

**DETEKSI DAN PERHITUNGAN JUMLAH LARVA KEPITING  
RAJUNGAN DENGAN METODE *OBJECT DETECTION***



**TUGAS AKHIR**

*Disusun dalam rangka memenuhi salah satu persyaratan*

*Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika*

*Fakultas Teknik Universitas Hasanuddin*

*Makassar*

**Disusun Oleh:**

**MUH. ARIEF WICAKSONO**

**D421 15 302**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2019**



## LEMBAR PENGESAHAN

### “DETEKSI DAN PERHITUNGAN JUMLAH LARVA KEPITING RAJUNGAN DENGAN METODE *OBJECT DETECTION*”

Disusun Oleh:

**MUH. ARIEF WICAKSONO**

**D421 15 302**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 30 Desember 2019.  
Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik (S.T)  
pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

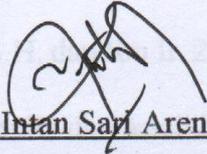
Gowa, 30 Desember 2019

Disetujui Oleh:

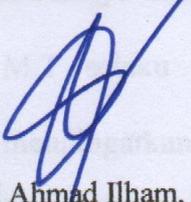
Pembimbing I,

Pembimbing II,

  
Dr. Eng. Muhammad Niswar, ST., M.IT.  
NIP. 19730922 199903 1 001

  
Dr. Eng. Intan Sari Areni, ST., M.T.  
NIP. 19750203 200012 2 002

Diterima dan disahkan oleh:  
Ketua Departemen Teknik Informatika

  
Dr. Amil Ahmad Ilham, S.T., M.IT  
NIP. 19731010 199802 1 001



## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul “DETEKSI DAN PERHITUNGAN JUMLAH LARVA KEPITING RAJUNGAN DENGAN METODE *OBJECT DETECTION*” 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, Bapak Andy Riswanto, S.H. dan Ibu Ir. Zaenab Muslimin, M.T. yang selalu menjadi motivasi penulis dari kecil hingga berada pada tahap ini dan menyelesaikan penyusunan tugas akhir ini;
2. Bapak Dr.Eng. Muhammad Niswar,ST., M.IT selaku pembimbing I yang senantiasa memberikan saran-saran serta bantuan selama proses pengambilan data hingga selesainya sistem ini dibuat, dan Ibu Dr.Eng Intan Sari Areni, ST., M.T selaku pembimbing II yang selalu menyediakan waktu untuk mengingatkan penulis untuk selalu semangat menyelesaikan penulisan penyusunan tugas akhir ini;
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. Bapak Ir. Christoforus Yohannes, M.T., Ibu Dr. Ir. Ingrid Nurtanio, M.T., Bapak Dr. Ir. Zahir Zainuddin, M.Sc., Bapak Dr. Indrabayu, S.T., M.T., M.Bus.Sys., Bapak Dr. Adnan, 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;
5. Bapak Ady Wahyudi Paundu, S.T., M.T., selaku tenaga pengajar Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis dalam kelancaran *Research Internship* ke Jepang;
6. Kak Wahyudi, Kak Novi, dan Kak Furqan selaku mahasiswa S2 yang senantiasa membantu penulis dalam pengambilan data serta memberikan arahan kepada penulis dalam penyelesaian sistem pada tugas akhir ini;
7. Saudara Ryan Rafli, Fuad Khairi Hamid, Sabtian Juliana, Said Syamil Amas, Fadel Rezky Ramadhan, Muh. Zulfachril, Saudari Laura Natalia, Umniyah Nur Aprilyah, Charina, Reka Regina, Jusmiati, Khusnul Khatima selaku sahabat-sahabat SAPULIDI yang senantiasa memberikan semangat serta memotivasi penulis untuk selalu ceria hingga menyelesaikan penyusunan tugas akhir ini;



8. Teman-teman IoT Army yang senantiasa membantu penulis dalam masa perkuliahan;
9. Teman-teman lab UBICON yang senantiasa menemani dan menghibur selama masa penyusunan tugas akhir ini;
10. Saudara/saudari HYPERV15OR atas dukungan dan semangat yang diberikan kepada penulis;
11. Segenap Staff dan Dosen Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis;
12. Putri Dwiyanti Mardan atas dukungannya selama ini;
13. Orang-orang berpengaruh lainnya yang tanpa sadar telah memotivasi dan membantu penulis.

Wassalam

Makassar, Desember 2019

Penulis



## ABSTRAK

Kepiting Rajungan merupakan salah satu jenis kepiting yang banyak terdapat di perairan payau Indonesia dan menjadi salah satu komoditas andalan Indonesia. Namun rajungan yang ada di alam dari waktu ke waktu terus mengalami penurunan, karena selama ini perdagangan rajungan dilakukan dengan mengambil stok langsung dari alam dan bukan berasal dari hasil budidaya. Maka dari itu Balai Perikanan Budidaya Air Payau (BPBAP) Takalar berupaya melakukan pembenihan rajungan secara massal untuk kepentingan pengembangan usaha budidaya, namun BPBAP Takalar menemukan beberapa tantangan, salah satunya terkait perhitungan *survival rate* dari larva rajungan yang diperkirakan sekitar 50%, nilai *survival rate* ini dihitung hanya dengan menggunakan perkiraan dan metode *sampling*. *Survival rate* yang dimaksudkan disini ialah jumlah larva rajungan dari fase *zoea* yang selanjutnya menjadi *megalopa*. Maka dari itu diperlukan adanya suatu sistem yang dapat mendeteksi larva rajungan serta menghitung jumlahnya, agar *survival rate* dari larva rajungan dapat dihitung secara jelas. Dengan menggunakan salah satu metode pada *machine learning* yaitu *deep learning*, maka sistem yang dapat mendeteksi dan menghitung larva kepiting rajungan dapat dibuat. Algoritma yang digunakan ialah YOLO (*You Only Look Once*) versi ke 3, jumlah data yang digunakan sebanyak 375 gambar dan dibagi menjadi 300 data *training* 75 data *testing* . Akurasi yang didapatkan dengan nilai *max batches* 2000 ialah sebesar 86.16%.

**Kata kunci** : larva, kepiting rajungan, *object detection*, *deep learning*, *convolutional neural network*, YOLOV3, *counting*.



## DAFTAR ISI

LEMBAR PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
KATA PENGANTAR .....	i
ABSTRAK .....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	3
1.6. Metode Penulisan .....	4
1.6.1. Metode Pengambilan Data .....	4
1.6.2. Studi Literatur .....	4
1.6.3. Diskusi dan Konsultasi.....	4
1.7. Sistematika Penulisan.....	5
BAB II.....	7
2.1. Kepiting Rajungan.....	7
2.2. Pengolahan Citra .....	8
2.3. <i>Machine Learning</i> .....	10
2.3.1. <i>Training, Development, Testing Set</i> .....	11
2.3.2. <i>Supervised Learning</i> .....	13
2.3.3. <i>Regresi</i> .....	17
2.3.4. <i>Convolutional Neural Network</i> .....	17
2.3.5. <i>Pooling</i> .....	18



2.5.	Algoritma YOLO V3 ( <i>You Only Look Once</i> ).....	20
2.6.	Penelitian terkait.....	23
BAB III.....		27
3.1.	Tahapan Penelitian .....	27
3.2.	Waktu dan Lokasi Penelitian.....	29
3.3.	Instrumen Penelitian.....	29
3.4.	Teknik Pengambilan Data .....	29
3.5.	Perancangan Sistem.....	30
3.5.1.	<i>Labelling Image</i> .....	31
3.5.2.	<i>Preprocess</i> .....	34
3.5.3.	<i>Feature Extraction</i> .....	34
3.5.4.	<i>Classification</i> .....	37
3.5.5.	<i>Visualization</i> .....	38
3.5.6.	<i>Counting</i> .....	39
3.6.	Analisis kinerja sistem.....	39
BAB IV .....		41
4.1.	Hasil Penelitian.....	41
4.2.	Pembahasan .....	46
4.2.1.	<i>Batch</i> .....	46
4.2.2.	<i>Max Batches</i> .....	47
4.2.3.	<i>Subdivision</i> .....	47
4.2.4.	<i>Width, Height dan Channel</i> .....	48
4.2.5.	<i>Learning Rate, Steps, dan Scale</i> .....	48
4.2.6.	<i>Data Augmentation (angle, saturation, exposure, hue)</i> .....	48
4.2.7.	Hasil pengujian sistem .....	49
4.2.8.	Hasil pengujian sistem pada objek yang berbeda .....	49
4.2.9.	Hasil pengujian nilai <i>confidence</i> .....	49
BAB V .....		51
	Kesimpulan.....	51
	Saran.....	51



DAFTAR PUSTAKA .....	53
LAMPIRAN .....	54
Lampiran 1. <i>Source code</i> training YOLO V3 dengan framework darknet .....	54
Lampiran 2. <i>Source code</i> deteksi dan visualisasi larva dengan YOLO V3.....	68
Lampiran 3. <i>Source code counting</i> larva dengan YOLO V3 .....	74



## DAFTAR GAMBAR

<b>Gambar 2.1.</b> Kepiting Rajungan (Dictio.id, 2018) .....	7
<b>Gambar 2.2.</b> Citra Lena dan Citra Kapal (Rinaldi, 2004).....	10
<b>Gambar 2.3.</b> <i>Supervised Learning</i> (Jan Wira, 2019) .....	13
<b>Gambar 2.4.</b> <i>Supervised Learning – mathematical explanation</i> (Jan Wira, 2019) .....	14
<b>Gambar 2.5.</b> <i>Supervised Learning – mathematical explanation 2</i> (Jan Wira, 2019) .....	14
<b>Gambar 2.6.</b> <i>Supervised Learning Framework</i> (Jan Wira, 2019) .....	15
<b>Gambar 2.7.</b> Ilustrasi <i>binary classification</i> (Jan Wira, 2019).....	16
<b>Gambar 2.8.</b> Ilustrasi <i>multi-label</i> dan <i>multi-class classification</i> (Jan Wira, 2019) .....	16
<b>Gambar 2.9.</b> <i>Sliding Window</i> (Jan Wira, 2019).....	18
<b>Gambar 2.10.</b> Contoh <i>pooling</i> (Jan Wira, 2019) .....	19
<b>Gambar 2.11.</b> <i>Convolution</i> dan <i>pooling</i> (Jan Wira, 2019) .....	19
<b>Gambar 2.12.</b> <i>Convolutional Neural Network</i> (Jan Wira, 2019).....	20
<b>Gambar 2.13.</b> Perbandingan kinerja deteksi pada <i>COCO Dataset</i> (pjreddie.com, 2018) .....	21
<b>Gambar 2.14.</b> Ilustrasi proses deteksi YOLO (pjreddie.com, 2018).....	22
<b>Gambar 2.15.</b> Ilustrasi lapisan konvolusi YOLO V3 (Aditya, 2018).....	23
<b>Gambar 3.1</b> Diagram Tahapan Penelitian.....	27

**3.2** *Diagram* perancangan sistem deteksi dan perhitungan larva

.....31



<b>Gambar 3.3</b> Contoh gambar masukan .....	32
<b>Gambar 3.4.</b> <i>Interface</i> dari program <i>YOLO Mark</i> .....	33
<b>Gambar 3.5.</b> Contoh file anotasi dari program <i>YOLO Mark</i> .....	33
<b>Gambar 3.6</b> Ilustrasi proses <i>resize</i> gambar .....	34
<b>Gambar 3.7</b> Perkalian matriks gambar dan matriks filter.....	35
<b>Gambar 3.8.</b> <i>Output Matrix</i> .....	36
<b>Gambar 3.9</b> Pergeseran <i>filter</i> dengan <i>stride</i> =1 .....	37
<b>Gambar 3.10.</b> <i>Fully Connected Layer</i> .....	38
<b>Gambar 4.1</b> Gambar larva untuk data training .....	41
<b>Gambar 4.2</b> Gambar larva yang diujicoba.....	45
<b>Gambar 4.3</b> Hasil pengujian nilai <i>confidence</i> .....	46



## DAFTAR TABEL

<b>Tabel 2.1.</b> Penelitian terkait.....	26
<b>Tabel 3.1</b> <i>Confussion Matrix</i> .....	40
<b>Tabel 4.1.</b> Evaluasi nilai mAP sistem menggunakan <i>confussion matrix</i> .....	42
<b>Tabel 4.2.</b> Hasil pengujian sistem.....	43
<b>Tabel 4.3.</b> Hasil pengujian sistem pada objek yang berbeda.....	44
<b>Tabel 4.4.</b> Hasil pengujian nilai <i>confidence</i> pada <i>bounding box</i> .....	46



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Rajungan (*Portunus pelagicus*) merupakan salah satu sumberdaya perikanan Indonesia yang mempunyai potensi besar untuk menjadi komoditas ekspor unggulan non migas. Permintaan rajungan terus mengalami peningkatan setiap tahunnya (Katisya dkk, 2017). Berdasarkan data dari Direktorat Jenderal Penguatan Daya Saing Produk Kelautan dan Perikanan terkait Kinerja Ekspor Produk Perikanan Indonesia Tahun 2018, pada periode Januari – September 2018, ekspor produk rajungan mencapai USD 370.14 juta atau sebesar 10.50% dari total nilai ekspor perikanan Indonesia, sedangkan dari sisi volume ekspor rajungan mencapai 21.57 ribu ton atau setara dengan 2.69% dari total volume ekspor perikanan Indonesia. Pasar utama produk Rajungan Indonesia adalah USA, Jepang, China, Malaysia dan Singapura. Berdasarkan data BPS, pasar USA menyerap 49.44% produk rajungan Indonesia dan menyumbang devisa sebesar USD 280.82 juta. Sebagian besar kebutuhan ekspor rajungan diperoleh dari hasil tangkapan di alam.

Namun tidak dapat dipungkiri bahwa populasi rajungan yang ada di alam dari waktu ke waktu terus mengalami penurunan sehingga upaya untuk menjaga populasi rajungan di alam adalah dengan melakukan kegiatan budidaya. Kegiatan pembenihan rajungan saat ini masih mengalami kendala karena hingga saat ini keberhasilan pembenihan rajungan menunjukkan tingkat *survival rate* (SR) dari

ngan masih rendah yaitu 25–30%. Saat ini teknologi pembenihan rajungan ngkan di 2 (dua) balai budidaya yaitu BPBAP (Balai Perikanan Budidaya



Air Payau) Takalar Jepara dan BPBAP Takalar yang telah berhasil melakukan pembenihan rajungan secara massal, dan saat ini telah berhasil dikembangkan di masyarakat Kabupaten Demak, Jepara, Tarakan, Balikpapan, Belitung, Pangkalan Susu, dan Bangka.

Perhitungan *survival rate* dari larva rajungan dilakukan dengan menggunakan rumus dari Effendi (1979) yaitu Jumlah larva pada akhir pembesaran / Jumlah larva pada awal pembesaran 100X. Namun untuk menghitung jumlah larva pada awal dan akhir pembesaran, pembudidaya hanya menggunakan perhitungan manual menggunakan metode sampling serta melakukan perkiraan untuk menentukan jumlah dari larva rajungan. Hal ini akan membutuhkan waktu yang cukup lama serta ketelitian yang sangat tinggi, akan tetapi hasilnya tidak akurat.

Oleh karena itu pada tugas akhir ini dibangun sebuah sistem yang dapat mendeteksi dan menghitung larva kepiting rajungan dengan metode *Object Detection*. Untuk melakukan perhitungan, larva di kumpulkan dalam wadah 1 liter dan diambil gambarnya oleh kamera, hal ini di lakukan berulang kali hingga terkumpul 200 gambar dimana ada 200 liter air dalam kolam pembesaran. Sistem akan mendeteksi dan menghitung jumlah larva pada tiap gambar lalu menjumlahkan keseluruhan larva kepiting.

## 1.2. Rumusan Masalah

Rumusan masalah yang akan diuraikan dalam penelitian ini antara lain :

1. Bagaimana cara mendeteksi dan menghitung larva rajungan dengan kamera atas air menggunakan metode *object detection* ?



2. Bagaimana kinerja sistem deteksi dan perhitungan jumlah larva rajungan dalam suatu kolam pemeliharaan?

### 1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Untuk membuat sistem yang dapat digunakan mendeteksi dan menghitung larva rajungan menggunakan metode *object detection*
2. Untuk mengetahui kinerja sistem dalam mendeteksi dan menghitung larva rajungan dalam suatu kolam

### 1.4. Manfaat Penelitian

Dengan dilakukannya penelitian ini, diharapkan manfaat yang didapatkan antara lain :

1. Bagi masyarakat yang dalam hal ini peternak atau pembudidaya kepiting rajungan, penelitian ini dapat digunakan sebagai sistem yang dapat digunakan untuk menghitung jumlah rajungan dengan efisien.
2. Bagi peneliti, penelitian ini dapat digunakan untuk menambah pengetahuan khususnya dibidang kecerdasan buatan untuk mendeteksi objek terkhusus larva rajungan.
3. Bagi instansi pendidikan, penelitian ini dapat digunakan sebagai acuan dan referensi ilmiah untuk melakukan penelitian-penelitian lanjutan.

### 1.5. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

Objek penelitian berupa larva rajungan di BPBAP Takalar.



2. Perhitungan jumlah larva kepiting rajungan dengan mendeteksi jumlah larva rajungan pada sebuah wadah untuk mewakili satu kolam.
3. Objek yang akan dideteksi berupa gambar.
4. Pengambilan data dilakukan pada siang hari dengan kondisi objek dapat terlihat dengan jelas.

## **1.6. Metode Penulisan**

Terdapat beberapa metode penulisan yang digunakan pada penelitian ini yaitu :

### **1.6.1. Metode Pengambilan Data**

Metode pengambilan data dilakukan dengan cara mengambil gambar objek secara langsung dengan mendatangi Balai Perikanan Budidaya Air Payau Takalar. Pengambilan gambar dilakukan dengan mengambil gambar larva rajungan pada sebuah wadah menggunakan kamera.

### **1.6.2. Studi Literatur**

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

### **1.6.3. Diskusi dan Konsultasi**

Diskusi dilakukan dengan melakukan diskusi bersama mahasiswa S2 yang sedang melakukan penelitian di tempat yang sama serta konsultasi secara

langsung kepada dosen pembimbing serta pihak-pihak professional lainnya berhubungan dengan penelitian ini.



## **1.7. Sistematika Penulisan**

Laporan penelitian ini dibagi menjadi lima bab yang tersusun secara sistematis sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang diangkatnya judul penelitian “Deteksi dan Perhitungan Jumlah Larva Kepiting Rajungan dengan Metode Object Detection”, disertai dengan rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metode penulisan, dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas tentang teori-teori umum yang berkaitan dengan konsep dasar metode deteksi objek berbasis sistem jaringan saraf dan perhitungan jumlah objek yang terdeteksi.

### **BAB III METODOLOGI PENELITIAN**

Bab ini memberikan gambaran mengenai perancangan sistem deteksi dan perhitungan jumlah larva kepiting rajungan dengan metode object detection beserta konsep perancangannya.

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

Bab ini berisi tentang hasil deteksi dan hitung jumlah larva kepiting rajungan serta pembahasan yang dilengkapi dengan tabel hasil implementasi dan ujicoba sistem.



## **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. Kepiting Rajungan

Rajungan adalah nama sekelompok kepiting dari beberapa marga anggota suku *Portunidae*. Jenis-jenis kepiting ini dapat berenang dan sepenuhnya hidup di laut. Rajungan sangat bergantung akan air. Dia tidak akan bisa tahan hidup tanpa air laut, dalam jangka waktu tertentu. Karena ketergantungannya itulah, si rajungan juga dijuluki sebagai si swimmer crab. Walaupun sih, sebenarnya dia lebih banyak ditemukan di berjalan di dasar laut, dibandingkan melayang di dalam air.

Secara umum morfologi kepiting rajungan berbeda dengan kepiting bakau (*Scylla serrata*), di mana si rajungan (*Portunus pelagicus*) memiliki bentuk tubuh yang lebih ramping dengan capit yang lebih panjang dan memiliki berbagai warna yang menarik pada karapasnya. Crajungan dapat dpilihat pada **Gambar 2.1**.



**Gambar 2.1.** Kepiting Rajungan (Dictio.id, 2018)



gan melihat warna dari karapas dan jumlah duri pada karapasnya, maka sudah dapat dibedakan dengan kepiting bakau. Rajungan memiliki karapas

berbentuk bulat pipih, sebelah kiri-kanan mata terdapat duri sembilan buah, di mana duri yang terakhir berukuran lebih panjang.

Di beberapa spot diving di nusantara, terutama di daerah secret bay, Gilimanuk, Bali, ternyata rajungan dianggap hama. Ini karena rajungan memangsa kuda laut dan frog fish yang menjadi obyek wisata selam di secret bay ini. Secara rutin, orang-orang adat di sekitar secret bay, yang juga mengelola spot diving secret bay, mencari dan menangkapnya. Selain untuk mengurangi populasinya yang memang sangat banyak, sehingga membahayakan populasi hewan unik lainnya, rajungan dicari juga untuk dimakan.

Saat ini, rajungan sudah mulai dikalengkan dan diekspor ke luar negeri. Karena hasilnya cukup menggiurkan, rajungan terus dicari dan ditangkap habis-habisan. Padahal keberadaan rajungan cukup penting untuk keberlangsungan ekosistem di laut. Dan karena eksploitasi itu pula, pemerintah mulai membudidayakan si rajungan, agar penangkapan di alam semakin berkurang.

## 2.2. Pengolahan Citra

Data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio (bunyi, suara, musik), dan video. Keempat macam data atau informasi ini sering disebut multimedia. Era teknologi informasi saat ini tidak dapat dipisahkan dari multimedia. Situs website di Internet dibuat semenarik mungkin dengan menyertakan visualisasi berupa gambar atau video yang dapat diputar.

Beberapa waktu lalu istilah SMS (*Short Message Service*) begitu populer bagi

telepon genggam (handphone atau HP). Tetapi, saat ini orang tidak hanya mengirim pesan dalam bentuk teks, tetapi juga dapat mengirim pesan berupa



gambar maupun video, yang dikenal dengan layanan MMS (*Multimedia Message Service*). (Rinaldi, 2004)

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). **Gambar 2.2** adalah citra seorang gadis model yang bernama Lena, dan gambar di sebelah kanannya adalah citra kapal di sebuah pelabuhan. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat :

1. optik berupa foto,
2. analog berupa sinyal video seperti gambar pada monitor televisi,
3. digital yang dapat langsung disimpan pada suatu pita magnetik.

Citra yang dimaksudkan adalah “citra diam” (*still images*). Citra diam adalah citra tunggal yang tidak bergerak. **Gambar 2.2** adalah contoh sebuah citra diam.





(a) Lena



(b) Kapal

**Gambar 2.2.** Citra (a) Lena dan Citra (b.) Kapal (Rinaldi, 2004)

Citra bergerak (*moving images*) adalah rangkaian citra diam yang ditampilkan secara beruntun (sekuensial) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri atas ratusan sampai ribuan *frame*.

### 2.3. *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,

misalnya, pada *machine learning*, inferensi yang dimaksud lebih ke ranah variabel. misalnya, apakah penjualan akan meningkat apabila biaya ditingkatkan. Bila kamu ingat dengan mata pelajaran matematika SMA



(logika preposisi), kamu sadar bahwa membuat sistem cerdas menggunakan representasi pengetahuan simbolik itu susah. Representasi pengetahuan secara tradisional dianggap relatif kurang *scalable*, khususnya apabila bekerja dengan data yang besar. Sementara itu, *machine learning* berada pada daerah representasi data/ilmu/pengetahuan dalam bentuk matematis karena keilmuan *machine learning* diturunkan dari matematika dan statistika.

### 2.3.1. *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).

*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.

Satu sampel pada himpunan data disebut sebagai *data point* atau *instance* yang merepresentasikan suatu kejadian statistik (*event*). Perlu diingat, *training*, *development*, dan *testing data* diambil (*sampled*) dari distribusi yang sama dan memiliki karakteristik yang sama (*independently and identically distributed*). Distribusi pada masing-masing dataset ini juga sebaiknya seimbang (*balanced*) dan memuat seluruh kasus. Misal, sebuah dataset *binary classification* sebaiknya memuat 50% kasus positif dan 50% kasus negatif.

Pada umumnya, rasio pembagian *dataset* adalah (80% : 10% : 10%) atau (90% : 5% : 5%) (*training* : *development* : *testing*). *Development set* pada umumnya bisa tidak digunakan apabila *dataset* berukuran kecil (hanya dibagi menjadi *training* dan *testing set* saja). Dalam kasus ini, pembagian dataset menjadi *training* dan *testing set* pada umumnya memiliki rasio (90% : 10%), (80% : 20%), (70% : 30%), atau (50% : 50%). Pada kasus ini, kinerja saat *training* diuji menggunakan *training set* (dikenal sebagai *closed testing*).

Saat tidak menggunakan *development set* (hanya ada *training* dan *testing set*), opsi lain untuk mengevaluasi model dengan metode *K cross-validation*. Artinya, *training dataset* dibagi menjadi *K* bagian. *K* bagian digunakan untuk *training*, kemudian menguji kinerja model saat latihan (*validation*)

gunakan satu bagian. Hal ini diulangi sebanyak *K* kali dimana sebuah *n* data digunakan sebagai *testing set* sebanyak sekali (bergilir).



### 2.3.2. Supervised Learning

*supervised learning* adalah pembelajaran terarah/terawasi. Artinya, pada pembelajaran ini, ada guru yang mengajar (mengarahkan) dan siswa yang diajar. Manusia berperan sebagai guru, kemudian mesin berperan sebagai siswa. Perhatikan **Gambar 2.3** sebagai ilustrasi, seorang guru menuliskan angka di papan “8, 6, 2” sebagai contoh untuk siswanya, kemudian gurunya memberikan cara mem- baca yang benar untuk masing-masing angka. Contoh angka melambangkan *input*, kemudian cara membaca melambangkan *desired output*. Pasangan *input desired output* ini disebut sebagai *instance* (untuk kasus *supervised learning* ). Pembelajaran metode ini disebut *supervised* karena ada yang memberikan contoh jawaban (*desired output* ).

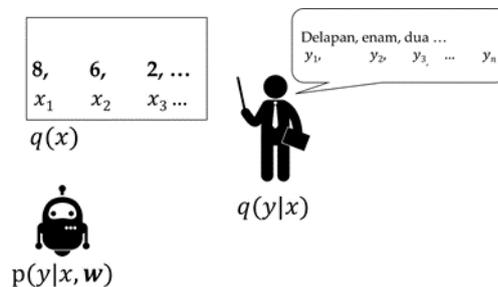


**Gambar 2.3.** *Supervised Learning* (Jan Wira, 2019)

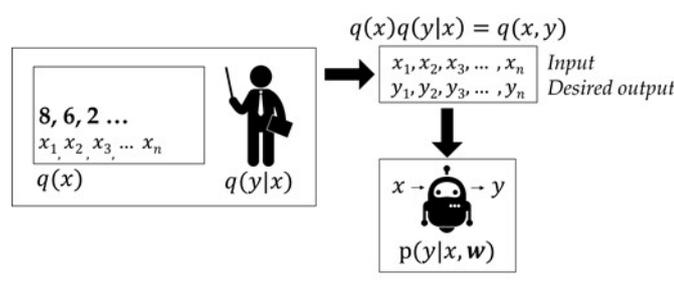
Perhatikan **Gambar 2.4** dan **Gambar 2.5**,  $x$  adalah kejadian (*event random variable*), untuk *event* tertentu dapat dinotasikan sebagai  $\{x_1, x_2, x_3, \dots, x_N\}$ .  $x$  dapat berupa vektor, teks, gambar, dan lain sebagainya (perhatikan konteks pembahasan buku). Seorang guru sudah mempunyai jawaban yang benar untuk masing-masing contoh dengan suatu



fungsi distribusi probabilitas kondisional (*conditional probability density function*)  $q(y | x)$  baca: *function q for y given x*, melambungkan hasil yang benar/diharapkan untuk suatu *event*. Siswa (mesin) mempelajari tiap pasang pasangan *input-desired output* (*training data*) dengan mengoptimalkan *conditional probability density function*  $p(y | x, w)$ , dimana  $y$  adalah target (*output*),  $x$  adalah input dan vektor  $w$  adalah *learning parameters*. Proses belajar ini, yaitu mengoptimalkan  $w$  disebut sebagai *training*. Proses *training* bertujuan untuk mengaproksimasi  $q(y | x)$  melalui  $p(y | x, w)$ .

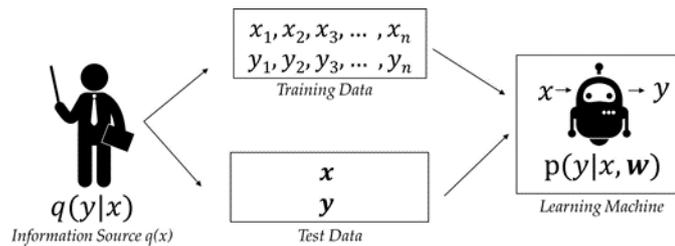


**Gambar 2.4.** *Supervised Learning – mathematical explanation* (Jan Wira, 2019)



**Gambar 2.5.** *Supervised Learning – mathematical explanation 2* (Jan Wira, 2019)





**Gambar 2.6.** *Supervised Learning Framework* (Jan Wira, 2019)

Pada **Gambar 2.6** model memiliki panah ke *training data* dan *test data*, artinya model hasil *training* sangat bergantung pada data dan guru. Model yang dihasilkan *training* (hasil pembelajaran kemampuan siswa) untuk data yang sama bisa berbeda untuk guru yang berbeda.

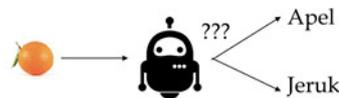
Tujuan *supervised learning*, secara umum untuk melakukan klasifikasi (*classification*). Misalkan mengklasifikasikan gambar buah (apa nama buah pada gambar), diilustrasikan pada **Gambar 2.7**. Apabila hanya ada dua kategori, disebut *binary classification*. Sedangkan bila terdapat lebih dari dua kategori, disebut *multi-class classification*. Contoh *multi-class classification* adalah mengklasifikasikan gambar buah ke dalam himpunan kelas: apel, mangga atau sirsak.

Ada tipe klasifikasi lain disebut *multi-label classification* yaitu ketika ingin mengklasifikasikan suatu instans ke dalam suatu himpunan kelas. Perbedaan *multi-class* dan *multi-label classification* agak *tricky*. Pada *multi-class classification*, suatu instans hanya bisa berkorespondensi

dengan satu kelas. Sedangkan pada *multi-label classification*, satu instans dapat berkorespondensi dengan lebih dari satu kelas. Misalnya, suatu berita



dapat masuk ke kategori agama dan politik pada waktu bersamaan. Artinya, label pada *multi-class classification* bersifat *mutually exclusive*, sedangkan label tidak bersifat *mutually exclusive* pada *multi-label classification*. Perhatikan **Gambar 2.8** sebagai ilustrasi, dimana setiap baris merepresentasikan kelas yang berkorespondensi dengan setiap instans, nilai “1” melambangkan *TRUE* dan nilai “0” melambangkan *FALSE*. *Multi-label classification* dapat didekomposisi menjadi beberapa *Binary classification*, yaitu mengklasifikasikan apakah instans dapat di-assign ke suatu kelas atau tidak.



**Gambar 2.7.** Ilustrasi *binary classification* (Jan Wira, 2019)

Instans	Apel	Mangga	Sirsak
Gambar-1	1	0	0
Gambar-2	0	1	0
Gambar-3	0	0	1
...			

Multi-class Classification

Instans	Agama	Politik	Hiburan
Berita-1	1	1	0
Berita-2	0	1	1
Berita-3	1	0	1
...			

Multi-label Classification

**Gambar 2.8.** Ilustrasi *multi-label* dan *multi-class classification* (Jan Wira, 2019)

Pemahaman *supervised learning* adalah mengingat  $p(y | x, w)$

Ada tiga hal penting pada *supervised learning* yaitu *input*, *desired output*,

dan *learning parameters*. Perlu ditekankan *learning parameters* berjumlah

lebih dari satu, dan sering direpresentasikan dengan vektor (*bold*) atau



matriks. Berdasarkan model yang dibuat, dapat dilakukan klasifikasi (misal simbol yang ditulis di papan adalah angka berapa). Secara konseptual, klasifikasi didefinisikan sebagai persamaan 2.1 yaitu memilih label (kelas/kategori  $y$ ) paling optimal dari sekumpulan label  $C$ , diberikan (*given*) suatu instans data tertentu.

$$\hat{y}_i = \arg \max p(y_i | x_i, w)$$

### 2.3.3. Regresi

Pada persoalan regresi, prediksi output berupa bilangan kontinu. Misalnya pada regresi suatu fungsi polinomial, fungsi  $f(x)$  diberikan data  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ . Setelah itu, digunakan fungsi aproksimasi untuk mencari tahu nilai  $y_{N+1}$  dari data baru  $x_{N+1}$ . Perbedaan regresi dan klasifikasi adalah pada tipe output. Untuk regresi, tipe output adalah nilai kontinu; sementara tipe output pada persoalan klasifikasi adalah suatu objek pada himpunan (i.e., memilih opsi pada himpunan jawaban).

## 2.4. Convolutional Neural Network

CNN (*Convolutional Neural Network*) adalah kemampuan mengenali aspek yang *informatif* pada regional tertentu (lokal). Dibanding *mengcopy* mesin pembelajaran beberapa kali untuk mengenali objek pada banyak regional, ide lebih baik adalah untuk menggunakan *sliding window*, setiap operasi pada *window* 1 bertujuan untuk mencari aspek lokal yang paling informatif. Ilustrasi diberikan oleh

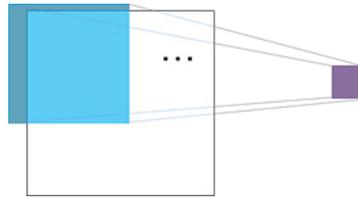
**Gambar 2.9.** Warna biru merepresentasikan satu *window*, kemudian kotak ungu

entasikan aspek lokal paling informatif (disebut *filter*) yang dikenali oleh

Dengan kata lain, suatu *window* mentransformasi menjadi suatu nilai



numerik (filter ). Suatu *window* (regional) dapat ditransformasi menjadi  $d$  nilai numerik ( $d$ -channels, setiap elemen berkorespondensi pada suatu *filter* ). *Window* ini kemudian digeser-geser sebanyak  $T$  kali, sehingga akhirnya didapatkan vektor dengan Panjang  $d * T$  . Keseluruhan operasi ini disebut sebagai *convolution*.

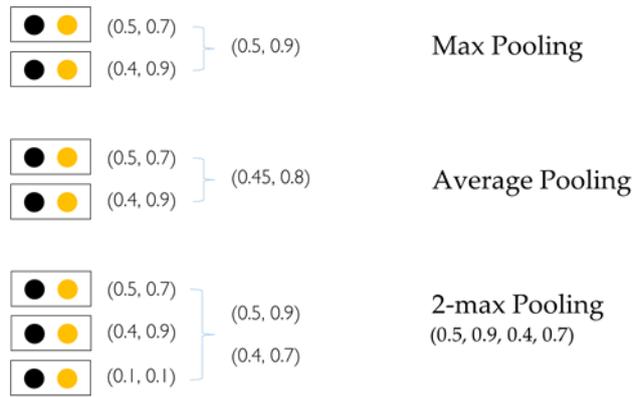


**Gambar 2.9.** *Sliding Window* (Jan Wira, 2019)

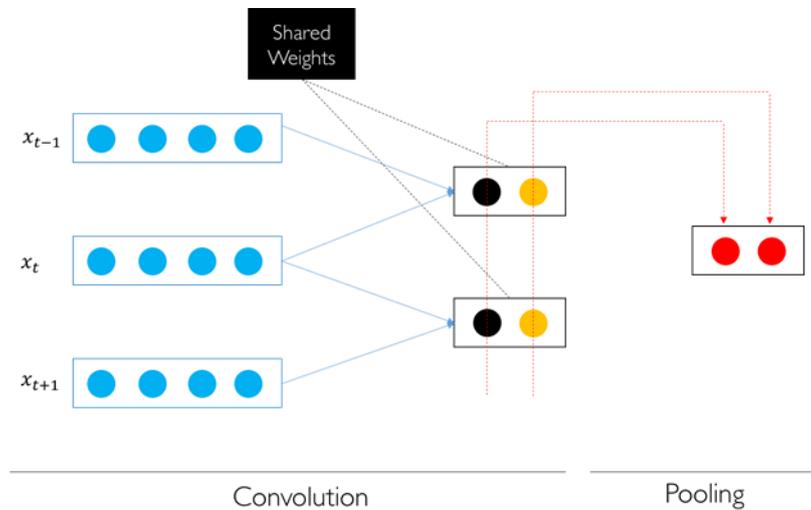
#### 2.4.1. Pooling

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 2.10**. *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 2.11**.





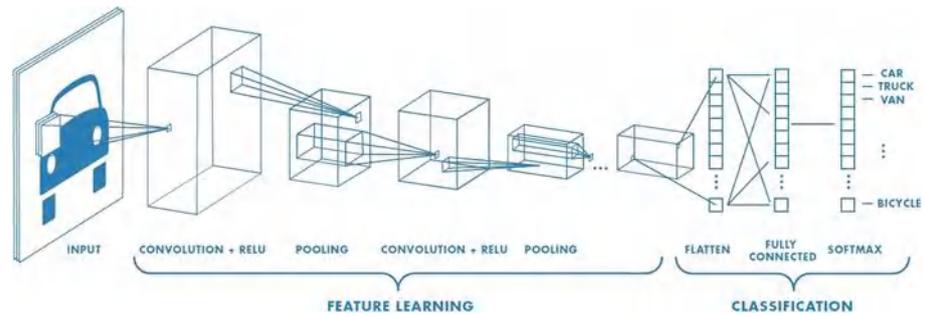
**Gambar 2.10.** Contoh *pooling* (Jan Wira, 2019)



**Gambar 2.11.** *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 2.12**).





**Gambar 2.12.** *Convolutional Neural Network* (Jan Wira, 2019)

### 2.5. Algoritma YOLO V3 (*You Only Look Once*)

YOLO merupakan sebuah algoritma deep learning yang digunakan untuk mendeteksi objek secara real-time dengan cepat. YOLO memiliki 3 versi yaitu YOLO V1 merupakan versi awal YOLO yang sudah jarang digunakan, YOLO V2 yang merupakan versi kedua dari YOLO yang masih banyak digunakan oleh praktisi untuk melakukan pendeteksian secara real-time, dan YOLO V3 yang merupakan versi paling update dan memiliki tingkat akurasi yang tinggi. (joseph, 2018)



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

**Gambar 2.13.** Perbandingan kinerja deteksi pada *COCO Dataset*

(pjreddie.com, 2018)

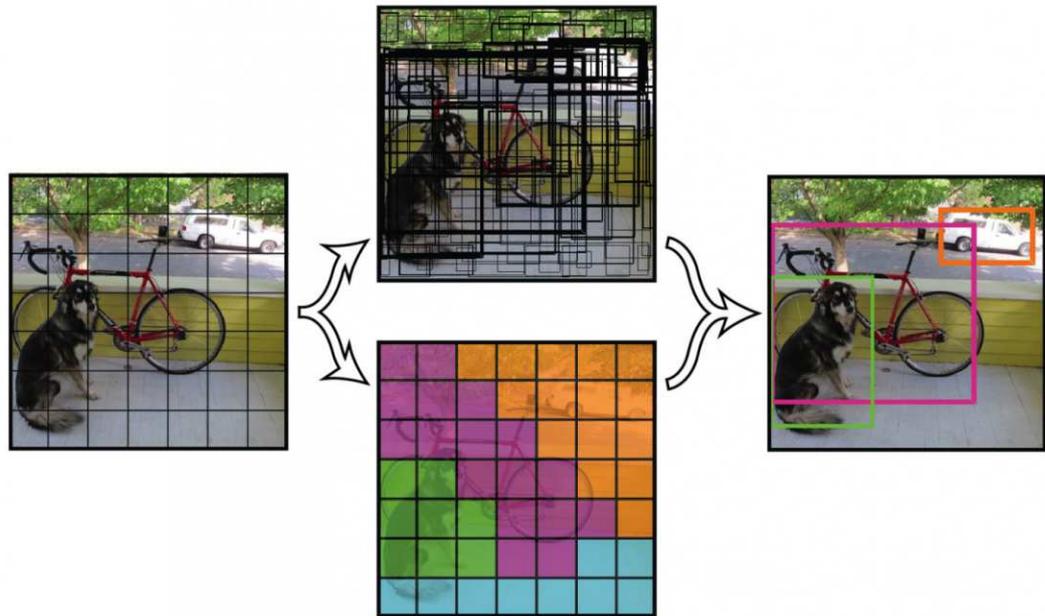
Pada **Gambar 2.13** dapat terlihat bahwa mAp (*mean Average precision*) pada YOLO lebih tinggi dibandingkan algoritma deteksi lain, artinya akurasi yang akan didapat kan jika menggunakan YOLO akan lebih baik jika dibandingkan dengan algoritma lain. (pjredie.com, 2018)

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

Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian liksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah



pembatas ditimbang probabilitasnya untuk mengklasifikasikan sebagai objek atau bukan. Ilustrasi dari proses pendeteksian YOLO dapat dilihat pada **Gambar 2.14**. (Aditya, 2018)



**Gambar 2.14.** 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 pembatas dan prediksi kelas. Kenapa 125? setiap sel grid memprediksi 5 kelas dan dijelaskan oleh 25 elemen data. (Aditya, 2018)



Ilustrasi dari lapisan konvolusi pada YOLO V3 dapat dilihat pada **Gambar**

**2.15.**

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 2.15.** Ilustrasi lapisan konvolusi YOLO V3 (Aditya, 2018)

## 2.6. Penelitian terkait

Beberapa penelitian telah dilakukan dalam membuat sistem yang dapat mendeteksi dan menghitung sebuah objek. Ibrahim Aliyu , dkk telah melakukan penelitian terkait perhitungan dan pengklasifikasian benih ikan di Nigeria. Mereka menggunakan metode *Image processing* dengan melalui beberapa tahapan : *Image*

*g*, *Image segmentation*, *Features extraction*, *Classification*, and *Counting*.



Hasil dari penelitian mereka menghasilkan sebuah sistem yang dapat menghitung dan mengklasifikasikan benih ikan namun akurasinya masih harus ditingkatkan.

Kemudian ada Valliappan Raman,dkk melakukan penelitian tentang peranan Machine Learning untuk mendeteksi dan menghitung larva atau *juvenile* ikan di Malaysia. Valliappan menggunakan 4 tahapan untuk mendeteksi dan menghitung jumlah larva dan *juvenile* ikan, Image Acquisition yang dimana menggunakan kamera untuk mengambil gambar dan mengubahnya menjad grayscale, lalu *Image Enhancement* yang dimana digunakan untuk menambahkan kontras pada gambar sehingga objek yang ingin di deteksi dapat dilihat dengan jelas, lalu selanjutnya Segmentation yang digunakan untuk mendeteksi *region of intereset* dari gambar yang memungkinkan sistem untuk mendeteksi kordinat dari tiap sudut *bounding box* yang berada di sekeliling objek, dan tahapan terakhir yaitu *classification* yang digunakan untuk mengklasifikasikan apakah larva atau *juvenile* saling berdekatan atau tidak sehingga akurasi perhitungannya dapat di tingkatkan. Sistem ini memiliki akurasi sebesar 82% untuk mendeteksi larva ikan dan 87% untuk mendeteksi *juvenile* ikan. Namun sistem ini hanya dapat di terapkan pada gambar yang backgroundnya tidak memiliki *noise*, dan akurasi akan berkurang jikalau pada gambar terdapat banyak *noise*.

Pada tahun 2018, Jirabhorn Kaewchote, Sittichoke Janyong, dan Wasit Limprasert melakukan penelitian terkait perhitungan jumlah anak udang setelah tahapan larva dengan metode *Image Processing*. Mereka melakukan pengujian

akan 2 metode ekstraksi fitur yaitu RGB dan *Local Binary Pattern*. Hasil



dari pengujiannya ialah metode *Local Binary Pattern* lebih unggul dengan akurasi 98.50% sedangkan untuk RGB masih salah untuk menghitung jumlah anak udang.

Geoffrey French, Mark Henry Fisher, Michal Mackiewicz, dan Coby Needle melakukan penelitian terkait penggunaan CNN untuk menghitung jumlah ikan pada jarring nelayan melalui CCTV. Mereka melakukan pemrosesan dengan mengekstrak gambar dari video, lalu melakukan pewarnaan kepada ikan yang terdapat pada gambar lalu setelah itu, sudut-sudut dari ikan yang diwarnai di ekstrak untuk dapat melihat *object* yang terdeteksi secara jelas, dan tahapan akhir dengan menggabungkan pewarnaan dengan sudut yang telah di ekstrak akan menghasilkan prediksi yang cukup tinggi. Akurasi dari sistem ini dihitung dengan melakukan pengujian pada 6 jaring dan menghasilkan akurasi rata-rata sebesar 90.94%. Berikut adalah **Tabel 2.1.** yang berisi penelitian terkait deteksi dan perhitungan objek dengan metode *machine learning*.



**Tabel 2.1.** Penelitian terkait

Peneliti	Judul	Objek	Tahun	Metode	Hasil
Valliappan Raman, Sundresan Perumal, Sujata Navaratnam, Siti Fazilah	<i>Computer Assisted Counter System for Larvae and Juvenile Fish in Malaysian Fishing Hatcheries by Machine Learning Approach</i>	Larva dan Juvenile Ikan	2015	Segmentasi dan Klasifikasi	Akurasi pada Larva 82% dan akurasi pada <i>Juvenile</i> 87%
Geoffrey French, Mark Henry Fisher, Michal Mackiewicz, Coby Needle	<i>Convolutional Neural Networks for Counting Fish in Fisheries Surveillance Video</i>	ikan	2015	<i>Convolutional Neural Network</i>	Rata-rata akurasi sebesar 90.94%
A. Sanchez-Ortiz, Arista Jlife, Atoany Nazareth Fierro, M. Cedillo Hernandez	<i>Mosquito larva classification method based on convolutional neural networks</i>	Telur Nyamuk	2017	<i>Convolutional Neural Network</i>	Akurasi pada sebuah telur nyamuk 96.88%
Jirabhorn Kaewchote, Sittichoke Janyong, Wasit Limprasert	<i>Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp</i>	Larva Udang	2018	<i>Local Binary Pattern dan Random Forest</i>	Akurasi sistem menggunakan <i>Local Binary Pattern</i> adalah sebesar 98.50%
nez, er A dez	<i>Evaluation of mangrove crab classification system</i>	Larva Kepiting Bakau	2019	<i>K-Nearest Neighbour</i>	Rata-rata akurasi sebesar 85%

