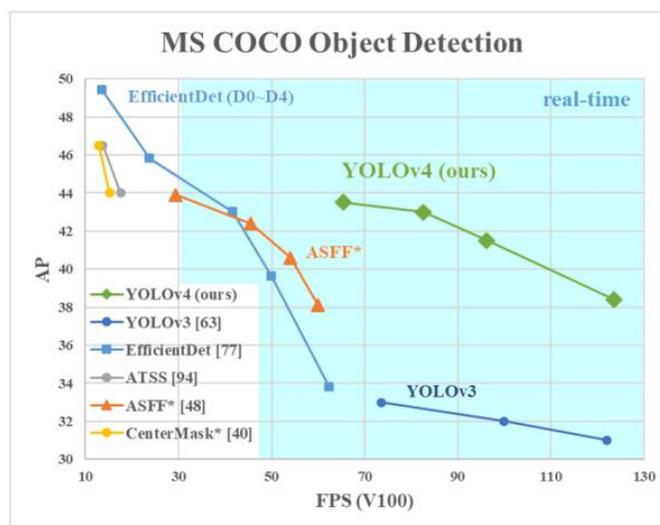
## 2.6 YOLO (You Only Look Once)

YOLO atau *You Only Look Once* merupakan sebuah algoritma yang menggunakan *convolutional neural networks* untuk pendeteksian objek. YOLO merupakan salah satu algoritma pendeteksi 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).

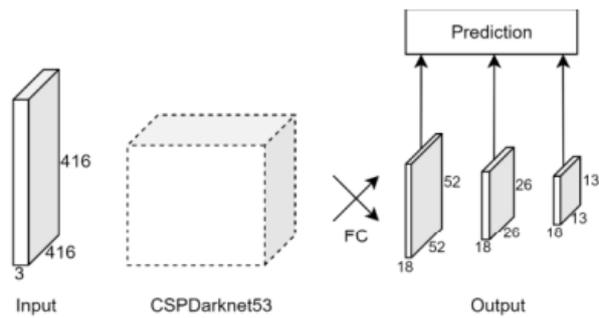
YoloV4 adalah pengembangan dari versi sebelumnya, yaitu YoloV3. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao mengusulkan algoritma YoloV4 dengan perubahan signifikan dari versi sebelumnya dan akurasi yang jauh lebih baik yaitu dengan peningkatan AP (*Average Precision*) dan FPS (*Frame Per 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 pendeteksian 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 dimasukkan kedalam arsitektur backbone CSPDarknet53 (Dhiaegana & Munir, 2020).