

SKRIPSI

**DETEKSI GENANGAN SAMPAH PADA PERMUKAAN AIR KANAL  
DENGAN MENGGUNAKAN METODE *DEEP LEARNING***

Disusun dan diajukan oleh

ARTAMEVIA KHAIRUNNISA EKA AMRULLAH

D421 15 508



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2021**

## LEMBAR PENGESAHAN SKRIPSI

### DETEKSI GENANGAN SAMPAH PADA PERMUKAAN AIR KANAL DENGAN MENGGUNAKAN METODE *DEEP LEARNING*

Disusun dan diajukan oleh

**ARTAMEVIA KHAIRUNNISA EKA AMRULLAH**  
**D421 15 508**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 3 Maret 2021 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama,

Pembimbing Pendamping,

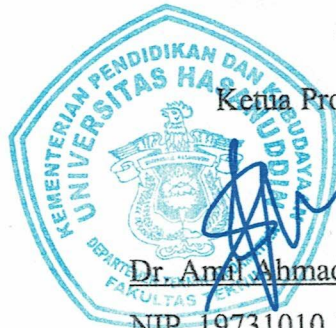
Dr. Indrabayu, ST., M.T., M.Bus.Sys

NIP. 19750716 200212 1 004

Dr. Ir. Ingrid Nurtanio, M.T

NIP. 19610813 198811 2 001

Ketua Program Studi,



Dr. Amri Ahmad Ilham, S.T., M.IT.

NIP. 19731010 199802 1 001

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Artamevia Khairunnisa Eka Amrullah  
NIM : D421 15 508  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulis saya berjudul

Deteksi Genangan Sampah pada Permukaan Air Kanal  
dengan menggunakan Metode *Deep Learning*

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

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

Makassar, 3 Maret 2021

Yang Menyatakan



Artamevia Khairunnisa Eka Amrullah

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul “DETEKSI GENANGAN SAMPAH PADA PERMUKAAN AIR KANAL DENGAN MENGGUNAKAN METODE *DEEP LEARNING*” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Kedua orang tua penulis, terkhusus untuk ayahanda penulis Bapak (Alm) Drs. Mayor. Inf. Amrullah Daming yang senang dan tersenyum dari alam sana melihat penulis telah menyelesaikan masa studi S1 dan teruntuk ibunda penulis tersayang Ibu Dr. Ratna Sari., SE., M.Si., Ak., CA., yang selalu memberikan dukungan baik berupa moril seperti doa, motivasi dan semangat maupun dukungan materil, serta selalu sabar dalam mendidik penulis sejak kecil hingga berada pada tahap ini dan berhasil menyelesaikan penyusunan tugas akhir ini.
2. Bapak Dr. Indrabayu S.T., M.T., M.Bus.Sys. selaku pembimbing I, yang senantiasa memberikan saran-saran serta bantuan selama proses pengambilan data hingga selesainya sistem ini dibuat, dan Ibu Dr. Ir. Ingrid Nurtanio, M.T. selaku

- pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, semangat dan perhatian yang luar biasa dalam membimbing penulis menyusun tugas akhir.
3. Bapak Dr. Amil Ahmad Ilham, ST., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membimbing penulis selama masa perkuliahan.
  4. Ibu Dr. Eng. Intan Sari Areni, S.T., M.T., yang senantiasa memberikan nasehat, kritik, masukan, serta semangat kepada penulis dalam menyusun tugas akhir.
  5. Bapak Ir. Christoforus Yohannes, M.T., Bapak Dr. Ir. Zahir Zainuddin, M.Sc., Bapak Dr.Eng. Muhammad Niswar,ST., M.IT., Bapak Dr. Adnan, S.T., M.T., Ibu Anugrahyani Bustamin., S.T., M.T., Ibu Novy Nur R.A Mokobombang, S.T., Ms.TM., Ibu Elly Warni, S.T., M.T., Bapak Dr. Eng. Zulkifli Tahir, S.T., M.Sc., Bapak A. Ais Prayogi Alimuddin, S.T., M.Eng. selaku tenaga pengajar Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah memberikan bimbingan dan ilmunya kepada penulis selama masa perkuliahan.
  6. Kakak Rahma, Adek Fauzan, dan Adek Husain yang selalu menemani, dan memberikan semangat, serta mendoakan penulis saat penulis menyusun tugas akhir.
  7. Teman-teman, Kakak-kakak dan Adek-adek AIMP *Research Group* FT UH yang telah memberikan masukan dan bantuan selama penelitian berupa proses

pengambilan data, diskusi penyusunan tugas akhir, dan bantuan sehari-hari lain selama menjadi anggota AIMP.

8. Lutvi Harnantik Salsabila S.T yang senantiasa memberikan semangat dan mendorong penulis untuk menyelesaikan tugas akhir.
9. Saudara Rian Nugraha SH., Muh. Kusdianto Frans S.Pd., Iqra Nurhuda S.M., Nur Alisha Chandra S.Psi, dan Isda Ilyas SE., selaku sahabat-sahabat PENCITRAAN yang senantiasa membantu, menemani dan memberikan semangat kepada penulis untuk menyelesaikan penyusunan tugas akhir ini.
10. Teman-teman HYPERV15OR FT-UH atas dukungan dan semangat yang diberikan selama ini kepada penulis.
11. Segenap Staff dan Dosen Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis.
12. Orang-orang berpengaruh lainnya yang tanpa sadar telah memotivasi dan membantu penulis.

Wassalam

Makassar, Januari 2020

Penulis

## ABSTRAK

Kota Makassar memiliki tiga aliran kanal utama yang mengatur sistem panyaluran drainase untuk pembuangan yakni, Kanal Panampu, Jongaya dan Sinrijala. Pembuatan kanal merupakan upaya mengatasi banjir dengan memanfaatkan kelebaran dan kedalaman kanal. Di samping itu juga dimaksudkan agar berpotensi sebagai area penyaluran dan pembuangan air hujan kawasan setiap saat. Namun faktanya, kanal mengalami pendangkalan sungai akibat sedimentasi beragam limbah, tumpukan sampah membuat fungsi kanal di Kota Makassar terganggu, Ini dipicu karena kesadaran masyarakat yang masih kurang dalam menjaga kebersihan lingkungan. Selain terganggunya ekosistem perairan secara signifikan, kualitas air yang buruk berdampak pada tercemarnya Pantai Losari sebagai salah-satu landmark Kota Makassar. Masalah kesehatan juga berpotensi muncul akibat pencemaran ini, seperti diare, demam berdarah, dan lain sebagainya. Tujuan penelitian ini membuat suatu sistem yang dapat mendeteksi genangan sampah pada permukaan perairan, agar dapat mengurangi jumlah sampah yang ada diperaian kanal. Dengan memanfaatkan teknologi *deep learning*, dapat dibuat sebuah sistem yang dapat mendeteksi keberadaan genangan sampah di atas permukaan perairan. Objek penelitian adalah genangan sampah dan benda disekitarnya pada kanal Jongaya karena kanal Jongaya terkoneksi menuju kanal Jalan Metro Tanjung Bunga dan Pantai Losari. Pengambilan data menggunakan *drone* dengan format video. Data yang digunakan dalam penelitian ini sebanyak 676 citra gambar dan dibagi menjadi 540 data *training* serta 135 data *testing*. Adapun parameter yang digunakan dalam sistem ini adalah *batch* 64, *subdivisions* 16, *width* 416, *height* 416, *channel* 3, *momentum* 0.9, *learning rate* 0.001, *steps* 3200:3600, *scale* 0.1:0.1 dan *max\_batches* 4000. Sedangkan *algoritma* yang digunakan yaitu *algoritma You Only Look Once* (YOLO) generasi ke-3 untuk pendeteksian objek. Hasil rata-rata akurasi yang diperoleh sistem ini adalah sebesar 96,02%. Hasil penelitian menunjukkan bahwa data dari sistem yang telah dibuat dapat membantu petugas kebersihan untuk mendeteksi benda diatas permukaan air kanal sampah atau bukan sampah sehingga dapat mengefisiensikan waktu, tenaga dan biaya ketika melakukan pembersihan kanal.

**Kata kunci** : sampah, genangan sampah, *object detection*, *deep learning*, *convolutional neural network*, YOLOV3.

## DAFTAR ISI

KATA PENGANTAR .....	ii
ABSTRAK v	
DAFTAR ISI vi	
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi
BAB I .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	9
1.3. Tujuan Penelitian .....	10
1.4. Manfaat Penelitian .....	10
1.5. Batasan Masalah .....	11
1.6. Metode Penulisan.....	11
1.7. Sistematika Penulisan .....	12
BAB II .....	14
2.1. Sampah.....	14
2.2. Pengolahan Citra Digital.....	16
2.3. Visi Komputer.....	17
2.4. Machine Learning .....	18
2.4.1. <i>Training, Development, Testing Set</i> .....	19
2.5. Jaringan Saraf Tiruan.....	20



2.4.2.	<i>Convolution Neural Network</i> .....	24
2.6.	You Only Look Once v3 (YOLO) .....	31
2.7.	Mengukur Kinerja Algoritma klasifikasi dengan Confussion Matrix .....	40
BAB III	.....	44
3.1.	Tahapan Penelitian.....	44
3.2.	Waktu dan Lokasi Penelitian .....	46
3.3.	Instrumen Penelitian .....	46
3.4.	Teknik Pengambilan Data.....	47
3.5.	Perancangan Sistem .....	48
3.5.1.	Labelling Image .....	49
3.5.2.	Input Citra .....	51
3.5.3.	<i>Preprocessing (Resize)</i> .....	51
3.5.4.	<i>Extract Features</i> .....	53
3.5.5.	<i>Classification</i> .....	54
3.5.6.	<i>Output Prediction</i> .....	55
3.5.7.	<i>Backpropagation Weight Update</i> .....	57
3.5.8.	<i>Visualization</i> .....	57
3.6.	Analisis Kerja Sistem.....	58
3.6.1.	<i>Intersection over Union (IoU)</i> .....	58
3.6.2.	<i>Confussion Matrix</i> .....	59
BAB IV	.....	62
4.1.	Hasil Pembahasan .....	62
4.2.	Pembahasan.....	73
4.2.1.	<i>Batch</i> .....	73
4.2.2.	<i>Subdivision</i> .....	74

4.2.3.	<i>Width, Height dan Channel</i> .....	74
4.2.4.	Momentum.....	75
4.2.5.	<i>Learning Rate, Steps, dan Scale</i> .....	75
4.2.6.	<i>Data Augmentation (angle, saturation, exposure, hue)</i> .....	76
4.2.7.	<i>Max Batch (Iterations)</i> .....	76
BAB V	.....	79
5.1.	KESIMPULAN.....	79
5.2.	SARAN .....	81
DAFTAR PUSTAKA	.....	83
Lampiran 1.	Data Training Sistem .....	85
Lampiran 2.	Gambar aliran kanal untuk mengetahui letak tempat pengambilan data uji sistem.....	91
Lampiran 3.	File Konfigurasi Training YOLOv3.cfg.....	99
Lampiran 4.	Kode Program Pengujian Sistem .....	100

## DAFTAR GAMBAR

<b>Gambar 1.1.</b> Data Jenna R. Jambeck.....	3
<b>Gambar 2.1.</b> Genangan Sampah pada Kanal Jongaya (Muhammad Nur Abdurrahman, 2019) .....	15
<b>Gambar 2.2.</b> Single Layer Network.....	23
<b>Gambar 2.3.</b> Multi-Layer Network.....	24
<b>Gambar 2.4.</b> Contoh Filter Konvolusi Beserta Hasilnya.....	25
<b>Gambar 2.5.</b> ReLU .....	26
<b>Gambar 2.6.</b> Leaky ReLU (medium.com).....	28
<b>Gambar 2.7.</b> Contoh pooling (Jan Wira, 2019).....	29
<b>Gambar 2.8.</b> Convolution dan pooling (Jan Wira, 2019) .....	29
<b>Gambar 2. 9.</b> Convolutional Neural Network (Jan Wira, 2019).....	30
<b>Gambar 2.10 .</b> Fully Connected Layer .....	30
<b>Gambar 2. 11.</b> Arsitektur Network YOLOv3 (medium.com) .....	32
<b>Gambar 2. 12.</b> Model Layer Network Darknet-53 (medium.com).....	34
<b>Gambar 2.13.</b> Process Flow dari YOLOv3 (medium.com).....	37
<b>Gambar 2.14.</b> Perbandingan kinerja deteksi YOLO pada COCO Dataset (pjreddie.com, 2018) .....	37
<b>Gambar 2.15.</b> Ilustrasi proses deteksi YOLO (pjreddie.com, 2018).....	38
<b>Gambar 2.16.</b> Ilustrasi lapisan konvolusi YOLOv3 (Aditya, 2018).....	39

<b>Gambar 3.1.</b> Diagram Tahapan Penelitian .....	44
<b>Gambar 3.2.</b> Gambaran sistem deteksi .....	48
<b>Gambar 3.3.</b> Contoh gambar masukan .....	49
<b>Gambar 3.4.</b> Interface dari program YOLO Mark .....	50
<b>Gambar 3.5.</b> Contoh file anotasi dari program YOLO Mark .....	51
<b>Gambar 3. 6.</b> (a) Gambar sebelum di resize, (b) Gambar setelah di resize .....	52
<b>Gambar 3.7.</b> Convolution Layer .....	53
<b>Gambar 3. 8.</b> Max Pooling Layer (Brownlee, 2019).....	54
<b>Gambar 3. 9.</b> Fully Connected Layer .....	55
<b>Gambar 3.10.</b> Output dari Fully Connected Layer .....	56
<b>Gambar 3.11.</b> Visualisasi Output dari Sistem .....	58
<b>Gambar 3.12.</b> Gambaran Perhitungan IoU .....	59
<b>Gambar 3.13.</b> Gambaran IoU Pada Genangan Sampah .....	59
<b>Gambar 3.14.</b> Confussion Matrix .....	60
<b>Gambar 4.1.</b> Contoh gambar genangan sampah dan bukan sampah untuk data training .....	62
<b>Gambar 4.2.</b> Rincian dari proses training dengan menggunakan 4.000 iterasi .....	65

## DAFTAR TABEL

<b>Tabel 4.1.</b> Confusion Matriks pelatihan kinerja sistem dengan 4.000 iterasi .....	63
<b>Tabel 4.2.</b> Hasil perhitungan performance algoritma kinerja sistem .....	64
<b>Tabel 4.3.</b> Gambar hasil pengujian kinerja sistem .....	66
<b>Tabel 4. 4</b> Gambar dan rincian dari nilai size width & height yang berbeda-beda ....	69
<b>Tabel 4.5</b> Penjabaran rincian dari gambar pada Tabel 4.4 .....	71

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Kanal merupakan salah satu objek vital dalam konsepsi tata ruang kota. Urgensi pembuatan kanal seumumnya dimanapun, sesungguhnya merupakan upaya mengatasi banjir dengan memanfaatkan kelebaran dan kedalaman kanal. Di samping itu pembuatan kanal juga dimaksudkan agar berpotensi sebagai area penyaluran dan pembuangan air hujan kawasan setiap saat. Bila struktur kota dianalogikan dengan jaringan peredaran darah dalam tubuh, maka kanal adalah salah satu urat nadinya, sedangkan bila dianalogikan dengan sistem air buangan pada rumah tinggal, maka kanal adalah komponen primernya.

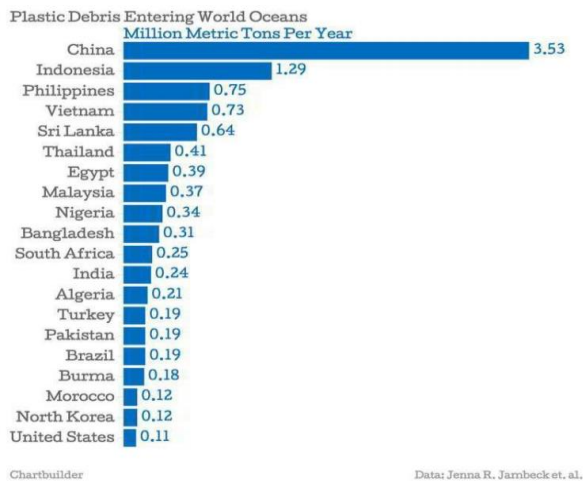
Sistem Kanal merupakan salah satu cara yang dianggap efektif dalam mengendalikan masalah banjir. Selain berfungsi sebagai pencegah banjir, kanal juga memiliki kegunaan lain seperti distribusi air, irigasi tanaman dan juga pariwisata, seperti yang dilakukan di berbagai negara. Salah satu rujukan daerah pemanfaatan kanal terbaik di dunia adalah di Venesia, Italia. Konsep kanal di Venesia menghubungkan antara kanal sebagai mitigasi bencana banjir dan aspek ekonomi pariwisata. Oleh karena itu, sistem kanal harus didesain sedemikian rupa agar menarik minat pengunjung, salah satu prasyaratnya adalah keindahan dan kebersihan kanal harus dijaga.

Di Kota Makassar terdapat jaringan kanal yang membelah wilayah kota yang membujur utara selatan yaitu Kanal Pannampu yang bermuara di Paotere serta Kanal Jongaya yang bermuara di Tanjungbunga. Kanal itu berakses ke timur pada Sungai Pampang dan Sungai Sinassara yang bermuara di Sungai Tallo. Kota Makassar memiliki tiga aliran kanal utama mengatur sistem panyaluran drainase untuk pembuangan, yakni, Kanal Panampu, Jongaya dan Sinrijala dengan panjang aliran masing-masing 40 km. Untuk kanal tersier 3.200 km. Dua diantaranya dapat dijadikan sebagai media transportasi air, yakni Kanal Jongaya dan Panampu. Sejak 1990, Kanal ini berfungsi sebagai drainase perkotaan dan basis utama pengendalian banjir. (Koran Fajar, 2004).

Kanal sebenarnya memiliki banyak fungsi yang sangat besar dalam membangun peradaban perkotaan. Kanal Banjir berfungsi sebagai pengendali banjir, namun sayangnya fungsi kanal di Kota Makassar yang panjangnya sekitar 34 kilometer (Km) ibarat kata pepatah 'jauh panggang dari api'. Kesadaran masyarakat yang masih kurang, terutama yang bermukim di bantaran kanal dan sungai dalam menjaga kebersihan lingkungan dengan membuang sampah di kanal, membuat kanal dipenuhi sampah sehingga pintu kanal terganggu padahal kanal dilengkapi dengan fasilitas berupa pintu penggelontoran, dimana biaya untuk membuat pintu penggelontoran sangat mahal, sangat disayangkan kalau tidak dirawat. Kanal tidak lagi sesuai fungsinya. Bahkan tak sedikit yang jadi tempat sampah. kumuh dan berbau.

Kondisi kanal semakin mendangkal akibat banyaknya lumpur dan tumpukan sampah plastik.

Salah satu permasalahan yang memiliki dampak buruk untuk segala aspek kehidupan adalah sampah yang berada di perairan. Menurut Kementerian Lingkungan Hidup dan Kehutanan (KLHK) Indonesia menilai persoalan sampah di perairan Indonesia sudah meresahkan. Hal itu seperti salah satu data Jambeck seperti pada Gambar 1.1 bahwa Indonesia berada di peringkat ke-2 penghasil sampah terbesar di dunia, dengan total sampah yang dibuang ke laut mencapai 129 juta ton pertahunnya.



**Gambar 1.1.** Data Jenna R. Jambeck

Namun hal ini tidak disadari oleh masyarakat bahwa produksi sampah setiap tahunnya mengalami kenaikan. Bahkan yang semula pada tahun 2005 tercatat hanya sebanyak 11% dari total sampah yang ada, meningkat menjadi 15% pada tahun 2015. Jenis sampah plastik merupakan penyumbang terbanyak sampah ke laut. Setiap tahun produksi plastik menghasilkan sekitar delapan



persen hasil produksi minyak dunia atau sekitar 12 juta barel minyak atau setara 14 juta pohon. Lebih dari satu juta kantong plastik digunakan setiap menitnya, dan 50 persen dari kantong plastik tersebut dipakai hanya sekali lalu langsung dibuang. (Wahyuni, 2016).

Banyaknya limbah sampah perairan di Indonesia mengakibatkan terganggunya ekosistem perairan secara signifikan. Hal ini dikuatkan dengan laporan dari Pusat Penelitian Terumbu Karang Australia bahwa terumbu yang terpapar limbah perairan berpotensi 89% terkena penyakit dibandingkan 4% yang tidak terkena dampak limbah. Teori tersebut dibuktikan sekelompok peneliti asal Indonesia, Amerika Serikat, Australia dan Kanada yang mengamati kondisi 159 terumbu karang antara 2011-2014. Hasilnya paparan limbah perairan pada terumbu karang paling banyak muncul di Indonesia, yakni 26 bagian per 100 meter persegi. (WD, 2018).

Di kota Makassar sendiri masalah limbah sampah perairan khususnya di kanal masih menyimpan bom waktu. Masalah sampah yang menumpuk di kanal menjadi ancaman banjir dan kesehatan masyarakat yang hidup di sekitar kanal.

Seperti kasus yang terjadi di Kanal Rajawali (Radhif, 2019), hampir setiap waktu masyarakat harus dibiasakan mencium aroma bau busuk yang keluar dari kanal. Kondisi ini diperparah dengan adanya tumpukan sampah, yang tertahan di depan pelelangan ikan (Lelong) Rajawali. Hal ini disinyalir

masyarakat sekitar akibat dari posisi kanal Rajawali sebagai pembuangan terakhir dari kanal-kanal di Makassar. Banyak dari masyarakat panambungan mengeluhkan pencemaran ini. Apalagi jarak rumah warga yang berhadapan dengan kanal. Hal ini tentu mengganggu aktivitas keseharian warga, apalagi sebagian besar penduduknya menggantungkan hidupnya dari hasil melaut. Mereka harus mengarungi kanal untuk dapat keluar melaut.

Kasus serupa juga terlihat di Kanal Jongaya yang melintasi Jalan Nuri Baru dan Jalan Tanjung Alang, Kecamatan Mariso (Muhammad Nur Abdurrahman, 2019). Kawasan permukiman perkotaan yang berada di pinggiran Kanal Jongaya sebagai salah satu penyebab tercemarnya lingkungan abiotik seperti penumpukan sampah yang tinggi serta kualitas air yang buruk yang berdampak pada tercemarnya Pantai Losari sebagai salah-satu landmark Kota Makassar. Kanal Jongaya dipenuhi sampah, selain dipenuhi sampah, air di kanal itu pun tampak berwarna hitam. Kanal yang airnya berwarna hitam pekat ini, terlihat hamparan sampah plastik dan sisa makanan yang mengapung di antara perahu-perahu nelayan. Kanal Jongaya terkoneksi menuju kanal Jalan Metro Tanjung Bunga dan Pantai Losari. Selain jorok dipandang, sampah-sampah ini mengeluarkan aroma tidak sedap yang mengganggu warga di bantaran kanal depan kampus Akademi Maritim Indonesia (AMI) Veteran.

Bila ditelusuri, potret buram di sepanjang jaringan kanal di Kota Makassar, diakibatkan antara lain, tak lepas dari perilaku destruktif masyarakat

dan rute pengairan kanal serta perlakuan yang rendah terhadap fisik kanal. Perlakuan dimaksud adalah kebijakan Pemerintah Kota yang cenderung menomorsekiankan pengelolaan kanal serta sikap hidup yang menyimpang dari sebagian masyarakat terhadap kanal-kanal yang ada seperti menjadikan kanal sebagai lahan pembuangan limbah baik cair berlebih yang padat.

Kanal seringkali hanya 'dilirik' bila musim hujan menjelang tiba, dan cenderung terabaikan bahkan terasa tiada saat musim kemarau. Saat musim hujan, debit-debit air memenuhi kanal ternyata tidak mengalir lancar melainkan kubangan panjang. Sampah-sampah plastik, dan potongan-potongan kayu mengapung. Jika volume air berkurang, pasir dan tanah yang bercampur air menumpuk dan mengendap di dasar-dasar kanal. Pelan-pelan membuat lapisan tebal.

Permasalahan kanal di kota Makassar, tak sesempit masalah bau semata. Kanal juga merupakan satu-satunya akses nelayan yang hendak melaut. Belum lagi masalah kesehatan yang berpotensi muncul akibat pencemaran ini. Seperti diare, demam berdarah, dan lain sebagainya. Hal ini memerlukan perhatian pemerintah dan masyarakat umum.

Berbagai upaya dilakukan pemerintah dengan memberi perhatian khusus kepada rute pengairan dan kondisi kebersihan kanal, untuk membersihkan saluran kanal dari sampah-sampah yang menjadi penyebab banjir. Pada awal tahun 2019 pemerintah kota Makassar membentuk Satgas Drainase yang terdiri

dari Dinas PU Makassar dan kantor kecamatan setempat. Tugasnya membersihkan saluran kanal dari sampah-sampah yang menjadi penyebab banjir. Gerakan aksi bersih-bersih kanal di sejumlah kanal di Makassar, seperti Kanal Jongaya, Kanal Sinrijala, Kanal Pabaeng-baeng, dan Kanal Pannampu. Petugas dari Pemkot Makassar bersama masyarakat melakukan aksi bersih-bersih di kanal yang melintasi dua kecamatan, Kecamatan Mariso dan Kecamatan Tamalate ini. (Dinas Komunikasi dan Informatika, 2019).

Meskipun upaya telah dilakukan oleh pemerintah kota Makassar, namun fakta di lapangan berbicara bahwa sebagian besar dari luasan jaringan kanal yang ada di kota Makassar mengalami pendangkalan sungai akibat sedimentasi beragam limbah, tumpukan sampah. Hal demikian menyebabkan menurunnya kapasitas penyaluran dan pembuangan debit air, sehingga akibat yang paling mungkin dan sangat mudah diprediksi adalah terjadinya pelambatan aliran air.

Karena fungsi kanal yang berubah menjadi tempat pembuangan sampah oleh masyarakat, membuat kanal tidak dapat berfungsi secara optimal, maka saat curah hujan meninggi pada musim hujan dengan mudah terjadi genangan air dan banjir. Selain itu jika dilihat dari aspek lingkungan, sistem kanal dianggap tidak memperhatikan keberlangsungan lingkungan dan cenderung merusak ekosistem alamiah sungai. Akibat lain yang bisa ditimbulkan oleh kondisi kanal seperti itu (bila terdapat genangan air dan gundukan limbah) adalah kemungkinan menurunnya kesehatan lingkungan.

Sesering apapun upaya yang dilakukan oleh pemerintah kota Makassar, bila perilaku masyarakat cenderung tidak berubah, maka permasalahan sampah di kanal akan terus menjadi pekerjaan berat bagi pemerintah. Oleh karena itu dibutuhkan hadirnya teknologi sebagai salah satu pilihan penyelesaian masalah dalam mengurangi jumlah sampah pada perairan secara komprehensif demi menjaga ekosistem perairan dengan seimbang. Salah satu teknologi yang dapat digunakan untuk mendeteksi genangan sampah pada permukaan air kanal adalah metode *deep learning*.

*Deep Learning* adalah salah satu contoh bidang *Machine Learning* yang menggunakan jaringan syaraf tiruan pada penerapan permasalahan dengan dataset yang besar (Panella et al. 2018; L. Zhang et al., 2017).

Teknik *Deep Learning* sangat mendukung proses *Supervised Learning* pada komputer dikarenakan pengenalan data set yang sangat kuat sehingga mampu menjadi pondasi dalam proses klasifikasi (Socher et al., n.d.; Xin et al. 2018).

*Neural network* sendiri adalah model yang terinspirasi oleh bagaimana *neuron* dalam otak manusia bekerja. Tiap *neuron* pada otak manusia saling berhubungan dan informasi mengalir dari setiap *neuron* tersebut. *Convolutional Neural Network* merupakan salah satu jenis *neural network* yang juga merupakan bagian dari *Deep Learning*. (X. Zhang et al. 2018; Sharma et al., 2017). Metode ini sangat efektif dan lebih mudah dalam mengidentifikasi pola

dari data yang dimasukkan. Pada umumnya *Convolutional Neural Network* digunakan pada pengolahan data yang berupa image atau data lainnya (Krizhevsky, Sutskever, and Hinton 2017; Marifatul Azizah, Fadillah Umayah, and Fajar 2018; Darmanto and Probolinggo, 2019).

*Convolutional neural network* ini juga diklaim sebagai algoritma terbaik dan paling banyak digunakan untuk mendeteksi objek dari data citra digital (Eka Putra 2016). Metode ini juga sangat berdampak pada kemajuan perkembangan yang telah dicapai *Artificial Intelligence* secara bertahap. Tidak hanya untuk perangkat lunak, namun para penggunanya juga telah merambah diberbagai bidang industry. (Aprilia Safitri, 2018).

Berdasarkan latar belakang yang telah dikemukakan di atas maka peneliti tertarik untuk melakukan penelitian tentang sistem deteksi genangan sampah pada permukaan air kanal dengan menggunakan metode *deep learning*.

## **1.2. Rumusan Masalah**

Rumusan masalah pada tugas akhir ini adalah:

- a. Bagaimana cara mendeteksi genangan sampah pada permukaan air kanal dengan menggunakan metode *deep learning*?
- b. Bagaimana hasil pengujian sistem pendeteksian genangan sampah pada permukaan air kanal dengan menggunakan metode *deep learning*?

### **1.3. Tujuan Penelitian**

Tujuan penelitian pada tugas akhir ini adalah:

- a. Untuk membuat sistem yang dapat mendeteksi genangan sampah pada permukaan air kanal dengan menggunakan metode *deep learning*.
- b. Untuk mengetahui hasil pengujian sistem pendeteksian genangan sampah pada permukaan air kanal dengan menggunakan metode *deep learning*.

### **1.4. Manfaat Penelitian**

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan adalah:

- a. Bagi masyarakat, penelitian ini dapat digunakan sebagai alat bantu untuk mengurangi jumlah sampah di perairan secara komprehensif, mengurangi tingkat kerusakan ekosistem air akibat sampah, dan meningkatkan kualitas air untuk kebutuhan hidup makhluk hidup.
- b. Bagi peneliti, penelitian ini dapat digunakan untuk menambah pengetahuan dan kemampuan di bidang *deep learning* dalam mendeteksi objek-objek yang berada di permukaan perairan khususnya kanal.
- c. Bagi institusi pendidikan, penelitian ini dapat digunakan sebagai referensi ilmiah untuk penelitian - penelitian selanjutnya.

## 1.5. Batasan Masalah

Adapun yang menjadi batasan masalah dalam penelitian ini adalah:

- a. Objek penelitian berupa genangan sampah dan benda bukan sampah yang berada di permukaan air kanal.
- b. Pengambilan data dilakukan di permukaan air kanal jongaya menggunakan *drone* dalam format video.
- c. Rancangan model pendeteksi dibuat menggunakan *deep learning* dengan metode *YOLO Mark* untuk pengenalan objek serta *drone* untuk pengambilan dan verifikasi data.

## 1.6. Metode Penulisan

Terdapat beberapa metode yang digunakan pada penelitian ini yaitu:

### 1. Metode Pengambilan Data

Metode pengambilan data dilakukan dengan mengumpulkan data yang diambil secara langsung menggunakan *drone* dan situs penyedia video seperti Youtube.

### 2. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan informasi yang terkait dengan penelitian ini dari berbagai sumber seperti buku, internet, dan sumber lainnya.

### 3. Diskusi dan Konsultasi



Diskusi dan konsultasi dilakukan dengan mengadakan tanya jawab secara langsung kepada dosen pembimbing dan pihak-pihak profesional lainnya yang berhubungan dengan penelitian ini.

### **1.7. 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 bahasan masalah, tujuan, manfaat, dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan sampah, visi komputer, pemrosesan citra, *machine learning*, *deep learning*, dan metode yang digunakan.

#### **BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang tahapan penelitian, instrumen penelitian, dan penerapan algoritma serta teknik pengolahan data.

#### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

## BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Sampah**

Sampah merupakan material sisa yang tidak diinginkan setelah berakhirnya suatu proses. Sampah didefinisikan oleh manusia menurut derajat keterpakaiannya, dalam proses - proses alam sebenarnya tidak ada konsep sampah, yang ada hanya produk - produk yang dihasilkan setelah dan selama proses alam tersebut berlangsung. Akan tetapi karena dalam kehidupan manusia didefinisikan konsep lingkungan maka sampah dapat dibagi menurut jenis-jenisnya.

Menurut (Billy Adytya, 2020), jenis – jenis sampah terbagi menjadi tiga, yaitu:

1. Berdasarkan sumbernya: Sampah Alam, Sampah Manusia, Sampah Konsumsi, Sampah Nuklir, Sampah Industri dan Sampah Pertambangan.
2. Berdasarkan sifatnya:
  - Sampah organik - dapat diurai (*degradable*)

Sampah Organik, yaitu sampah yang mudah membusuk seperti sisa makanan, sayuran, daun-daun kering, dan sebagainya. Sampah ini dapat diolah lebih lanjut menjadi kompos. Contohnya: Daun, kayu, kulit telur, bangkai hewan, bangkai tumbuhan, kotoran hewan dan manusia, Sisa makanan, Sisa manusia. kardus, kertas dan lain-lain.

- Sampah anorganik - tidak terurai (*undegradable*)

Sampah Anorganik, yaitu sampah yang tidak mudah membusuk, seperti plastik wadah pembungkus makanan, kertas, plastik mainan, botol dan gelas minuman, kaleng, kayu, dan sebagainya. Sampah ini dapat dijadikan sampah komersial atau sampah yang laku dijual untuk dijadikan produk lainnya. Beberapa sampah anorganik yang dapat dijual adalah plastik wadah pembungkus makanan, botol dan gelas bekas minuman, kaleng, kaca, dan kertas, baik kertas koran, HVS, maupun karton.

- beracun (B3): limbah dari bahan-bahan berbahaya dan beracun seperti limbah rumah sakit, limbah pabrik dan lain-lain.

### 3. Berdasarkan bentuknya: Sampah Padat dan Sampah Cair.



**Gambar 2.1.** Genangan Sampah pada Kanal Jongaya (Muhammad Nur Abdurrahman, 2019)

## 2.2. 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 1280x720 piksel. Dengan pengolahan citra digital, gambar tersebut dapat diubah ukurannya menjadi 640x360 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 didalam 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 di manipulasi 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.

### **2.3. Visi Komputer**

Visi komputer merupakan ilmu pengetahuan yang mempelajari bagaimana cara membuat komputer dapat mengamati dan mengenali objek. Visi komputer bertujuan untuk meniru visualisasi manusia dan mengaplikasikan ke dalam komputer (Nugraha 2014).

Visi komputer bekerja dengan memproses data berupa citra yang dimasukkan lalu menggunakan kombinasi dari algoritma - algoritma pengolahan citra dan kecerdasan buatan sehingga menghasilkan informasi dari citra tersebut (Szeliski 2010).

Visi komputer diaplikasikan saat ini dalam berbagai bidang di dunia, meliputi:

1. *Optical Character Recognition* (OCR): mengidentifikasi huruf pada sebuah citra lalu mengubahnya ke dalam bentuk tulisan.

2. *Machine inspection*: menginspeksi komponen mesin atau rangka untuk *quality assurance* menggunakan citra sinar-X.
3. *3D Model Building*: Konstruksi otomatis model 3D dari foto udara yang digunakan dalam sistem seperti *Bing Maps*.
4. *Medical Imaging*: Mencatat atau merekam citra *pre-operative* dan *intra-operative* untuk tujuan klinis atau melakukan studi jangka panjang dari morfologi citra bagian tubuh manusia.
5. *Automotive Safety*: Mendeteksi rintangan yang tak terduga seperti pejalan kaki di jalan raya.
6. *Surveillance*: Pemantauan untuk penyusup dan menganalisis lalu lintas jalan raya.
7. *Fingerprint Recognition and Biometrics*: Otentikasi akses otomatis dan juga aplikasi forensik (Szeliski 2010).

#### **2.4. Machine Learning**

*Machine learning*, yaitu teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti *machine learning* adalah untuk membuat model (matematis) yang merefleksikan pola-pola data. (Jan Wira, 2019)

*Machine learning* sama-sama melakukan inferensi, tetapi pada representasi yang berbeda. Inferensi pada bidang keilmuan representasi pengetahuan mencakup tentang bagaimana cara (Langkah dan proses)

mendapatkan sebuah keputusan, diberikan premis. pada *machine learning*, inferensi yang dimaksud lebih ke ranah hubungan variabel.

#### **2.4.1. Training, Development, Testing Set**

Terdapat dua istilah penting dalam pembangunan model *machine learning* yaitu: *training* dan *testing*. *Training* adalah proses membangun model dan *testing* adalah proses menguji kinerja model pembelajaran. *Dataset* adalah kumpulan data (sampel dalam statistik). Sampel ini adalah data yang digunakan untuk membuat model maupun mengevaluasi model *machine learning*. Umumnya, *dataset* dibagi menjadi tiga jenis yang tidak beririsan (satu sampel pada himpunan tertentu tidak muncul pada himpunan lainnya):

1. *Training set* adalah himpunan data yang digunakan untuk melatih atau membangun model.
2. *Development set* atau *validation set* adalah himpunan data yang digunakan untuk mengoptimisasi saat melatih model. Model dilatih menggunakan *training set* dan pada umumnya kinerja saat latihan diuji dengan *development set*. Hal ini berguna untuk generalisasi (agar model mampu mengenali pola secara generik).
3. *Testing set* adalah himpunan data yang digunakan untuk menguji model setelah proses latihan selesai. *Testing set* adalah *unseen data*. Artinya, model dan manusia tidak boleh melihat sampel



ini saat proses latihan. Banyak orang yang tergoda untuk melihat *testing set* saat proses latihan walaupun itu adalah tingkah laku yang buruk karena menyebabkan bias.

## 2.5. Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan teknik pembelajaran mesin yang menyimulasikan mekanisme pembelajaran pada organisme biologis. Sistem saraf manusia mengandung sel, yang disebut sebagai *neuron*. Mekanisme biologis ini disimulasikan dalam jaringan saraf tiruan yang berisi unit perhitungan yang juga disebut *neuron*. Unit komputasi (*neuron*) tersebut terhubung satu sama lain melalui bobot, yang berperan seperti koneksi sinaptik dalam organisme biologis. Setiap masukan ke *neuron* diskalakan dengan bobot, yang mempengaruhi nilai fungsi yang dihitung pada unit tersebut.

Jaringan saraf tiruan menghitung fungsi masukan dengan menyebarkan nilai yang dihitung dari *neuron* masukan ke *neuron* keluaran dan menggunakan bobot sebagai parameter perantara. Pembelajaran terjadi dengan mengubah bobot yang menghubungkan *neuron*. Sama seperti rangsangan eksternal yang diperlukan untuk belajar dalam organisme biologis, rangsangan eksternal dalam jaringan saraf tiruan berupa data pelatihan yang berisi pasangan masukan dan keluaran dari fungsi yang harus dipelajari. Pasangan data pelatihan ini dimasukkan ke dalam jaringan saraf tiruan dengan menggunakan representasi

masukan untuk membuat prediksi tentang label keluaran. Data pelatihan memberikan umpan balik terhadap nilai bobot dalam jaringan saraf tiruan berdasarkan pada seberapa baik keluaran yang dihasilkan untuk masukan tertentu. [1]

Haykin menyatakan bahwa “Jaringan saraf adalah prosesor terdistribusi paralel besar-besaran yang memiliki kecenderungan alami untuk menyimpan pengetahuan pengalaman dan membuatnya tersedia untuk digunakan. Ini menyerupai otak dalam dua hal: 1. Pengetahuan diperoleh oleh jaringan melalui proses pembelajaran; 2. Kekuatan interkoneksi antar *neuron*, yang dikenal sebagai bobot sinaptik atau bobot, digunakan untuk menyimpan pengetahuan” [2].

Berdasarkan cara belajarnya, jaringan saraf tiruan terbagi atas dua, yaitu *Supervised Learning* dan *Unsupervised Learning* [3].

#### 1. *Supervised Learning*

*Supervised learning* membuat sebuah model yang membuat prediksi dari data latih yang telah disiapkan dengan jawabannya, misalkan sebuah set gambar binatang lengkap dengan nama dari binatang-binatangnya. Tujuan pembelajaran ini yaitu membuat model yang dapat mengidentifikasi gambar (yang berisi binatang yang dipakai saat pelatihan) yang tidak pernah dilihat sebelumnya secara tepat.

## 2. *Unsupervised Learning*

*Unsupervised learning* menemukan pola atau struktur intrinsik tersembunyi dalam data. Teknik ini digunakan untuk menarik sebuah kesimpulan dari *dataset* yang terdiri dari data tanpa label. *Clustering* merupakan teknik *unsupervised learning* yang paling umum.

*Clustering* digunakan untuk analisis data eksplorasi untuk menemukan pola atau pengelompokan tersembunyi dalam data. Aplikasi *clustering* diantaranya analisi urutan gen, riset pasar, dan pengenalan objek.

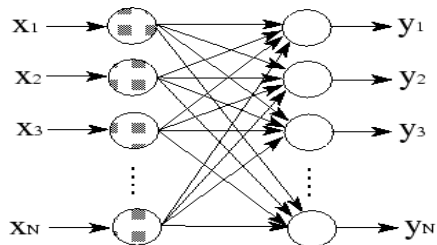
Algoritma yang umum untuk *clustering* diantaranya *k-means* dan *k-medoids*, *hierarchical clustering*, *Gaussian mixture models*, *hidden Markov models*, *self-organizing maps*, *fuzzy c-means clustering*, dan *subtractive clustering*.

Sedangkan 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 2.2** menunjukkan arsitektur dari *single layer network*.

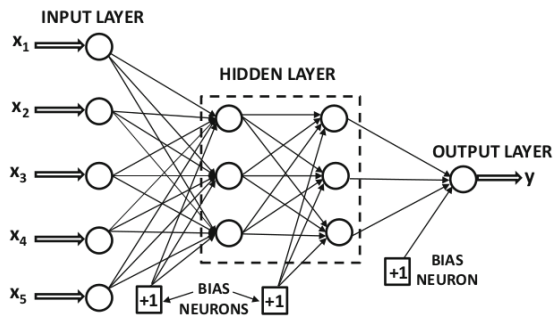


**Gambar 2.2.** *Single Layer Network*

## 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 2.3** menunjukkan arsitektur *multi layer network*.



**Gambar 2.3.** *Multi-Layer Network*

#### 2.4.2. *Convolution Neural Network*

Pada bidang *neural network*, *convolutional neural network* (CNN) adalah kategori utama untuk melakukan pengenalan dan pengklasifikasian gambar. CNN mirip dengan *feedforward neural network*, dimana *neuron* memiliki nilai bobot dan bias yang mampu untuk belajar.








Klasifikasi gambar CNN mengambil sebuah gambar masukan, memrosesnya, dan mengklasifikasikannya ke dalam sebuah kategori. Komputer melihat gambar masukan sebagai sebuah larik piksel. Secara teknis, setiap gambar masukan akan melalui deretan *convolution layer* dengan *filter (Kernels)*, *Pooling*, *fully connected layers (FC)*, dan menggunakan fungsi Softmax untuk mengklasifikasikan sebuah objek dengan nilai probabilitas di antara 0 dan 1.

##### 1. *Convolution Layer*

*Convolution layer* merupakan layer pertama untuk mengekstraksi fitur dari sebuah gambar masukan. *Convolution*

*layer* mempertahankan hubungan antar piksel dengan mempelajari fitur dari gambar menggunakan kotak kecil dari gambar masukan. Proses ini merupakan operasi matematik yang hanya membutuhkan dua masukan yaitu matriks gambar dan *filter* atau *kernal*.

Proses konvolusi dengan *filter* berbeda-beda dapat melakukan operasi seperti deteksi tepi serta pengaturan keburaman dan ketajaman. **Gambar 2.4** merupakan contoh hasil dari proses konvolusi beserta *filter*-nya.

Operation	Filter	Convolved Image
<b>Identity</b>	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
<b>Edge detection</b>	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
<b>Sharpen</b>	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
<b>Box blur</b> (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
<b>Gaussian blur</b> (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

**Gambar 2.4.** Contoh Filter Konvolusi Beserta Hasilnya

## 2. *Stride*

*Stride* adalah jumlah piksel yang bergeser di atas matriks masukan. Ketika *stride* bernilai 1 maka *filter* akan berpindah 1 piksel tiap waktu. Ketika *stride* bernilai 2 maka *filter* akan berpindah 2 piksel tiap waktu dan seterusnya.

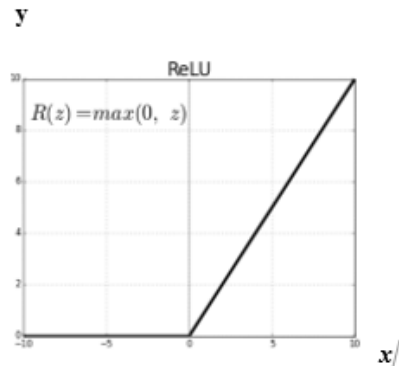
## 3. *Padding*

*Padding* digunakan ketika ukuran *filter* tidak pas dengan gambar masukan. Ada 2 yang macam *padding* yang biasa digunakan, yaitu:

- *Pad* citra dengan nilai nol sehingga pas.
- Hilangkan bagian citra yang tidak pas dengan *filter*.

## 4. 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 2.5**.

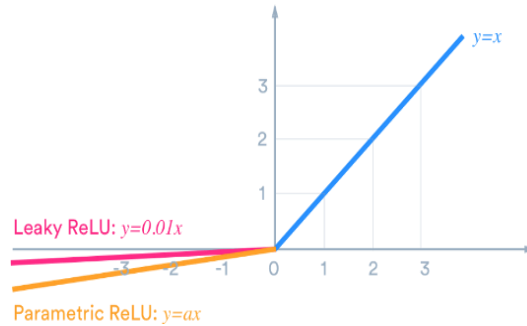


**Gambar 2.5.** *ReLU*

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

Selain itu, dengan mengubah data negatif menjadi 0, maka *neuron*-neuran 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 2.6.** menunjukkan grafik dari Leaky ReLU (Sharma, 2019).

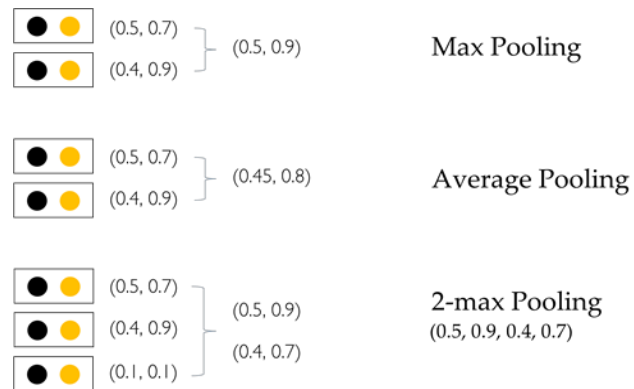




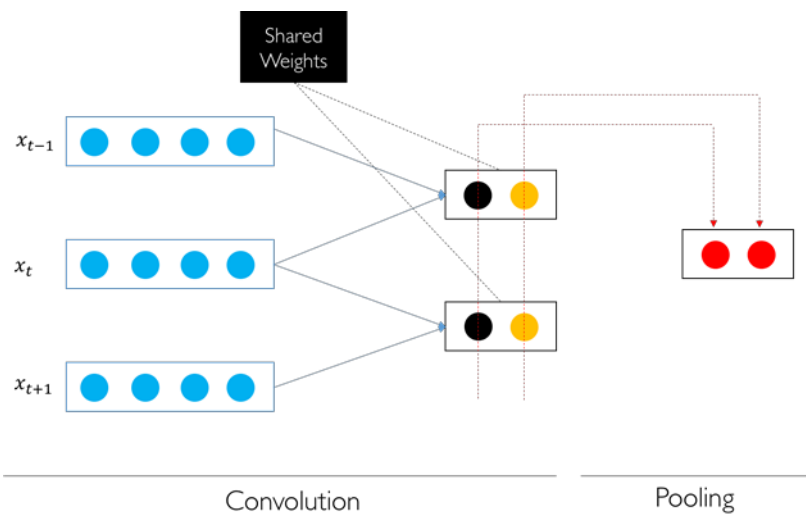
**Gambar 2.6.** *Leaky ReLU* (medium.com)

## 5. Pooling Layer

Pada tahap *convolution*, Setiap *k-sized window* diubah menjadi satu vektor berdimensi  $d$  (yang dapat disusun menjadi matriks  $D$ ). Semua vektor yang dihasilkan pada tahap sebelumnya dikombinasikan (*pooled*) menjadi satu vektor  $c$ . Ide utamanya adalah mengekstrak informasi paling informatif (semacam meringkas). Ada beberapa teknik *pooling*, diantaranya: *max pooling*, *average pooling*, dan *K-max pooling* 3; diilustrasikan pada **Gambar 2.7.** *Max pooling* mencari nilai maksimum untuk setiap dimensi vektor. *Average pooling* mencari nilai rata-rata tiap dimensi. *K-max pooling* mencari  $K$  nilai terbesar untuk setiap dimensinya (kemudian hasilnya digabungkan). Gabungan operasi *convolution* dan *pooling* secara konseptual diilustrasikan pada **Gambar 2.8**



**Gambar 2.7.** Contoh *pooling* (Jan Wira, 2019)

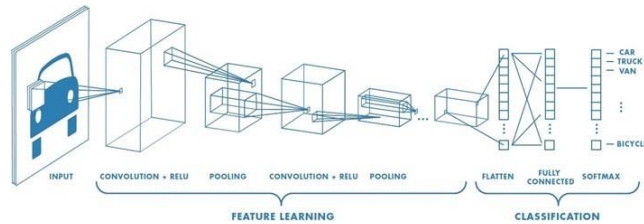


**Gambar 2.8.** *Convolution dan pooling* (Jan Wira, 2019)

Setelah melewati berbagai operasi *convolution* dan *pooling*, vektor kemudian dilewatkan pada *multilayer perceptron (fully connected)* untuk melakukan sesuatu (tergantung permasalahan),

misal klasifikasi gambar, klasifikasi sentimen, dsb (Ilustrasi pada

**Gambar 2.9)**

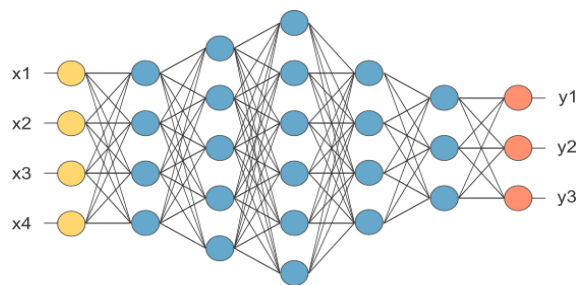


**Gambar 2.9.** *Convolutional Neural Network* (Jan Wira, 2019)

## 6. *Fully Connected Layer*

Matriks *feature map* akan dikonversi menjadi vektor, lalu *fully connected layer* akan mengkombinasikan fitur tersebut secara bersamaan untuk membuat sebuah model. Lalu dengan menggunakan fungsi aktivasi, citra keluaran akan diklasifikasikan sesuai dengan kelasnya. Gambaran dari FC Layer dapat dilihat pada

**Gambar 2.10.**



**Gambar 2.10 .** *Fully Connected Layer*

## 2.6. *You Only Look Once v3 (YOLO)*

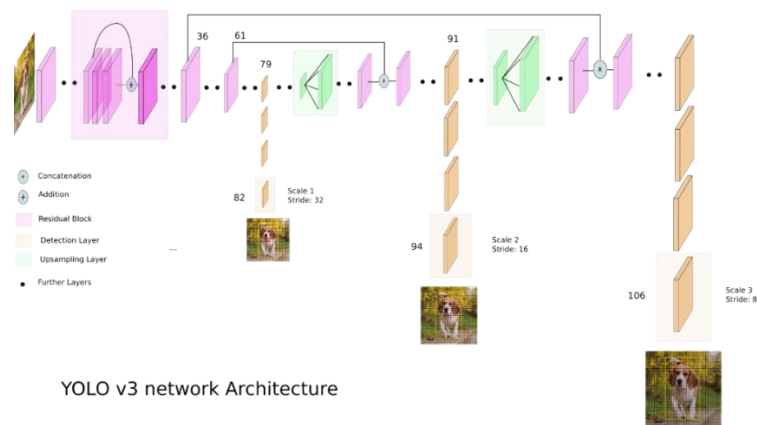
Deteksi objek merupakan permasalahan yang jauh lebih kompleks dibandingkan proses klasifikasi, di mana sistem tidak hanya dapat mengenali objek, akan tetapi juga menunjukkan posisi di bagian mana objek tersebut berada, dan sistem tersebut tidak akan bekerja pada gambar dengan lebih dari satu objek.

*You Only Look Once*, atau biasa disingkat YOLO, adalah salah satu algoritma *deep learning* deteksi objek tercepat saat ini. Jika biasanya untuk melakukan proses klasifikasi objek, algoritma yang dibuat adalah dengan menggunakan algoritma klasifikasi seperti *VGGNet* atau *Inception* yang dijalankan pada setiap bagian kecil dari citra menggunakan *sliding window*. Proses ini akan berhasil mendeteksi objek akan tetapi sistem akan sangat lambat, karena algoritma klasifikasi tersebut akan dijalankan berulang-ulang kali.

Algoritma YOLO melakukan pendekatan yang sangat berbeda dengan algoritma lain. Algoritma ini akan melakukan *scanning* pada citra hanya sekali dengan cara yang efisien. YOLO menggunakan *convolutional neural networks* untuk mendeteksi objek. Meskipun bukan algoritma yang terakurat, akan tetapi algoritma ini merupakan pilihan yang baik untuk melakukan deteksi objek secara real-time, 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, akan tetapi juga bisa mendeteksi berbagai objek yang berada didalam 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 pada **Gambar 2.11**.



**Gambar 2. 11.** Arsitektur *Network* YOLOv3 (medium.com)

Pada *paper* YOLOv3, Joseph Redmon sebagai penulis *paper* memperkenalkan arsitektur baru yang memiliki *network* lebih kompleks 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 et al. 2015)

Jika 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 down sampling secara kasar. Untuk mengatasi hal ini, YOLOv2 menggunakan *identity mapping*, fitur map gabungan dari layer sebelumnya untuk mendapatkan *low level features*. Meskipun begitu, 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 Darknet-53 dapat dilihat pada **Gambar 2.12**.

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

Darknet-53 model

**Gambar 2. 12.** 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* di mana outputnya di *generate* dengan cara mengaplikasikan filter berupa kernel ke dalam fitur map. Dalam YOLOv3, deteksi dilakukan dengan cara mengaplikasikan kernel pendeteksi berukuran 1x1 pada fitur map pada 3 ukuran yang berbeda dan 3 lokasi yang berbeda pada *network*. (Redmon et al. 2015)

Terdapat beberapa parameter yang terdapat pada setiap *layer convolutional*, antara lain *filters*, *size*, *stride*, dan *pad*. *Filters* adalah jumlah dari

filter yang terdapat pada layer tersebut. Filter tersebut dapat berupa matriks kernel dengan nilai-nilai tertentu yang berfungsi mengekstraksi fitur - fitur inputan. Kemudian *size* merupakan ukuran dari kernel filter tersebut, misalnya dengan value 3, maka filturnya berukuran 3x3. *Stride* merupakan parameter yang mengatur bagaimana filter bergerak pada citra, misalnya jika nilai *stride* 1 maka kernel akan bergerak 1 pixel/unit pada satu waktu. Dan terakhir adalah pad, untuk menambahkan padding pada bagian tepi input citra atau fitur.

Ukuran dari kernel pendeteksi adalah  $1 \times 1 \times (B \times (5+C))$ . Disini B adalah jumlah dari *bounding boxes* didalam 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 ada diatas.

Sebagai contoh, input citra memiliki ukuran 416x416 piksel. YOLOv3 melakukan proses deteksi objek pada 3 skala berbeda dengan cara downsampling 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 didownsampling oleh *network*, dengan *stride* 32. Jika input citra berukuran 416x416, fitur map hasil dari *network* memiliki ukuran 13x13. Proses pendeteksian pertama akan dilakukan dengan menggunakan filter



pendeteksi berukuran  $1 \times 1$ , sehingga memberikan fitur map dengan ukuran  $13 \times 13 \times 255$ .

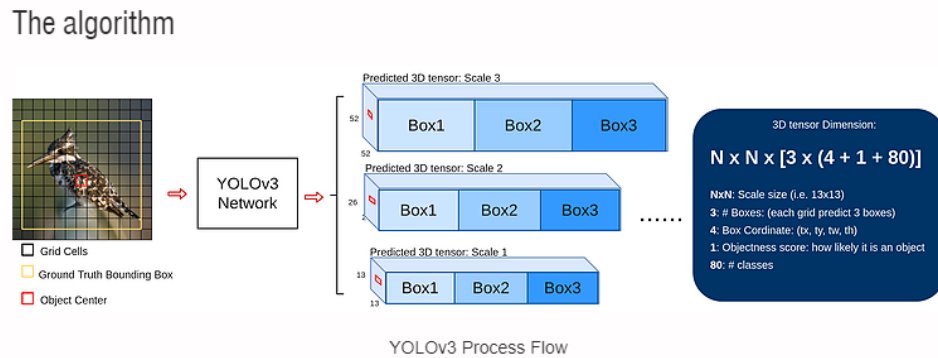
Kemudian, fitur map dari layer 79 akan ditujukan ke beberapa layer konvolusi sebelum di upsampling ke dimensi berukuran  $26 \times 26$ . Fitur map ini memiliki hubungan dengan fitur map pada layer 61. Kemudian fitur map yang ada akan diteruskan ke beberapa convolutional layer untuk digabungkan dengan 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  $13 \times 13$  bertanggung jawab dalam pendeteksian objek berukuran besar, dimana layer  $52 \times 52$  mendeteksi objek yang kecil,

sementara layer 26x26 mendeteksi objek dengan ukuran sedang. Process Flow dari proses deteksi objek dapat dilihat pada **Gambar 2.13**.



**Gambar 2.13.** Process Flow dari YOLOv3 (medium.com)

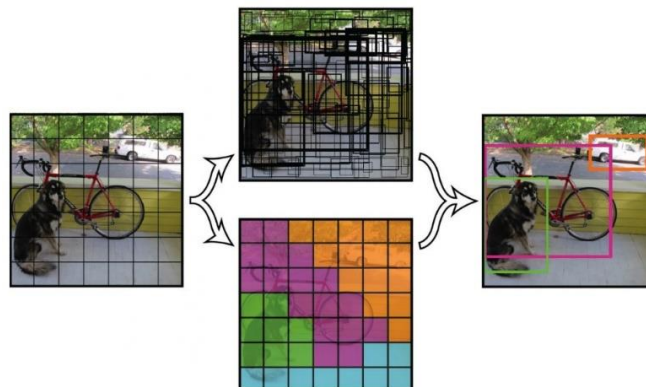
Untuk tingkat akurasi, algoritma YOLO dapat bersaing dengan algoritma deteksi objek lain yang sudah ada. Berikut adalah **Gambar 2.14** yang menunjukkan perbandingan kinerja deteksi YOLO pada *COCO Dataset* dibandingkan dengan algoritma lain. (pjreddie.com, 2018).

Performance on the COCO Dataset						
Model	Train	Test	mAP	FLOPS	FPS	Cfg Weights
SSD300	COCO trainval	test-dev	41.2	-	46	link
SSD500	COCO trainval	test-dev	46.5	-	19	link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg weights
SSD321	COCO trainval	test-dev	45.4	-	16	link
DSSD321	COCO trainval	test-dev	46.1	-	12	link
R-FCN	COCO trainval	test-dev	51.9	-	12	link
SSD513	COCO trainval	test-dev	50.4	-	8	link
DSSD513	COCO trainval	test-dev	53.3	-	6	link
FPN FRCN	COCO trainval	test-dev	59.1	-	6	link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14	link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11	link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5	link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg weights

**Gambar 2.14.** Perbandingan kinerja deteksi YOLO pada COCO Dataset (pjreddie.com, 2018)

Pada **Gambar 2.14** dapat terlihat bahwa mAP (*mean Average precision*) pada YOLO lebih tinggi dibandingkan algoritma deteksi lain, artinya akurasi yang akan didapat kan jika menggunakan YOLO akan lebih baik jika dibandingkan dengan algoritma lain. (pjredie.com, 2018).

Kebanyakan sistem deteksi sebelumnya menggunakan pengklasifikasian atau *localizer* untuk melakukan deteksi dengan menerapkan model ke gambar di beberapa lokasi dan memberi nilai *confident* pada gambar sebagai bahan untuk pendeteksian. YOLO menggunakan pendekatan yang sangat berbeda dengan algoritma sebelumnya, yakni menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasian sebagai objek atau bukan. Ilustrasi dari proses pendeteksian YOLO dapat dilihat pada **Gambar 2.15**. (Aditya, 2018).



**Gambar 2.15.** Ilustrasi proses deteksi YOLO (pjreddie.com, 2018)

YOLO memiliki arsitektur yang sederhana yaitu jaringan saraf tiruan. Jaringan saraf ini hanya menggunakan jenis lapisan standar, konvolusi dengan kernel  $3 \times 3$  dan *max-pooling* dengan  $2 \times 2$  kernel. Lapisan konvolusional terakhir memiliki  $1 \times 1$  kernel digunakan untuk mengecilkan data ke bentuk  $13 \times 13 \times 125$ .  $13 \times 13$  ini seharusnya terlihat familiar karena merupakan ukuran grid yang dibagi menjadi gambar. 125 merupakan Channel untuk setiap grid. 125 ini berisi data untuk kotak pembatas dan prediksi kelas. Kenapa 125? setiap sel grid memprediksi 5 kotak sekeliling dan dijelaskan oleh 25 elemen data. (Aditya, 2018).

Ilustrasi dari lapisan konvolusi pada YOLOv3 dapat dilihat pada **Gambar 2.16**.

Layer	kernel	stride	output shape
Input			(416, 416, 3)
Convolution	3x3	1	(416, 416, 16)
MaxPooling	2x2	2	(208, 208, 16)
Convolution	3x3	1	(208, 208, 32)
MaxPooling	2x2	2	(104, 104, 32)
Convolution	3x3	1	(104, 104, 64)
MaxPooling	2x2	2	(52, 52, 64)
Convolution	3x3	1	(52, 52, 128)
MaxPooling	2x2	2	(26, 26, 128)
Convolution	3x3	1	(26, 26, 256)
MaxPooling	2x2	2	(13, 13, 256)
Convolution	3x3	1	(13, 13, 512)
MaxPooling	2x2	1	(13, 13, 512)
Convolution	3x3	1	(13, 13, 1024)
Convolution	3x3	1	(13, 13, 1024)
Convolution	1x1	1	(13, 13, 125)

**Gambar 2.16.** Ilustrasi lapisan konvolusi YOLOv3 (Aditya, 2018)

## 2.7. Mengukur Kinerja Algoritma klasifikasi dengan *Confussion Matrix*

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya confusion matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya.

Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dapat dibagi menjadi 4 (empat) jenis yaitu klasifikasi binary, multi-class, multi-label dan hierarchical. Pada klasifikasi binary, data masukan dikelompokkan ke dalam salah satu dari dua kelas. Jenis klasifikasi ini merupakan bentuk klasifikasi yang paling sederhana dan banyak digunakan. Contoh penggunaannya antara lain dalam sistem yang melakukan deteksi orang atau bukan, sistem deteksi kendaraan atau bukan, dan sistem deteksi pergerakan atau bukan.

Sementara itu, pada bentuk klasifikasi multi-class, data masukan diklasifikasikan menjadi beberapa kelas. Bentuk klasifikasi multi-label pada dasarnya sama dengan multi-class dimana data dikelompokkan menjadi beberapa kelas, namun pada klasifikasi multi-label, data dapat dimasukkan dalam beberapa kelas sekaligus. Bentuk klasifikasi yang terakhir adalah hierarchical. Data masukan dikelompokkan menjadi beberapa kelas,

namun kelas tersebut dapat dikelompokkan kembali menjadi kelas-kelas yang lebih sederhana secara hirarkis.

Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Nilai True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, True Positive (TP) merupakan data positif yang terdeteksi benar. False Negative (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif.

Pada jenis klasifikasi binary yang hanya memiliki 2 keluaran kelas, confusion matrix dapat disajikan seperti pada **Tabel 2.1**

<b>Kelas</b>	<b>Terklasifikasi Positif</b>	<b>Terklasifikasi Negatif</b>
Positif	TN (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

**Tabel 2.1.** Tabel *Confusion Matrix*

Berdasarkan nilai True Negative (TN), False Positive (FP), False Negative (FN), dan True Positive (TP) dapat diperoleh nilai *accuracy*, *presisi*, *recall* dan *F1 score*. Accuracy tepat digunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data *False Negatif* dan *False Positif*

yang sangat mendekati (*symmetric*). Namun jika jumlahnya tidak mendekati, maka sebaiknya menggunakan F1 Score sebagai acuan.

- **Accuracy**, yaitu menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan Persamaan 2.1.

$$Accuracy (A) = \left( \frac{TP+TN}{TP+TN+FP+FN} \right) \times 100\% \quad (2.1)$$

- **Precision (P)**, yaitu menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Nilai presisi dapat diperoleh dengan Persamaan 2.2.

$$Presisi (P) = \left( \frac{TP}{TP+FP} \right) \times 100\% \quad (2.2)$$

- **Recall (R)**, atau sensitivity, yaitu menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai recall diperoleh dengan Persamaan 2.3

$$Recall (R) = \left( \frac{TP}{TP+FN} \right) \times 100\% \quad (2.3)$$

- **F1 Score (F1)**, yaitu menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan. Nilai F1 Score diperoleh dari Persamaan 2.4.

$$F1\ Score (F1) = 2 \left( \frac{Precision*Recall}{Precision+Recall} \right) \times 100\% \quad (2.4)$$

dimana :

- TP adalah True Positive, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
- TN adalah True Negative, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
- FN adalah False Negative, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
- FP adalah False Positive, yaitu jumlah data positif namun terklasifikasi salah oleh sistem