**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**

# **BAB I**

**PENDAHULUAN**

## **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 kelulus hidupan/survival rate (SR) dari larva rajungan masih rendah yaitu 25–30%. Saat ini teknologi pembenihan rajungan dikembangkan di 2 (dua) balai budidaya yaitu BBPBAP 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 X 100. 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 jug atidak 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 akan diambil gambarnya oleh kamera, hal ini di lakukan berulang kali hingga terkumpul 200 gambar yang dimana ada 200 liter air dalam kolam pembesaran. Sistem akan mendeteksi dan menghitung jumlah larva pada tiap gambar lalu menjumlahkan keseluruhan larva kepiting.

## **Rumusan Masalah**

Rumusan masalah yang akan diuraikan dalam penelitian ini antara lain :

1. Bagaimana cara mendeteksi dan menghitung larva rajungan dengan kamera diatas air menggunakan metode object detection ?
2. Bagaimana cara mengevaluasi kinerja sistem deteksi dan perhitungan jumlah larva rajungan dalam suatu kolam pemeliharaan?

## **Tujuan Penelitian**

Tujuan dari penelitian ini adalah :

1. Untuk membuat system yang dapat digunakan mendeteksi larva rajungan menggunakan metode object detection
2. Untuk mengetahui hasil dari implementasi sistem deteksi dan perhtungan larva rajungan dalam suatu kolam

## **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 khsusnya 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.

## **Batasan Masalah**

Adapun Batasan masalah dalam penelitian ini adalah :

1. 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. Deteksi object berupa gambar.
4. Pengambilan data dilakukan pada siang hari dengan kondisi cahaya yang cukup.

## **Metode Penulisan**

Terdapat beberapa metode penulisan yang digunakan pada penelitian ini yaitu :

### 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.

### Studi Literatur

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

### Diskusi dan Konsultasi

Diskusi dan konsultasi dilakuakn dengan melakukan diskusi bersama kakak S2 yang sedang melakukan penelitian di tempat yang sama, konsultasi dilakukan dengan mengadakan tanya jawab secara langsung kepada dosen pembimbing serta pihak-pihak professional lainnya yang berhubungan dengan penelitian ini.

## **Sistematika Penulisan**

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

1. **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.

1. **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.

1. **METODOLOGI PENELITIAN**

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

1. **HASIL DAN PEMBAHASAN**

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

1. **PENUTUP**

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

# **BAB III**

**METODOLOGI PENELITIAN**

### **Tahapan Penelitian**

Sistem yang dibuat pada penelitian ini menggunakan algoritma *Deep Learning* YOLO (*You Only Look Once*) versi 3, YOLO menggunakan *Convolutional Neural Network Layer*  untuk melakukan ekstraksi fitur. Untuk menentukan objek larva pada gambar, YOLO melakukan *scoring* sehingga *region* pada gambar yang memiliki *scoring* yang tinggi dapat dipastikan merupakan objek larva pada gambar. Adapun tahapan-tahapan yang dilakukan dapat dilihat pada **Gambar 3.1** .

**Gambar 3.1.** Diagram Tahapan Penelitian

Tahapan penelitian pada **Gambar 3.1.** dijelaskan sebagai berikut.

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahapan ini di lakukan untuk memngumpulkan penelitian-penelitian terkait metode *Deep Learning* untuk mendeteksi sebuah objek. Pada tahapan ini pula dilakukan penarikan kesimpulan dari penelitian-penelitian terkait.
2. Pengambilan Data dilakukan di Balai Perikanan Budidaya Air Payau Takalar. Data diambil menggunakan kamera *smartphone* android dan iOS, gambar juga diambil menggunakan kamera mirrorless untuk menghasilkan gambar dengan resolusi yang baik.
3. *Preprocessing* merupakan tahapan untuk melakukan penyortiran data untuk menyeleksi data yang resolusinya dibawah standar, setelah di sortir, data akan di beri label menggunakan program YOLO\_mark untuk menandai objek yang ingin di deteksi.
4. Metode yang digunakan untuk membuat sistem pada penelitian ini merupakan metode *Object Detection* pada *Deep Learning* dengan algoritma YOLO (*You Only Look Once*) versi ketiga.
5. Processing merupakan tahapan untuk melakukan pelatihan dari data yang telah di beri label, pada tahap ini pelatihan dilakukan menggunakan *Google Colaboratory. Google Colaboratory* merupakan sebuah *service* dari *Google* yang dikhususkan untuk pengembang *project* *Machine Learning*.
6. Implementasi dari Sistem dibuat menggunakan bahasa pemrograman Python dengan *framework* darknet.
7. Uji coba sistem dilakukan dengan melakukan pengujian sistem untuk mendeteksi larva pada gambar-gambar yang diberikan sehingga akurasi dari sistem dapat di lihat, pada tahapan ini juga diukur waktu yang digunakan oleh program untuk memproses 1 gambar hingga 200 gambar.
8. Tahapan akhir dalam penelitian ini adalah dengan melakukan penulisan laporan penelitian dalam bentuk skripsi sebagai bahan publikasi.

### **Waktu dan Lokasi Penelitian**

Penelitian ini telah dilakukan selama kurang lebih 5 bulan dimulai pada bulan Juni 2019 hingga bulan ditulisnya laporan ini yaitu bulan Oktober 2019. Pengambilan data pada penelitian ini dilakukan di Balai Perikanan Budidaya Air Payau Takalar.

### **Instrumen Penelitian**

Instrumen penelitian yang digunakan pada penelitian ini meliputi :

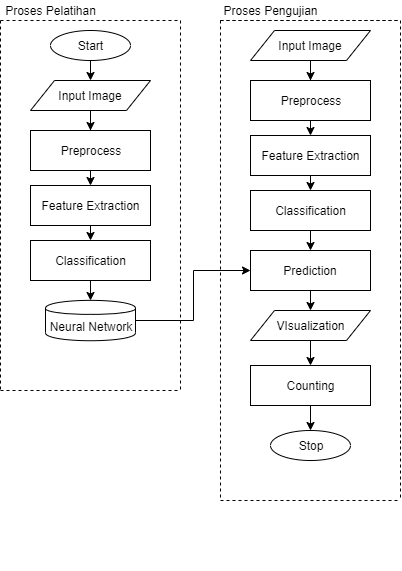
1. Software
2. Windows 10 Home 64-bit
3. Microsoft Visual Studio 2017 Community
4. Jupyter Notebook Environment
5. Google Colaboratory (Intel Xeon [CPU@2.20GHz](mailto:CPU@2.20GHz) NVIDIA Tesla K80, 12 GB GDDR5VRAM)
6. Hardware
7. Laptop Acer Aspire A715-71G, RAM 8GB, Prosesor Intel(R) Core(TM) i5-7300HQ CPU @2.50GHz
8. *Smartphone* Xiaomi Redmi A4
9. *Smartphone* iPhone 5
10. Kamera Sony RX 100

### **Teknik Pengambilan Data**

Data yang digunakan pada penelitian ini menggunakan gambar yang diambil langsung menggunakan kamera *smartphone* serta kamera *digital.* Larva yang akan diambil gambarnya dipindahkan di sebuah wadah dengan air yang cukup dan berisi 5 sampai 10 larva. Larva pada penelitian ini merupakan larva dengan tahapan *Zoea 1* hingga *Zoea 4.* Gambar yang telah dikumpulkan kemudian di saring untuk mendapatkan gambar dengan resolusi terbaik yang akan digunakan untuk sebagi data *training*.

### **Perancangan Sistem**

Secara umum sistem terbagi menjadi dua proses, yaitu proses pelatihan dan proses pengujian seperti yang dapat dilihat pada **Gambar 3.2.** Pada proses pelatihan, fitur yang telah di ekstrak dari data *training* menggunakan layer *Convolutional Neural Network* akan dijadikan input ke dalam *Fully Connected Layer* yang akan menghasilkan model deteksi larva. Model yang dihasilkan oleh proses pelatihan ini akan di gunakan untuk proses pengujian sistem.



**Gambar 3.2** *Flowchart*  perancangan sistem deteksi dan perhitungan larva rajungan

Berdasarkan **Gambar 3.2**, perancangan sistem deteksi dan perhitungan larva rajungan dapat diuraikan sebagai berikut.

* + 1. ***Input Image***

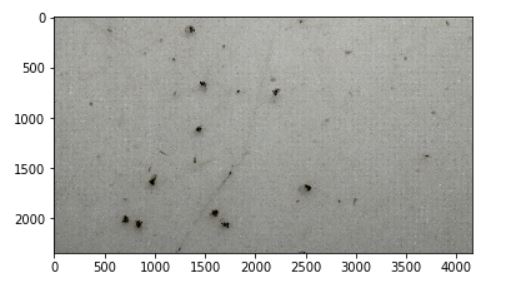
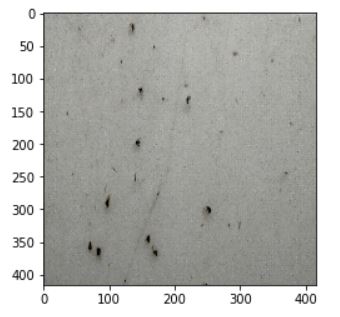
Langkah pertama ialah dengan mempersiapkan gambar larva kepiting rajungan yang akan digunakan untuk pelatihan dan pengujian. Dalam penelitian ini, gambar diambil menggunakan kamera *smartphone* dan kamera *digital,* selain itu data gambar juga di peroleh dari hasil ekstraksi gambar dari video yang diambil menggunakan kamera *smartphone.*Gambar di esktrak setiap 1 detik dari video menggunakan library python yang bernama FFMPEG. Berikut adalah contoh gambar yang diambil untuk dijadikan data latih seperti yang terlihat pada **Gambar 3.3.**



**Gambar 3.3.** Contoh gambar masukan

* + 1. ***Preprocess***

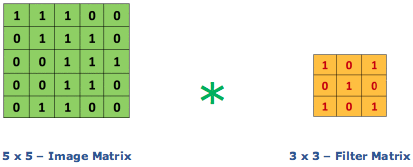
YOLO membutuhkan gambar masukan dengan resolusi 416x416 karena resolusi ini merupakan resolusi optimal untuk dapat mendeteksi objek, bahkan untuk mendeteksi objek berukuran kecil sekalipun. Nilai resolusi ini juga digunakan untuk mengantisipasi eror pada saat melakukan training dengan jumlah data gambar yang sangat banyak dan memiliki resolusi yang bervariasi. Maka dari itu setiap gambar yang dimasukkan ke *neural network* YOLO akan di *resize* resolusinya menjadi 416x416 piksel. Ilustrasi proses *resize* gambar dapat dilihat pada **Gambar 3.4.**

**Gambar 3.4.** Ilustrasi proses *resize* gambar

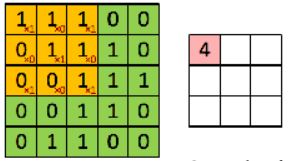
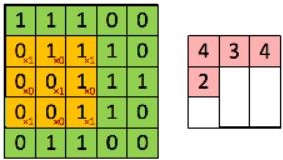
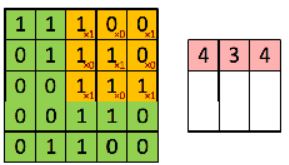
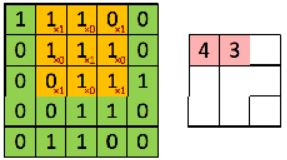
* + 1. ***Feature Extraction***

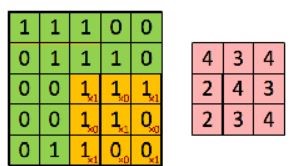
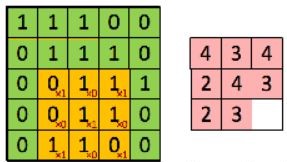
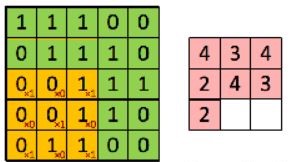
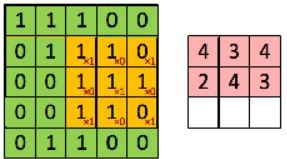
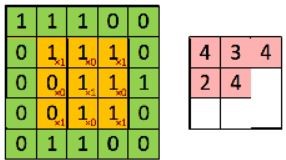
Pada model *Convolutional Neural Network* , setiap gambar masukan akan melalui beberapa tahapan *convolutional layer* dengan *filter(kernels)*, *Pooling layer,* dan *Fully connected layer.* Layer pertama untuk mengekstrak fitur adalah *Convolutional layer* , layer ini menjaga hubungan antar piksel pada gambar dengan menggunakan kotak-kotak kecil sebagai data masukan. Hal ini menggunakan operasi matematika yang dimana mengambil 2 masukan yang dalam hal ini adalah matriks dari gambar dan sebuah *filter* atau *kernel.* Perkalian matriks dapat dilihat pada **Gambar 3.5** yang dimana *image matrix* memiliki jumlah pixel 5x5 dan memiliki nilai pixel 0 dan 1. Kemudian matriks gambardikalikan dengan matriks filter dengan jumlah pixel 3x3.



**Gambar 3.5.** Perkalian matriks gambar dan matriks filter

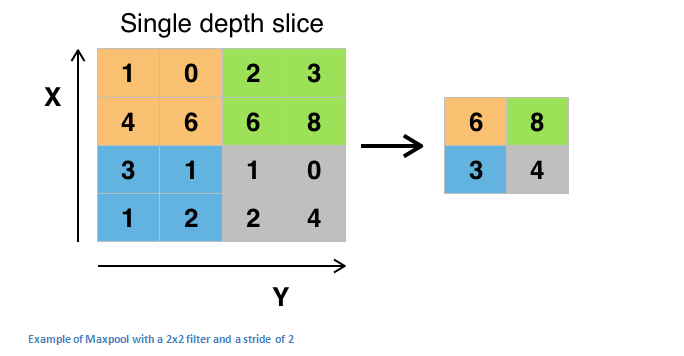
Perkalian dari matriks tersebut menghasilkan *output* berupa sebuah konvolusi yang disebut *Feature Map* yang dapat dilihat pada **Gambar 3.6.**

****



**Gambar 3.6.** *Output Matrix*

Setelah itu *filter* akan bergeser pada gambar sesuai dengan jumlah *stride. Stride* adalah sebuah parameter filter yang ditentukan untuk menentukan jumlah pregeseran piksel pada gambar. Contohnya jika nilai *stride* adalah 1 maka *filter* akan bergeser tiap 1 pixel dalam suatu waktu, ilustrasi dari proses perpindahan dapat *stride* dapat dilihat pada **Gambar 3.7.**

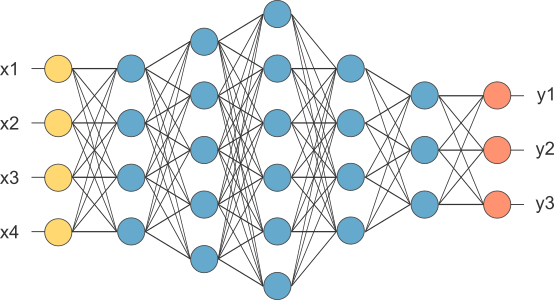


**Gambar 3.7.** pergeseran *filter* dengan *stride*=2

Dalam sistem yang dibuat terdapat *Pooling layer* yang berfungsi untuk mengurangi jumlah parameter yang digunakan saat gambar terlalu besar, ini diperlukan untuk mempercepat proses komputasi model. Dalam sistem yang digunakan ialah tipe pooling berjenis *Max Poolin*g yang berguna untuk mengurangi dimensi dari tiap map tanpa kehilangan informasi-informasi penting.

* + 1. **Classification**

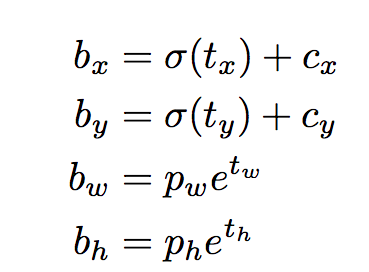
Matriks hasil dari *Max pooling layer* akan di sama ratakan menjadi bentuk vector yang kemudian di masukkan kedalam *Fully Connected Layer.* Dalam tahapan ini model akan di klasifikasikan untuk mengenali apakah objek yang terdapat pada gambar merupakan larva atau bukan, jenis klasifikasi ini merupakan *Binart Classification*. Ilustrasi dari *Fully Connected Layer* ini dapat dilihat pada **Gambar 3.8** yang dimana *feature map* akan di konversi menjadi bentuk vector (x1,x2,x3,…). Dengan *Fully Connected Layer*, Fitur akan di gabungkan untuk membentuk sebuah model. Gabungan dari fitur-fitur disebut akan membentuk sebuah *activation map* yang akan mengklasifikasikan larva atau bukan larva.



**Gambar 3.8.** *Fully Connected Layer*

* + 1. **Visualization**

Untuk menghasilkan gambar yang telah di proses oleh sistem, kita menggunakan OpenCV, yang dimana berfungsi untuk mempercepat proses komputasi data dan untuk menggambar *bounding box*. YOLO memprediksi sebuah object menggunakan *objectness score* pada setiap *bounding box* untuk menentukan apakah *bounding box* akan di gambarkan atau tidak. Secara default nilai threshold untuk *objectness score* adalah 0.5. Jika nilai objectness score pada sebuah bounding box terpenuhi maka sistem akan menggambar bounding box sesuai dengan nilai pada persamaan 3.1 yang dimana bx dan by merupakan kordinat titik tengah x dan y, kemudian bw dan bh merupakan *width* dan *height* dari prediksi. tx,ty,tw,th merupakan keluaran dari proses training. Sedangkan cx dan cy merupakan kordinat ujung kiri dari *grid* pada *bounding box*. pw dan ph adalah dimensi *anchor* untuk *bounding box.*

 (3.1)

* + 1. **Counting**

Proses perhitungan dilakukan dengan mendeteksi jumlah *bounding box* telah digambarkan*. Bounding box* yang telah di gambarkan akan di cocokkan *id* nya dengan *id* *class* yang ingin di hitung jumlahnya, jika *id* dari *bounding box* sama dengan *id class* maka jumlah larva akan bertambah 1. Total keseluruhan dari gambar larva yang di proses oleh sistem akan di jumlahkan dan ditampilkan di akhir pemrosesan sistem yang juga disimpan pada sebuah file txt.

* 1. **Analisis kinerja sistem**

Analisis kinerja sistem deteksi dan perhitungan larva kepiting rajungan dapat dilakukan dengan menghitung nilai akurasi menggunakan *confussion matrix.* Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. *Confussion Matrix* yang digunakan dapat dilihat pada **Tabel 3.1.**

**Tabel 3.1** *Confussion Matrix*

|  |  |  |
| --- | --- | --- |
| **Kelas** | **Terklasifikasi Positif** | **Terklasifikasi Negatif** |
| **Positif** | *TP (True Positive)* | *FP (False Positive)* |
| **Negatif** | *FN (False Negative)* | *TN (True Negative)* |

Keterangan dari *Confussion Matrix* adalah sebagai berikut :

* True Positive : Jumlah data positif yang terklasifikasi benar oleh sistem
* False Positive : Jumlah data positif yang terklasifikasi salah oleh sistem
* False Negative : Jumlah data negaitf yang terklasifikasi salah oleh sistem
* True Negative : Jumlah data negaitf yang terklasifikasi benar oleh sistem

Untuk menghitung nilai akurasi, dapat dilakukan menggunakan persamaan 3.2 berikut.

(3.2)

# **BAB IV**

**HASIL DAN PEMBAHASAN**

## **Hasil Penelitian**

Bab ini menjelaskan terkait hasil dari kinerja sistem deteksi dan perhitungan larva kepiting rajungan dengan metode *object detection* yang dalam hal ini menggunakan *Convolutional Neural Network* dengan algoritma YOLO. Pada penelitian ini menggunakan *Confussion Matrix* untuk evaluasi kinerja sistem, Total keseluruhan gambar yang digunakan untuk melakukan training ialah 375 gambar larva yang diambil dari berbagai resolusi, contoh gambar yang dijadikan *training* set dapat dilihat pada **Gambar 4.1.**



****

**Gambar 4.1.** gambar larva untuk data training

Parameter yang digunakan pada penelitian ini ialah *batch*, *subdivision*, *width*, *height*, *channels*, *learning rate*, *steps, scale, max\_batches*, *data augmentation* (perubahan *angle* , *saturation*, *exposure*, serta *hue* pada gambar data *training*). Nilai yang digunakan oleh parameter-parameter diatas merupakan nilai default seperti berikut :

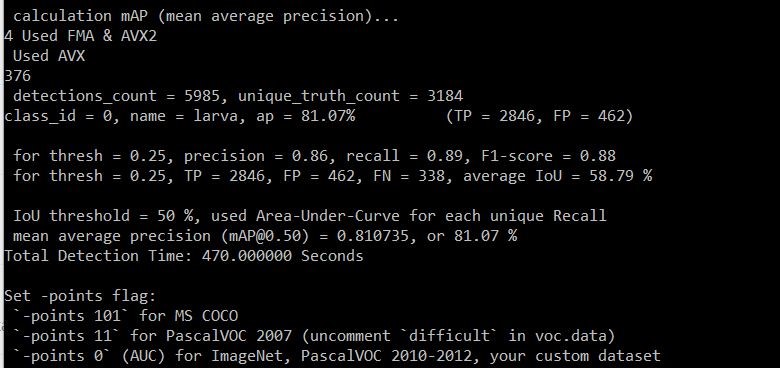
* Batch = 64
* Subdivisions = 16
* Width = 416
* Height = 416
* Momentum = 0.09
* Channel = 3
* Learning rate = 0.001
* Steps = 1600, 1800
* Scale = 0.1, 0.1
* Max\_batches = 2000

Adapun hasil dari evaluasi kinerja sistem menggunakan parameter-parameter diatas dilakukan menggunakan confussion matrix. Sistem kemudian di uji coba kepada 10 gambar larva yang memiliki ukuran resolusi yang berbeda, jumlah larva yang berbeda dan objek larva dengan tahapan yang berbeda. Hasil dari Pengujian sistem dapat dilihat pada **Tabel 4.1.**

**Tabel 4.1.** Hasil pengujian sistem

|  |  |  |
| --- | --- | --- |
| Gambar | Jumlah larva pada gambar | Jumlah larva yang terdeteksi |
| Gambar 1 | 61 larva | 63 larva |
| Gambar 2 | 43 larva | 44 larva |
| Gambar 3 | 59 larva | 61 larva |
| Gambar 4 | 33 larva | 16 larva |
| Gambar 5 | 13 larva | 11 larva |
| Gambar 6 | 42 larva | 40 larva |
| Gambar 7 | 53 larva | 45 larva |
| Gambar 8 | 11 larva | 9 larva |
| Gambar 9 | 10 larva | 12 larva |
| Gambar 10 | 79 larva | 72 larva |

Pada **Gambar 4.1** dapat dilihat hasil dari proses perhitungan akurasi model yang di training menggunakan 375 gambar dan 2000 iterasi



**Gambar 4.1.** Perhitungan akurasi model

Berdasarkan **Gambar 4.1.** dapat dilihat bahwa akurasi dari sistem atau biasa disebut dengan mAp (*mean average precision*)adalah sebesar 0.810735 atau 81.07%. Meskipun begitu, dapat dilihat pada **Tabel 4.1.** setelah sistem di ujicoba, sistem bahkan dapat mendeteksi dan menghitung jumlah larva dengan rata-rata akurasi hingga 95%.

## **Pembahasan**

Berikut penjelasan dari parameter-parameter yang digunakan untuk membuat sistem pada penelitian ini

### *Batch*

Parameter ini merupakan sebuah nilai yang menunjukkan berapa banyak data yang akan di proses oleh 1 waktu dan disebar ke *neural network*. Dalam hal ini nilai yang di gunakan adalah 64 yang artinya sistem akan mengambil 64 gambar pertama dari *training set* setiap waktu dan akan di proses di *neural network* kemudian setelah selesai diproses, sistem akan mengambil 64 gambar kedua dan dilakukan terus menerus hingga habis dan diulang lagi sesuai jumlah iterasi. Mengapa hal ini dilakukan?? Ini dikarenakan computer tidak dapat memproses data yang banyak dalam satu waktu sehingga dibutuhkan penyebaran data agar computer dapat melakukan training.

### *Max Batches (Iteration)*

Parameter ini menunjukkan berapa banyak pengulangan yang akan dilakukan oleh sistem selama training berlangsung hingga file bobot dapat di hasilkan. Tidak ada nilai pasti untuk parameter ini, namun referensi dari jumlah parameter *Max Batches* dapat ditentukan dengan rumus 2000\* jumlah kelas yang ingin dideteksi. Karena jumlah kelas yang ingin dideteksi pada penelitian ini hanya 1 yaitu larva, maka nilai parameter ini adalah 2000.

### *Subdivision*

Parameter ini menunjukkan nilai yang digunakan untuk membagi nilai batch, walaupun gambar yang diproses oleh sistem sudah dibatasi dengan menggunakan parameter batch size yang bernilai 64, tetap saja komputer akan kesulitan untuk memproses data sehingga akan muncul eror pada GPU yang disebabkan oleh penggunaan memori yang berlebihan. Maka dari itu data sebanyak 64 ini akan di bagi kembali dengan memasukkan nilai subdivision sebesar 16 maka sistem akan memproses 4 gambar dalam 1 waktu.

### *Width, Height dan Channel*

Parameter-parameter ini menunjukkan ukuran dari gambar yang digunakan untuk melakukan pelatihan. Nilai dari parameter *Width* dan *Height* adalah nilai resolusi gambar saat akan di *resize* ke resolusi 416x416 piksel. Nilai parameter *Channel* adalah 3 yang artinya gambar yang digunakan untuk melakukan pelatihan adalah gambar dengan 3 channel yaitu RGB (*Red*, *Green*, *Blue*)

### *Learning Rate, Steps,* dan *Scale*

Parameter *learning* *rate* menunjukkan sebarapa agresif sistem akan melakukan pembelajaran saat melakukan pelatihan. Nilai yang di terapkan ialah 0.001 (biasanya bernilai 0.001-0.0001), namun selama pelatihan berjalan dan sistem mengenali semakin banyak gambar maka nilai *learning* *rate* akan semakin berkurang. Nilai *steps* yang digunakan adalah 1600 dan 1800 yang artinya salama 1600 iterasi, nilai *learning rate* akan konstan dan setelah itu akan dikalikan dengan nilai *scale* yang dalam hal ini adalah 0.1.

### *Data Augmentation (angle, saturation, exposure, hue)*

Saat melakukan pelatihan, sistem membutuhkan banyak data *training*, namun untuk melakukan hal itu tentunya dibutuhkan usaha yang berlebih untuk mengumpulkan gambar lalu melabeli gambar satu persatu. Data augmentation berperan dalam hal menduplikasi data *training* yang telah dikumpulkan lalu memberikan sedikit perubahan seperti mengubah sudut gambar dengan melakukan rotasi, merubah warna dengan mengkonfigurasi *saturation*, *exposure*, dan *hue*.