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

**Muh. Arief Wicaksono1, Muhammad Niswar2, Intan Sari Areni3**

Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin

Jl. Poros Malino, Gowa, Sulawesi Selatan, Indonesia, Indonesia

Ariefw2211@gmail.com, mniswar@gmail.com, [intanareni@gmail.com](mailto:intanareni@gmail.com)

**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 melakukan 2000 iterasi ialah sebesar 86.16%.

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

1. **Pendahuluan**
   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 larva rajungan masih rendah yaitu 25–30%. Saat ini teknologi pembenihan rajungan dikembangkan 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. **Rumusan Masalah**

Rumusan masalah yang akan diuraikan dalam penelitian ini antara lain :

1. Bagaimana cara mendeteksi dan menghitung larva rajungan dengan kamera di atas air menggunakan metode object detection ?
2. Bagaimana kinerja sistem deteksi dan perhitungan jumlah larva rajungan dalam suatu kolam pemeliharaan?
   1. **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. **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.
4. **Tinjauan Pustaka**
   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. Contoh kepiting rajungan dapat dilihat pada **Gambar 2.1.**



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

Dengan melihat warna dari karapas dan jumlah duri pada karapasnya, maka dengan mudah 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.

* 1. **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 pengguna telepon genggam (*handphone* atau HP). Tetapi, saat ini orang tidak hanya dapat 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 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.



1. Lena



(b) Kapal

**Gambar 2.2.** (a) Citra Lena dan (b) Citra 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*.

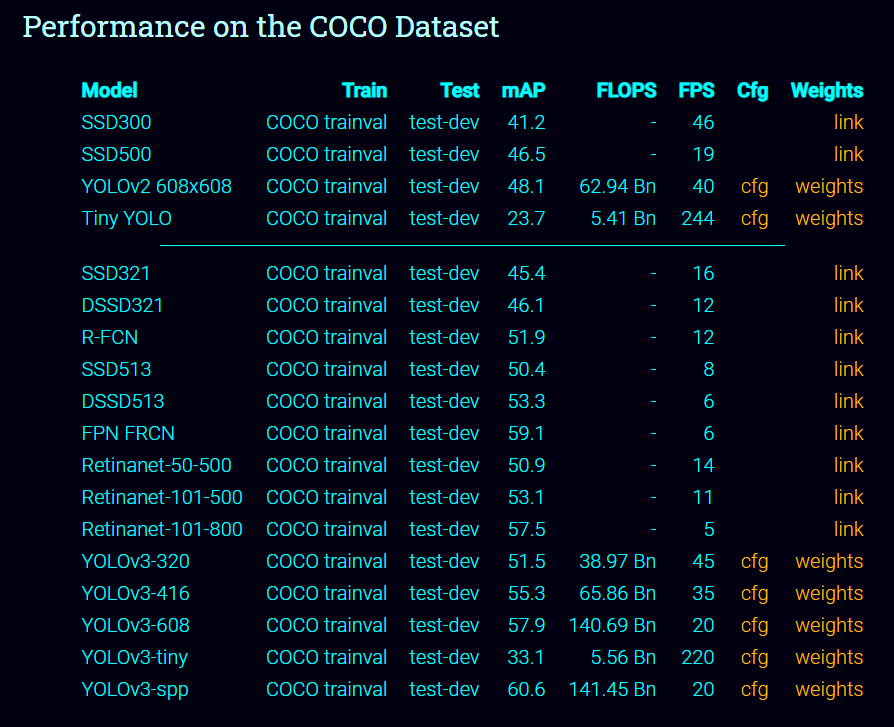
* 1. ***Machine Learning***

*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. misalnya, apakah penjualan akan meningkat apabila biaya marketing 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.

* 1. **Algoritma YOLO V3 (*You Only Look Once*)**

YOLO merupakan sebuah algoritma deep learning yang digunakan untuk mendeteksi objek secara real-time dengan cepat. YOLO mmeiliki 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)



**Gambar 2.3.** Perbandingan kinerja deteksi pada *COCO* *Dataset* (pjreddie.com, 2018)

Pada **Gambar 2.3** 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)

1. **Metodologi Penelitian**
   1. **Tahapan Penelitian**

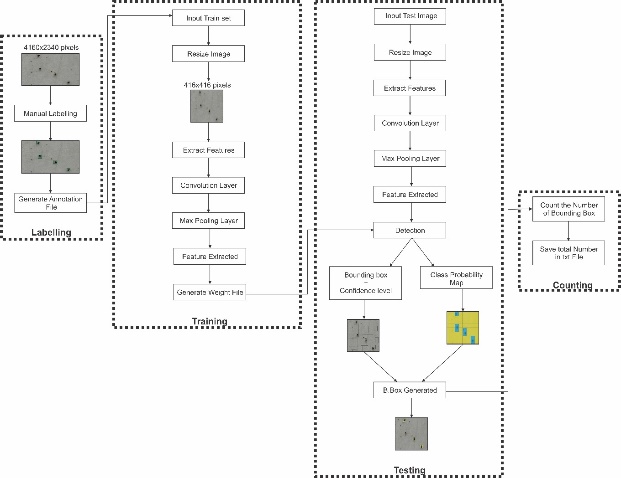
**Gambar 3.1** Diagram Tahapan Penelitian

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

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahapan ini di lakukan untuk mengumpulkan 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 diberi 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.
   1. **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. Sedangkan untuk pengolahan data di lakukan di laboratorium UBICON fakultas Teknik universitas hasanuddin.

* 1. **Perancangan Sistem**



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

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

* + 1. ***Labelling 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.

* + 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.

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

Pada model *Convolutional Neural Network* , setiap gambar masukan akan melalui beberapa tahapan *convolutional layer* dengan *filter*, *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.*

* + 1. ***Classification***

Matriks hasil dari *Max pooling layer* akan di *reshape* menjadi bentuk vektor yang kemudian di masukkan ke dalam *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 *Binary Classification*. Kemudian *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.

* + 1. ***Visualization***

Untuk menghasilkan gambar yang telah di proses oleh sistem, digunakan OpenCV, yang dimana berfungsi untuk mempercepat proses komputasi data dan untuk menggambar *bounding* *box*. Untuk menggambar *bounding* *box*, YOLO menggunakan 2 pendekatan yaitu *confidence* *Level* yang dimana sistem akan membagi gambar menjadi S x S *grid* *cell*, *grid* *cell* tersebut bertanggungjawab untuk mendeteksi objek. Setiap *grid* *cell* akan memprediksikan *bounding* *box* sebanyak B dan memiliki *confidence* *score* untuk setiap kotaknya. *Confidence* *score* merepresentasikan seberapa tinggi tingkat kepercayaan sistem bahwa kotak tersebut mengandung objek yang ingin dideteksi dan seberapa akurat sistem memprediksi objeknya. Pendekatan kedua ialah *class* *probability* *map*, setiap *bounding* *box* terdiri dari 5 prediksi : *x,y,w,h,* dan *confidence*. (x,y) merepresentasikan kordinat titik tengah dari *bounding* *box* *relative* terhadap *grid* *cell*. w dan h merepresentasikan lebar dan tinggi dari keseluruhan gambar. *Confidence* merepresentasikan IOU antara *bounding* *box* yang di prediksi dengan *bounding* *box* yang asli.

* + 1. ***Counting***

Proses perhitungan dilakukan dengan mendeteksi jumlah *bounding box* yang telah divisualisasikan pada gambar*. 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)* |

Untuk menghitung nilai akurasi, dapat dilakukan menggunakan persamaan berikut.

1. **Hasil dan Pembahasan**
   1. **Hasil Penelitian**

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 dan di bagi menjadi 80% data *training* dan 20% data *test* atau sekitar 300 data *training* dan 75 data *testing.*

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 di atas merupakan nilai *default* seperti berikut :

* Batch = 64
* Subdivisions = 32
* Width = 416
* Height = 416
* Channel = 3
* Learning rate = 0.001
* Steps = 1600, 1800
* Scale = 0.1, 0.1
* Max\_batches = 2000

*Training* awal dilakukan sebanyak 2000 iterasi, kemudian ditambah lagi sebanyak 2000 iterasi agar dapat mengevaluasi nilai akurasi dari file bobot yang dihasilkan. **Tabel 4.1** menunjukkan hasil dari evaluasi mAP sistem menggunakan *confussion matrix.*

**Tabel 4.1.** Evaluasi nilai mAP sistem menggunakan *confussion matrix*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jumlah Iterasi | mAP | *True Positive* | *False Negative* | *False Positive* | *True Negative* |
| 1000 | 70.12% | 419 | 202 | 96 | 0 |
| 2000 | 86.16% | 553 | 68 | 75 | 0 |
| 3000 | 63.73% | 367 | 49 | 254 | 0 |
| 4000 | 64.14% | 389 | 232 | 44 | 0 |

Sistem dengan iterasi sebanyak 2000 merupakan yang paling baik akurasinya, sehingga sistem dengan 2000 iterasi kemudian di uji coba pada 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 4.2.**

**Tabel 4.2.** Hasil pengujian sistem

|  |  |  |
| --- | --- | --- |
| Data | Jumlah larva pada gambar | Jumlah larva yang terdeteksi |
| Data 1 | 13 larva | 11 larva |
| Data 2 | 42 larva | 40 larva |
| Data 3 | 53 larva | 45 larva |
| Data 4 | 11 larva | 9 larva |
| Data 5 | 79 larva | 72 larva |

Sistem kemudian diujikan kepada objek yang berbeda untuk melihat apakah sistem dapat mendeteksi objek lain yang mirip dengan kumpulan larva. Onbjek yang digunakan pada pengujian ini ialah 2 gambar kumpulan semut, 2 gambar kumpulan rayap 2 gambar kumpulan manusia, dan 2 gambar kumpulan larva ikan. Hasil dari deteksi sistem pada objek yang berbeda dapat dilihat pada **Tabel 4.3.**

**Tabel 4.3.** Hasil pengujian sistem pada objek yang berbeda

|  |  |  |
| --- | --- | --- |
| Data | Gambar | Hasil |
| Data 1 | Kumpulan Semut | Terdeteksi |
| Data 2 | Kumpulan Semut | Terdeteksi |
| Data 3 | Kumpulan Rayap | Tidak Terdeteksi |
| Data 4 | Kumpulan Rayap | Tidak Terdeteksi |
| Data 5 | Kumpulan Manusia | Tidak Terdeteksi |
| Data 6 | Kumpulan Manusia | Tidak Terdeteksi |
| Data 7 | Kumpulan Larva Ikan | Tidak Terdeteksi |
| Data 8 | Kumpulan Larva Ikan | Terdeteksi |

* 1. **Pembahasan** 
     1. **Hasil Pengujian Sistem**

Sistem diujicoba dengan mendeteksi 5 data gambar larva yang memiliki resolusi berbeda, dan hasilnya sistem dapat mendeteksi keberadaan larva pada tiap gambar, namun sistem masih memiliki kendala pada akurasi sehingga sistem terkadang masih miss dalam melakukan deteksi. Sistem juga masih dapat mendeteksi objek yang bukan larva dan dikenali sebagai larva sehingga masih perlu dilakukan training dengan datayang lebih banyak.

* + 1. **Hasil Pengujian Sistem pada objek yang berbeda**

Sistem kemudian di ujicoba pada objek yang berbeda dan dilakukan pengujian, terdapat total 8 gambar yang digunakan dan di ujicoba oleh sistem. Hasil pengujiannya ialah sistem salah memprediksi gambar semut dan larva ikan sebagai larva rajungan, sehingga masih dibutuhkan pengujian lebih lanjut pada ekstrasi fitur saat dilakukan training.

1. **Penutup**
   1. **Kesimpulan**

Berdasarkan hasil pengujian dari implementasi sistem deteksi dan perhitungan lava kepiting rajungan dengan metode object detection yang kemudian di analisa dan disimpulkan bahwa :

1. Sistem deteksi dan perhitungan larva kepiting rajungan ini menggunakan algoritma deep learning YOLO V3 untuk mendeteksi objek. Gambar di ambil menggunakan kamera dan dikumpulkan lalu diberi label secara manual. Gambar yang telah di beri label akan di latih untuk menghasilkan sebuah file bobot. Metode yang digunakan untuk mengekstraksi fitur dan kalsifikasi adalah Convolutional Neural Network yang didalamnya terdapat 106 layer. Hasil dari deteksi program akan di visualisasi dengan secara otomatis menggambarkan bounding box di sekitar objek yang terdeteksi pada gambar. Bounding box kemudian akan dihitung untuk dapat menentukan hasil dari seluruh jumlah larva pada gambar.

2. Sistem menghasilkan nilai akurasi sebesar 86.16% dengan nilai parameter batch 64, subdivision 32, width 416, height 416, channels 3, learning rate 0.001, steps 1600 dan 1800, scale 0.1,serta max\_batches atau jumlah iterasinya mencapai 2000.

* 1. **Saran**

Saran dari penulis untuk pengembangan sistem ini yang dapat dilakukan kedepannya adalah sebagai berikut :

1. Sistem yang telah dibuat masih dapat ditingkatkan akurasinya dengan menambahkan data training hingga mencapai 2000 gambar/class. File konfigurasi YOLO juga dapat di ubah untuk meningkatkan akurasi, seperti merubah nilai parameter width dan height, max batch, layer, stride, dan nilai learning rate.
2. Sistem ini dapat diimplementasi dalam berbagai bidang sehingga diharaokan tidak hanya untuk bidang aquaculture, tapi sistem ini dapat diterapkan di bidang lain peternakan, pertanian, kedokteran,dll dengan mengubah data training nya.
3. Framework yang digunakan pada sistem ini adalah darknet, namun masih ada framework lain seperti Tensorflow atau Darkflow (gabungan dari darknet dan tensorflow) yang dapat digunakan untuk mempermudah pemrosesan dari sistem.
4. **Daftar Pustaka**

|  |  |
| --- | --- |
| [1] | J. Redmon, S. D. Vala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv,* 2016. |
| [2] | J. Redmon and A. Farhadi, "YOLOV3: An Incremental Impovement," *arXiv,* 2018. |
| [3] | V. Raman, S. Perumal, S. Navaratnam and S. Fazilah, "Computer Assisted Counter System for Larvae and Juvenile Fish in Malaysian Fishing Hatcheries by Machine Learning Approach," *Journal of Computers,* pp. 423-431, 2015. |
| [4] | J. W. G. Putra, Pengenalan Konsep Pembelajaran Mesin dan Deep Learning, Tokyo, 2019. |
| [5] | R. Munir, Pengolahan Citra Digital Dengan Pendekatan Algoritmik, Bandung: INFORMATIKA, 2004. |
| [6] | J. Kaewchote, S. Janyong and W. Limprasert, "Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp," *Agriculture and Natural Resources,* 2018. |
| [7] | G. French, M. H. Fisher, M. Mackiewicz and C. Needle, "Convolutional Neural Networks for Counting Fish in Fisheries Surveillance Video," *Workshop on Machine Vision of Animals and their Behaviour,* 2015. |
| [8] | A. N. Fierro, D. R. Camarillo and M. N. Miyatake, "Mosquito larva classification method based on convolutional neural networks," *International Conference on Electronics, Communications and Computers (CONIELECOMP),* 2017. |
| [9] | J. H. Almarinez and A. Hernandez, "Evaluation of Mangrove Crab Classification System," *International Journal of Recent Technology and Engineering (IJRTE),* pp. 2277-3878, 2019. |
| [10] | I. Aliyu, K. J. Gana, A. A. Musa, J. Agajo, A. M. Orire, F. T. Abiodun and M. A. Adegboye, "A PROPOSED FISH COUNTING ALGORITHM USING DIGITAL IMAGE PROCESSING," *Journal of Science, Technology & Education (JOSTE),* vol. V, 2017. |