PENGHITUNGAN OTOMATIS LARVA UDANG MENGGUNAKAN MENTODE YOLO

AUTOMATIC COUNTING OF SHRIMP LARVAE BASED YOU ONLY LOOK ONCE (YOLO)

SISKA ARMALIVIA

PROGRAM PASCASARJANA
UNIVERSITAS HASANUDDIN
MAKASSAR
2021

PENGHITUNGAN OTOMATIS LARVA UDANG MENGGUNAKAN METODE YOLO

Tesis

Sebagai Salah Satu Syarat Untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan diajukan oleh

SISKA ARMALIVIA

D032181026

Kepada

PROGRAM PASCASARJANA
UNIVERSITAS HASANUDDIN
MAKASSAR
2021

HALAMAN PENGESAHAN

Judul Lama : Implementasi Pengolahan Citra Counting Algorithm Untuk Perhitungan Jumlah Bibit Dan Ukuran Post Larva Benur Udang

Judul Thesis : Penghitungan Otomatis Larva Udang Menggunakan Metode Yolo

Nama : Siska Armalivia

NIM : D032181026

Jurusan : Teknik Elektro

Konsentrasi : Teknik Informatika

Diajukan sebagai salah satu syarat akademik pada program Pascasarjana Universitas Hasanuddin Makassar.

Menyetujui

Pembimbing I,	Pembimbing II,
Dr. Ir. Zahir Zainuddin, M.Sc.	Prof. Dr. Ir. Andani Achmad, M.T.
Nip. 196404271989101002	Nip. 196012311987031022

Makassar, 17 Juni 2021

Ketua Program Studi S2 Teknik Elektro

Prof. Dr. Eng. Syafaruddin, S.T.,M.Eng.

Nip. 197405301999031003

PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini:

Nama	:	Siska Armalivia
Nomor Pokok	:	D032181026
Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tisis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 4 April 2021

Yang menyatakan,

Siska Armalivia

ABSTRAK

SISKA ARMALIVIA. Penghitungan Otomatis Larva Udang Menggunakan Metode YOLO (dibimbing oleh Zahir Zainuddin dan Andani Achmad).

Penelitian ini bertujuan untuk menghitung otomatis jumlah larva udang menggunakan metode You Only Look Once (YOLO) generasi ke-3. Hal ini disebabkan penghitungan larva saat ini masih dilakukan secara manual, yaitu menggunakan metode sampling, dengan cara mengambil satu cangkir penuh larva udang dan dihitung secara manual sehingga banyak memakan waktu dan sering kali menimbulkan kesalahan manusia. Metode YOLOv3 digunakan karena dapat memprediksi suatu objek dengan lebih cepat dan memiliki tingkat keakuratan yang tinggi. Pada penelitian ini digunakan data latih dan data uji masing-masing sebanyak 325 dan 99 data gambar. Pengambilan data dilakukan di Balai Perikanan Budidaya Air Payau Takalar, Kabupaten Takalar. Pengambilan gambar dilakukan menggunakan kamera yang tersimpan diatas wadah putih berisikan air 2 cm dan larva udang dengan menggunakan sistem backlight agar tidak ada pantulan cahaya dari dalam air. Pengujian dilakukan dengan membandingkan hasil perhitungan menggunakan sistem dengan penghitungan manual. Hasil penelitian menunjukkan bahwa sistem mampu mengidentifikasi jumlah udang dengan rata-rata akurasi penghitungan sebesar 96.18%.

Kata kunci: larva udang, penghitungan objek, deteksi objek, YOLO

ABSTRACT

SISKA ARMALIVIA. Automatic Counting of Shrimp Larvae Based YOLO (supervised by Zahir Zainuddin and Andani Achmad).

This study aims to automatically count the number of shrimp larvae using the 3rd generation You Only Look Once (YOLO) method. This is because the counting of larvae is currently carried out manually, using the sampling method, by taking a full cup of shrimp larvae and counting them manually, which is time-consuming and often leads to human error. The YOLOv3 method is used because it can predict an object more quickly and has a high degree of accuracy. In this study, 325 and 99 image data were used, respectively, of the training data and test data. Data were collected at the Takalar Brackish Water Cultivation Fishery Center, Takalar Regency. Taking pictures is done using a camera stored on a white container containing 2 cm of water and shrimp larvae using a backlight system so that there is no light reflected from the water. Testing is done by comparing the results of calculations using a system with manual calculations. The results showed that the system was able to identify the number of shrimp with an average counting accuracy of 96.18%

Keywords: shrimp larvae, object counting, object detection, YOLO.

KATA PENGANTAR

Segala puji selalu dipanjatkan kepada Allah SWT Yang Maha Kuasa yang telah memberikan rahmat, hidayah dan pertolongan-Nya dalam menyelesaikan tesis, yang berjudul **"Penghitungan Otomatis Larva Udang Menggunakan Metode YOLO."** Tak lupa pula shalawat dan salam kepada Nabi Muhammad SAW yang telah menyinari dunia ini dengan keindahan ilmu dan akhlak yang diajarkan kepada seluruh umatnya.

Ucapan terima kasih pun penulis hanturkan kepada dosen pembimbing tesis Dr. Ir. Zahir Zainuddin, M.Sc. dan Prof. Dr. Ir. Andani Achmad, M.T. yang telah meluangkan waktunya untuk membimbing dan berkonsultasi tentang materi dalam tesis ini dan juga kepada seluruh dosen dan staf Departemen Teknik Elektro, Universitas Hasanuddin yang telah membantu dalam hal keilmuan maupun administrasi pada tahap tesis ini. Tak lupa penulis juga mengucapkan banyak terima kasih kepada yang terkasih, suami tercinta, yang selalu memberi dukungan dan motivasi dalam penyusunan tesis ini. Penulis menyadari bahwa tesis ini masih belum sempurna. Dengan demikian, penulis tetap mengharapkan kritik dan saran dengan harapan tulisan ini bisa memberikan manfaat kepada seluruh pihak.

Makassar, 4 April 2021

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN ii

PERNYATAAN KEASLIAN TESIS iii

ABSTRAK iv

ABSTRACT v

KATA PENGANTAR vi

DAFTAR ISI vii

DAFTAR TABEL x

DAFTAR GAMBAR xi

BAB I 1

PENDAHULUAN 1

A. LATAR BELAKANG 1

B. RUMUSAN MASALAH 3

C. TUJUAN PENELITIAN 3

D. MANFAAT PENELITIAN 3

E. BATASAN MASALAH 4

F. SISTEMATIKA PENULISAN 4

BAB II 6

TINJAUAN PUSTAKA 6

A. LANDASAN TEORI 6

1. Udang 6

2. *Image Processing* 12

3. Visi Komputer 13

4. YOLO 15
5. Google Colaboratory 21
6. *Image Annotation* 22
B. PENELITIAN TERKAIT 25
C. STATE OF THE ART 26
D. KERANGKA PIKIR 28
BAB III 29
Metodologi Penelitian 29
A. TAHAPAN PENELITIAN 29
B. WAKTU DAN LOKASI PENELITIAN 30
C. INSTRUMEN PENELITIAN 30
D. TEKNIK PENGAMBILAN DATA 31
E. PERANCANGAN SISTEM 32
1. Pelabelan Data 32
2. Training 34
3. Testing 37
F. ANALISIS KERJA SISTEM 39
BAB IV 42
HASIL DAN PEMBAHASAN 42
A. HASIL PENELITIAN 42
B. PEMBAHASAN 46
BAB V 52
Penutup 52
A. KESIMPULAN 52
B. SARAN 52

DAFTAR PUSTAKA 54

LAMPIRAN

DAFTAR TABEL

Tabel 1. *State of the art* 27
Tabel 2. Proses konvolusi 35
Tabel 3. Confusion Matrix 42
Tabel 4. Hasil Perbandingan 42
Tabel 5. Perhitungan RMSE kepadatan < 50 44
Tabel 6. Perhitungan RMSE kepadatan > 50 45

DAFTAR GAMBAR

Gambar 1. Grafik produksi udang 6
Gambar 2. Grafik produsen udang terbesar di Indonesia 7
Gambar 3. siklus hidup udang 10
Gambar 4. Digital Image 14
Gambar 5. YOLO 16
Gambar 6. Darknet-19 18
Gambar 7. Bounding box 19
Gambar 8. Model layer network darknet 53 20
Gambar 9. Tampilan Google Colaboratory 21
Gambar 10. Bounding Box 22
Gambar 11. 3D cuboids 23
Gambar 12. Polygonal segmentation 23
Gambar 13. Semantic segmentation 23
Gambar 14. Key-point annotation 24
Gambar 15. Line 24
Gambar 16. Image Classification 24
Gambar 17. Kerangka Pikir 28
Gambar 18. Diagram tahapan penelitian 29
Gambar 19. Pengambilan data citra 31
Gambar 20. Data latih 31
Gambar 21. Desain alur sistem 32
Gambar 22. Flowchart proses labelling yolo_mark 33
Gambar 23. Pelabelan menggunakan yolo_mark 33
Gambar 24. File konfigurasi 34
Gambar 25. Grid 3x3 35
Gambar 26. Class probabilities 36
Gambar 27. Komponen bounding box 36
Gambar 28. Perintah *download* data *training* 37

Gambar 29. Perintah *download* model YOLO 37
Gambar 30. Perintah *training* 37
Gambar 31. Proses testing menggunakan YOLO 38
Gambar 32. Contoh hasil deteksi 38
Gambar 33. Confusion Matrix 39
Gambar 34. Contoh TP, FP dan FN 40
Gambar 35. Grafik selisih 44
Gambar 36. Perhitungan manual menggunakan aplikasi *paint* 46
Gambar 37. Proses penghitungan larva udang 47
Gambar 38. Hasil perhitungan pada sistem 47
Gambar 39. Hasil deteksi sistem 51

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Udang merupakan salah satu komoditas primadona di sub sektor perikanan yang diharapkan dapat meningkatkan devisa negara. Permintaan pasar di luar negeri yang cenderung meningkat serta sumber daya yang cukup tersedia di Indonesia memberikan peluang sangat besar untuk dapat dikembangkan budidayanya. Dengan meningkatnya budidaya udang maka diperlukan ketersediaan benur secara kontinyu dan berkualitas, sehingga ketersediaan benur tersebut diharapkan mampu meningkatkan produktifitas udang (Haliman, 2005).

Saat ini dengan perkembangan teknologi membuat manusia ingin meningkatkan efektifitas dan efisiensi di segala aspek kehidupan, tak terkecuali dalam bidang citra digital. Pengolahan citra merupakan teknologi bidang citra yang berkembang pesat yang telah banyak diterapkan pada ilmu-ilmu murni dan teknik. Proses pengolahan citra mempunyai data masukan dan informasi keluaran bentuk citra, sehingga pengolahan citra adalah pemrosesan citra yang telah ada untuk menghasilkan citra yang lebih tinggi kualitasnya.

Analisa suatu citra untuk menggali informasi yang terkandung di dalamnya dapat dilakukan menggunakan pengolahan citra digital. Citra dalam kondisi kurang maksimal untuk di analisa seperti terdapat banyak noise dapat diatasi dengan pengolahan citra digital. Salah satu contoh analisa pada citra adalah dengan menghitung objek yang ada pada suatu citra, seperti menghitung jumlah sel darah merah. Dengan mengetahui jumlah sel darah merah, penyakit yang diderita pasien akan dapat diketahui dengan analisa lebih lanjut.

Penghitungan objek telah dilakukan di berbagai bidang industri, lembaga penelitian, laboratorium, industri pertanian dan bidang-bidang yang lain. Metode perhitungan konvensional biasanya menggunakan penghitungan manual yang selama ini sangat tergantung

pada keahlian dan jam terbang pelaku dari bidang tersebut. Kelelahan mata, bentuk yang kecil dan jumlah yang banyak seringkali menjadi faktor yang menyebabkan tingkat akurasi penghitungan berkurang. Oleh karenanya, *image processing* kemudian dijadikan sebagai salah satu solusi untuk melakukan penghitungan secara komputasi.

Penelitian terkait tentang mendeteksi dan penghitungan objek antara lain dilakukan oleh (Awalludin et al., 2020) (Awalludin et al., 2019) melakukan penghitungan larva ikan dan udang menggunakan metode tepi sobel dan *Blob Analysis*. (Loh et al., 2011) menggunakan prototipe *Aquatic Tool Kit* untuk menghitung larva dan juvenil ikan. (Flores et al., 2008) melakukan penghitung larva kerang menggunakan teknik pengolahan citra. (Clarke et al., 2018) menghitung larva siput menggunakan *Image Processing Pipeline*. (Lainez & Gonzales, 2019), (Nurlaela et al., 2019) melakukan penghitungan larva ikan dan larva kepiting menggunakan metode CNN.

Saat ini pemanfaatan teknologi *machine vision* telah secara luas dipergunakan untuk mengganti fungsi penglihatan pada manusia. Penggunaan kamera CCD atau kamera digital pada berbagai bidang terbukti mampu menggantikan menggantikan fungsi indera penglihatan manusia. Kelebihan teknologi tersebut adalah mampu melakukan pekerjaan secara cepat dan konsisten. Teknologi tersebut berpeluang besar untuk dikembangkan karena murah, tersedia dan mudah dibuat.

Sampai saat ini proses penghitungan benih udang masih dilakukan secara manual. Penghitungan benih ikan atau udang yang dilakukan dengan cara manual memiliki banyak kelemahan, antara lain: subyektifitas penghitungan, waktu yang lambat, kelelahan dalam penghitungan, serta akurasi yang tidak memadai khususnya untuk menghitung benih ikan atau udang dalam jumlah yang besar (SEMINAR, 2000).

Pengembangan teknologi berbasis pengolahan citra dengan kamera sebagai penangkap gambar sangat mungkin dikembangkan pada proses penghitungan benih udang. Dengan

teknologi berbasis pengolahan citra diharapkan mampu membantu penghitungan benih udang dan ukuran post larva secara cepat, akurat dan murah.

Oleh karena itu, berdasarkan beberapa penelitian diatas, penulis mengajukan sebuah sistem "Penghitungan Otomatis Larva Udang Menggunakan Metode YOLO". Diharapkan dengan adanya sistem ini mampu membantu para petani tambak udang dalam penghitungan benih udang secara cepat dan akurat, serta dapat meningkatkan penghasilan dan produksinya

B. RUMUSAN MASALAH

Berdasarkan uraian latar belakang diatas maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengimplementasikan algoritma YOLO untuk mendeteksi dan menghitung larva udang?
2. Bagaimana akurasi yang dihasilkan algoritma YOLO dalam mendeteksi dan mengitung larva udang?

C. TUJUAN PENELITIAN

Adapun tujuan yang akan dicapai pada penelitian ini adalah

1. Menerapkan algoritma YOLO dalam penghitungan larva udang.
2. Untuk mengetahui akurasi yang dihasilkan menggunakan algoritma YOLO dalam mendeteksi dan menghitung larva udang.

D. MANFAAT PENELITIAN

Manfaat dari penelitian ini adalah :

1. Bagi Masyarakat
    a. Penelitian ini dapat membantu penambak udang untuk meningkatkan produktifitas pertambakannya.
    b. Penelitian ini bermanfaat memberikan kemudahan bagi penambak udang dalam menghitung jumlah benih yang dijual agar tidak mengalami kerugian.

c. Penelitian ini bermanfaat memberikan kemudahan bagi penambak udang dalam menghitung panjang post larva agar dapat menghasilkan udang yang berkualitas.

2. Bagi peneliti, penelitian ini berguna untuk menambah pengetahuan dan kemampuan mengenai penghitungan objek dalam sebuah gambar.
3. Bagi institusi pendidikan Magister Jurusan Teknik Elektro Konsentrasi Teknik Informatika, dapat digunakan sebagai referensi ilmiah dalam penelitian untuk pengembangan perhitungan objek dengan *image processing*.

E. BATASAN MASALAH

Batasan masalah penelitian adalah:

1. Data yang diperoleh berupa data gambar dengan format JPEG.
2. Metode yang digunakan adalah YOLOv3.
3. Memberi batasan pada objek yang menumpuk.
4. Benih udang yang digunakan adalah benih udang yang telah mengalami pergantian kulit.
5. Dalam pengambilan gambar, benih udang diletakkan ke dalam sebuah wadah dengan latar warna yang jelas dan tidak ada kontak langsung dengan cahaya matahari
6. Pengambilan gambar menggunakan kamera *Smartphone iPhone 5s dan Redmi Note* 8

F. SISTEMATIKA PENULISAN

Sistematika penulisan pada penelitian ini adalah:

Bab I Pendahuluan

Bab I berisi penjelasan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian serta sistematika penulisan

Bab II Landasan Teori dan Kerangka Pemikiran

Bab II berisi penjelasan tentang landasan teori yang digunakan dalam penelitian dan kerangka pemikiran. Landasan teori merupakan suatu penjelasan tentang sumber acuan terbaru dari pustaka primer seperti buku, artikel, jurnal, prosiding dan tulisan asli lainnya untuk

mengetahui perkembangan penelitian yang relevan dengan judul atau tema penelitian yang dilakukan dan juga sebagai arahan dalam memecahkan masalah yang diteliti. Dalam bab ini juga diuraikan tentang kerangka pikir tentang masalah, metode, pengukuran dan hasil dari penggunaan metode yang sesuai dengan objek dalam penelitian yang diusulkan.

Bab III Metodologi Penelitian

Bab III ini merupakan penjelasan tentang metode penelitian, penentuan masalah, penentuan computing approach, juga penjelasan bagaimana pengembangan dan penerapan software dengan computing approach pada obyek penelitian, diuraikan pula cara evaluasi dan validasi hasil penerapan, metode pengumpulan data, metode analisis data, metode pengukuran penelitian, penerapan computing approach pada masalah penelitian, pengembangan software yang menerapkan computing approach, analisa kebutuhan, konstruksi sistem dan pengujian sistem.

Bab IV Hasil dan Pembahasan

Pada bab IV ini menjelaskan tentang hasil dan pembahasan penelitian serta implikasi dari penelitian yang dilakukan. Hasil merupakan suatu penjelasan tentang data kuantitatif yang dikumpulkan dari lapangan sesuai dengan metodologi yang telah ditetapkan. Pembahasan merupakan suatu penjelasan tentang pengolahan data dan interprestasinya, baik dalam bentuk diskriptif ataupun penarikan inferensinya. Implikasi penelitian merupakan suatu penjelasan tentang tindak lanjut penelitian yang terkait dengan aspek manajerial, aspek sistem, maupun aspek penelitian lanjutan.

Bab V Kesimpulan dan Saran

Pada bab V ini berisi ringkasan temuan, rangkuman kesimpulan dan saran. Kesimpulan merupakan pernyataan secara general atau spesifik yang berisi hal-hal penting dan menjadi temuan penelitian yang bersumber pada hasil dan pembahasan. Saran merupakan pernyataan atau rekomendasi peneliti yang berisi hal-hal penting sebagaimana yang telah disampaikan.

BAB II

TINJAUAN PUSTAKA

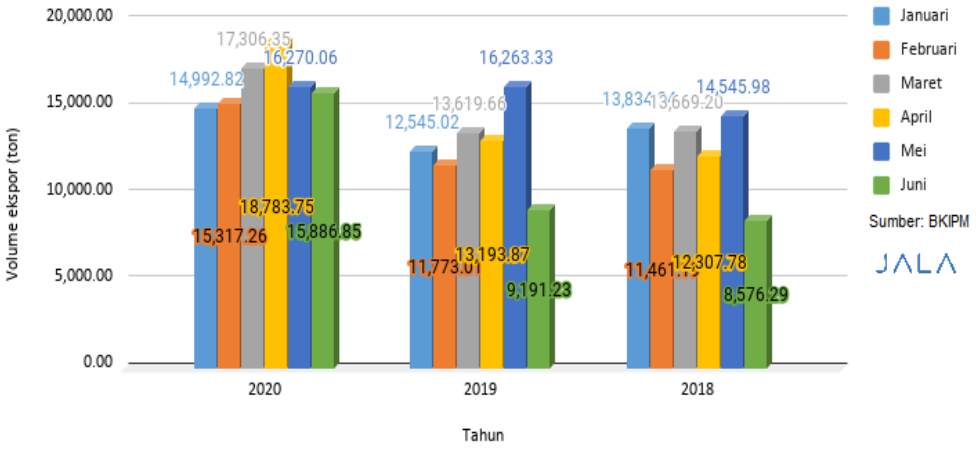
A. LANDASAN TEORI

1. Udang

a. Udang di Indonesia

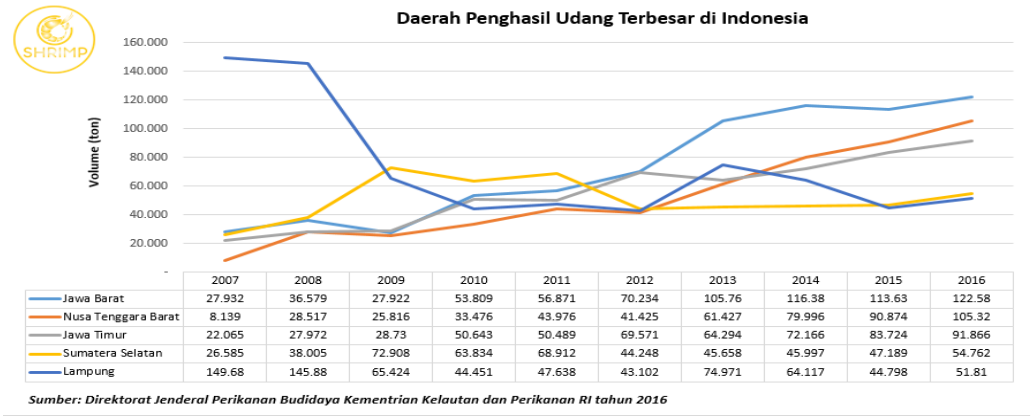
Udang menjadi salah satu komoditas ekspor di Indonesia. Jenis udang yang paling banyak dibudidayakan di Indonesia sampai dengan tahun 2016 adalah udang vaname dengan *share* produksi hampir mencapai 80% dari total produksi udang Indonesia.

Data Direktorat Jenderal Perikanan Budidaya Kementerian Kelautan dan Perikanan Republik Indonesia menyebutkan, bahwa volume udang di Indonesia pada tahun 2009 mencapai 337.930 ton. Udang vaname, windu sebesar 124.500 ton dan udang lainnya sebesar 42.530 ton. Gambar 1 menunjukkan ekspor udang di Indonesia.



Gambar 1. Grafik produksi udang (Zulfikar, 2020)

Pada gambar diatas menunjukkan tahun 2020, ekspor udang meningkat dari tahun sebelumnya. Gambar 2 menunjukkan daerah penghasil udang terbesar di Indonesia.



	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Jawa Barat	27.932	36.579	27.922	53.809	56.871	70.234	105.76	116.38	113.63	122.58
Nusa Tenggara Barat	8.139	28.517	25.816	33.476	43.976	41.425	61.427	79.996	90.874	105.32
Jawa Timur	22.065	27.972	28.73	50.643	50.489	69.571	64.294	72.166	83.724	91.866
Sumatera Selatan	26.585	38.005	72.908	63.834	68.912	44.248	45.658	45.997	47.189	54.762
Lampung	149.68	145.88	65.424	44.451	47.638	43.102	74.971	64.117	44.798	51.81

Sumber: Direktorat Jenderal Perikanan Budidaya Kementrian Kelautan dan Perikanan RI tahun 2016

Gambar 2. Grafik produsen udang terbesar di Indonesia (Oshrimp, 2020)

Pada gambar 2, dapat dilihat bahwa pada tahun 2007 hingga 2008, Lampung menghasilkan sebanyak >140.000 ton udang, namun pada tahun 2009, mengalami penurunan drastis. Sedangkan pada Jawa Barat mengalami peningkatan penghasilan sejak tahun 2012 hingga 2016.

b. Taksonomi Udang

Crustacea adalah hewan akuatik (air) yang terdapat di air laut dan air tawar. Kata *Crustacea* berasal dari bahasa latin yaitu kata Crusta yang berarti cangkang yang keras. Ilmu yang mempelajari tentang crustacean adalah karsinologi (OEMUJATI, 1990). Jumlah udang di perairan seluruh dunia diperkirakan sebanyak 343 spesies yang potensial secara komersil. Dari jumlah itu 110 spesies termasuk didalam *family Penaidae*. Udang digolongkan kedalam *Filum Arthropoda* dan merupakan Filum terbesar dalam Kingdom Animalia (Fast & Lester, 1992). Menurut (Sterrer, 1986) udang dapat diklasifikasikan sebagai berikut:

Kingdom	:	*Animalia*
Filum	:	*Arthropoda*
Kelas	:	*Crustacea*
Sub Kelas	:	*Malacostraca*
Ordo	:	*Decapoda*
Family	:	*Palaemonoidae*

		Penaeidae
Genus	:	*Macrobranchium* *Caridina* *Penaeus* *Metapenaeus*

c. Morfologi Udang

Udang merupakan jenis ikan konsumsi air payau, badan beruas berjumlah 13 (5 ruas kepala dan 8 ruas dada) dan seluruh tubuh ditutupi oleh kerangka luar yang disebut *eksosketelon*. Umumnya udang yang terdapat di pasaran sebagian besar terdiri dari udang laut. Hanya sebagian kecil saja yang terdiri dari udang air tawar, terutama di daerah sekitar sungai besar dan rawa dekat pantai. Udang air tawar pada umumnya termasuk dalam keluarga *Palaemonidae*, sehingga para ahli sering menyebutnya sebagai kelompok udang palaemonid. Udang laut, terutama dari keluarga Penaeidae, yang biasa disebut udang penaeid oleh para ahli (Menristek, 2003). Tubuh udang terbagi atas tiga bagian besar, yakni kepala dan dada, badan, serta ekor. Sedangkan persentasenya adalah 36-49% bagian kepala, daging keseluruhan 24-41% dan kulit ekor 17-23% dari seluruh berat badan, tergantung juga dari jenis udangnya (Novrihansa et al., 2016).

Ciri-ciri morfologi udang menurut (Fast & Lester, 1992), mempunyai tubuh yang bilateral simetris terdiri atas sejumlah ruas yang dibungkus oleh kintin sebagai eksoskleton. Tiga pasang *maksilliped* yang terdapat dibagian dada digunakan untuk makan dan mempunyai lima pasang kaki jalan sehingga disebut hewan berkaki sepuluh (Decapoda). Tubuh biasanya beruas dan sistem syarafnya berupa tangga tali. Dilihat dari luar, tubuh udang terdiri dari dua bagian, yaitu bagian depan dan bagian belakang. Bagian depan disebut bagian kepala, yang sebenarnya terdiri dari bagian kepala dan dada yang menyatu. Bagian kepala tertutup kerapak, bagian perut terdiri dari lima ruas yang masing-masing ruas mempunyai *pleopod* dan ruas terakhir terdiri dari ruas perut, dan ruas telson serta *uropod*

(ekor kipas). Tubuh udang mempunyai rostrum, sepasang mata, sepasang antena, sepasang antenula bagian dalam dan luar, tiga buah *maksilipied*, lima pasang *cholae (periopod)*, lima pasang *pleopod*, sepasang *telson* dan *uropod*.

d. Jenis Udang Laut Budidaya

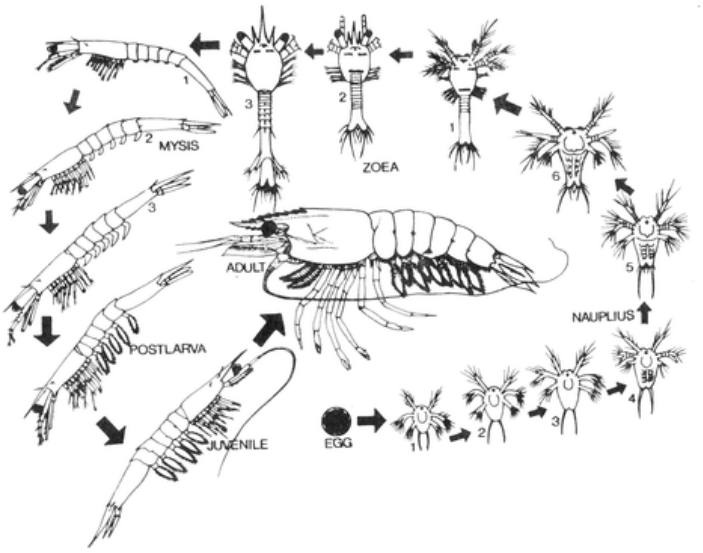
Saat ini terdapat beberapa jenis udang yang dibudidayakan di Indonesia, yaitu udang vanname *(Litopenaeus Vanname)*, udang windu *(Penaeus Mondon)*, udang putih *(P. Merguiensis dan P. Indicus),* udang rostris *(L. stylirostris),* udang api-api *(Metapenaeus sp.)*, dan udang galah *(Macrobachium rosenbergii)*. Adapun udang yang umumnya hidup di perairan Indonesia adalah udang windu, udang putih juga udang api-api.

1) Udang Vannamae

Udang vannamae atau udang putih (*Litopenaeus Vanname*) adalah salah satu spesies udang yang unggul yang sejak tahun 2002 mulai dikultur di tambak-tambak di Indonesia. Udang ini berasal dari perairan Amerika dan Hawaii yang sukses dikembangkan dibeberapa negara Asia termasuk Indonesia, Vannamei banyak diminati, karena memiliki banyak keunggulan antara lain, relatif tahan penyakit, pertumbuhan cepat (masa pemeliharaan 100 - 110 hari), padat tebar tinggi, sintasan pemeliharaan tinggi dan *Feed Convertion Ratio* rendah (Hendarajat et al., 2007). Tingkat kelulushidupan vannamei dapat mencapai 80 - 100% (Duraiappah et al., 2000), dan menurut (Boyd & Clay, 2002), tingkat kelulushidupannya mencapai 91%. Berat udang ini dapat bertambah lebih dari 3 gram tiap minggu dalam kultur dengan densitas tinggi (100 udang/m2). Ukuran tubuh maksimum mencapai 23 cm. Berat udang dewasa dapat mencapai 20 gram dan diatas berat tersebut, L.vannamei tumbuh dengan lambat yaitu 7 sekitar 1gram/ minggu. Udang betina tumbuh lebih cepat daripada udang jantan.

2) Udang Windu

Udang windu *(Penaeus Monodon)* atau sering pula disebut sebagai *black tiger shrimp* adalah spesies udang laut yang dapat mencapai ukuran besar, di alam bebas ia bisa mencapai ukuran 35 cm dan berat 260 g, sedangkan di tambak tubuhnya hanya bisa mencapai 20 cm dan berat 140 g. Udang Windu ini merupakan jenis udang yang memiliki nilai ekonomis cukup tinggi. Saat ini Udang windu perlahan bangkit dan mulai berkembang sangat baik di berbagai daerah di Indonesia meskipun beberapa tahun yang lalu budidaya udang ini sempat ambruk karena diserang hama penyakit. Budidaya udang windu terdapat hampir di semua wilayah Indonesia. Sentra budidaya udang windu sendiri terletak di provinsi Sumatera selatan, jawa barat dan sulawesi selatan.



Gambar 3. siklus hidup udang (Admin, 2019)

Gambar 3. siklus hidup udang (Admin, 2019), menunjukkan siklus hidup udang. Udang windu membutuhkan waktu 1 tahun untuk mencapai dewasa. Udang-udang dewasa ini memijah pada malam hari di perairan yang dalam di dasar laut. Seekor udang windu betina bisa menghasilkan telur sampai 150.000 butir. Telur-telur yang dibuahi akan mengalami masa inkubasi selama kurang lebih 12-16 jam, dan akan menetas menjadi nauplius pada

suhu 27-29ºC. Sebagai larva *nauplius* berganti kulit sampai 6 kali dan menjadi *nauplius* substadium VI dalam waktu kurang lebih 2 hari, kemudian menjadi zoea yang bentuk badannya memanjang dan mulai tampak matanya. Setelah 4-6 hari dan berganti kulit sebanyak 3 kali *zoea* ini telah berada pada tahapan *zoea* substadium III, lalu berubah menjadi *mysis*. Masa *mysis* ini dijalani selama 4-5 hari dan selama itu pula udang berganti kulit sebanyak 3 kali dan menjadi mysis stadium III. Sesudah itu berubah menjadi post larva dan beberapa kali pula berganti kulit. Pada fase ini anak udang memasuki muara sungai di tepi pantai. Setelah substadium terakhir dari post larva, anak udang sudah menjadi sempurna sehingga sudah dapat digolongkan sebagai *juvenile* (udang muda). Selama stadium ini hingga dewasa, udang masih beberapa kali berganti kulit. Sesudah tumbuh menjadi udang dewasa yang siap kawin, udang akan bermigrasi ke tengah laut lagi untuk melakukan pemijahan.

e. Benih Udang

Benih udang popular disebut benur singkatan dari benih urang (dalam bahasa Jawa disebut urang). Dalam usaha budidaya tambak udang, benur merupakan salah satu sarana kunci produksi. Bahkan, benur dinyatakan sebagai faktor pembatas. Maksudnya, jika benur sedikit maka hasil produksi akan sedikit. Sebaliknya, jika benur banyak maka hasil produksi pun akan banyak. Benih udang atau benur (benih urang) dapat berasal dari hasil tangkapan di alam atau dari hasil pembenihan di balai benih.

Benih udang vanname, karena merupakan udang introduksi, sepenuhnya berasal dari pembenihan di balai benih atau hatchri *(hatchery)*. Bahkan untuk memproduksi benih udang vanname, sebagian besar masih diimpor dari Amerika. Adapun benih udang windu juga udang putih dapat berasal dari hasil penangkapan maupun hasil pembenihan.

Benih udang yang siap ditebar haruslah benih berkualitas. Benih yang berkualitas tumbuh pesat, sehat, dan setiap hari berganti kulit. Pada udang windu, larva yang baru menetas

akan menjadi post larva (pl) pada umur 9-10 hari yang dikenal sebagai benur. Berdasarkan penelitian dan pengalaman benih yang baik untuk ditebar di tambak adalah benih yang telah mencapai stadia pl 30, karena benih ini telah cukup kuat dan tahan terhadap kondisi tambak. Namun umumnya pengusaha benih menjual benih pada pl 12 atau pl 15. Karena itu, petambak melakukan pendederan selama 30 hari, baru dilanjutkan kegiatan pembesaran. Ada juga sebagian petambak membeli benih pl 12 atau pl 15 kemudian melakukan kegiatan pendederan, dan benih hasil dari pendederan pl 45 dijual kepada petambak yang melakukan kegiatan pembesaran. Kegiatan pemeliharaan benih ini biasa disebut pendederan atau pengipukan.

2. *Image Processing*

Image processing atau pengolahan citra merupakan teknik dalam pemrosesan gambar dengan input berupa citra dua dimensi yang bertujuan untuk menyempurnakan citra atau mendapatkan informasi yang berguna untuk diolah menjadi beberapa keputusan. Dalam operasi pemrosesan citra, operasi yang sering dilakukan dalam gambar *grayscale.* Gambar *grayscale* didapatkan dari pemrosesan gambar berwarna yang didekomposisi menjadi komponen merah (R), hijau (G), dan biru (B) yang diproses secara independen sebagai gambar *grayscale. Image processing* terbagi menjadi dalam 3 tingkatan (Tyagi, 2018):

1) Low-level Image Processing

Merupakan operasi sederhana dalam pengolahan gambar dimana input dan output berupa gambar. Contoh: *contrast enchancement* dan *noise reduction*

2) Mid-level Image Processing

Merupakan operasi pengolahan gambar yang melibatkan ekstrasi atribut dari gambar input. Contoh: *edges, contours* dan *regions.*

3) High-Level Image Processing

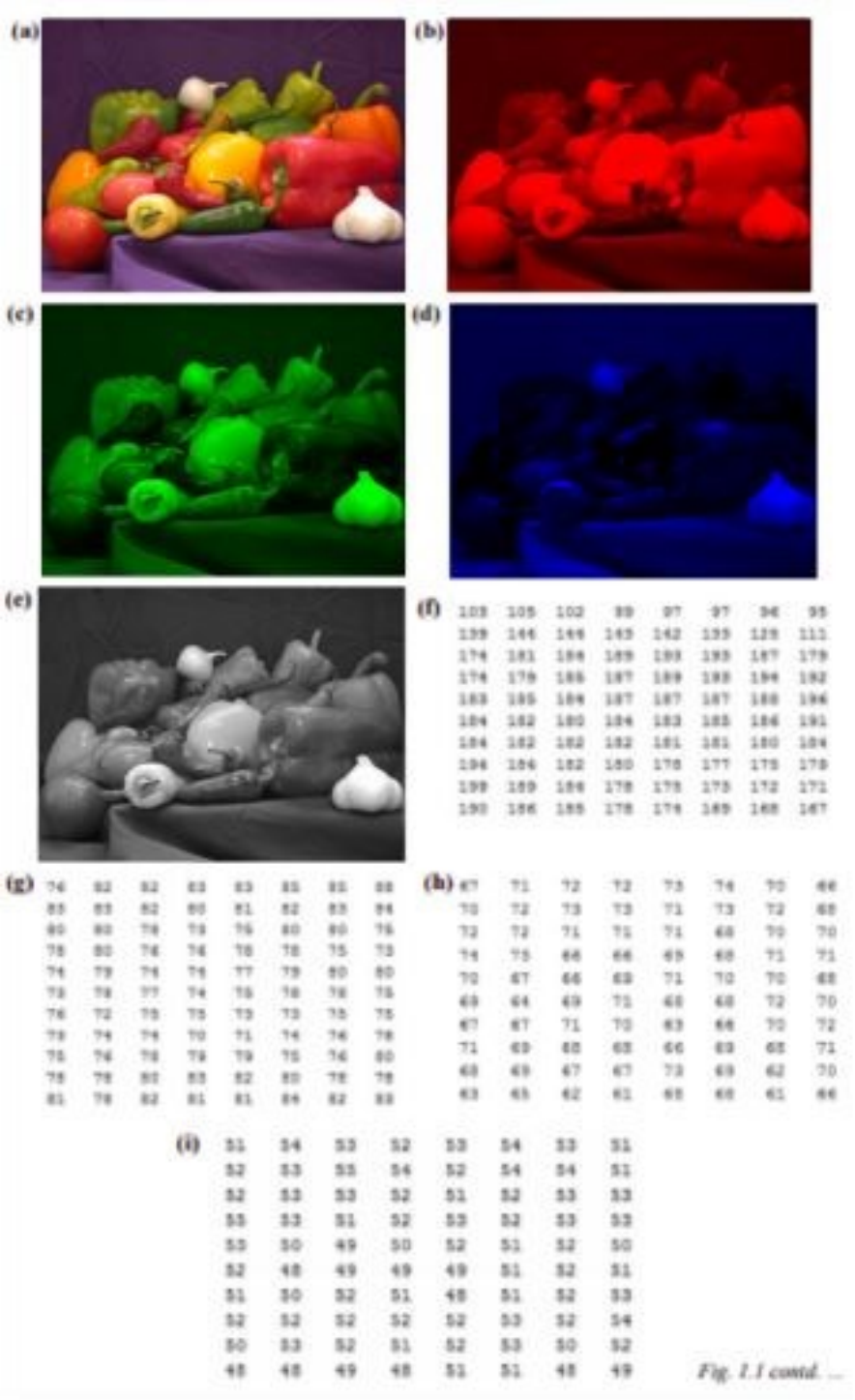
Merupakan kategori yang melibatkan pemrosesan gambar kompleks yang terkait dengan analisis dan interprestasi konten dalam sebuah keadaan untuk pengambilan keputusan.

Digital image merupakan fungsi dua dimensi f(x,y) yang merupakan proyeksi dari bentuk tiga dimensi kedalam bentuk dua dimensi dimana x dan y merupakan lokasi elemen gambar atau piksel yang berisikan nilai. Ketika nilai x, y dan intensitasnya berupa diskrit, maka gambar tersebut dapat dikategorikan sebagai digital *image.* Secara matematis, digital *image* adalah representasi matriks dari gambar dua dimensi menggunakan piksel. Setiap piksel diwakili oleh nilai numerik. Untuk gambar *grayscale*, hanya memiliki satu nilai dengan kisaran antara 0-255. Untuk gambar yang berwarna, memiliki tiga nilai yang mewakili merah (R), hijau (G), dan biru (B) yang masing-masing memiliki kisaran nilai yang sama antara 0-255 pada Gambar 4. Jika suatu gambar hanya memiliki dua intensitas, gambar tersebut dikenal sebagai binary image (Tyagi, 2018).

3. Visi Komputer

Visi Komputer adalah cabang *Artificial Intelligent* (AI) yang mencakup proses analisa citra dan video. Visi komputer mengimplementasikan beberapa kemampuan visual manusia yang diteruskan menuju otak seperti deteksi benda, pengenalan wajah dan mengenali bahaya.

Pada visi komputer, *Deep Learning* sering digunakan untuk pengenalan dan deteksi objek. Proses *Deep Learning* pada visi komputer memanfaatkan piksel pada citra untuk ekstrasi pola atau atribut dari citra yang ingin dideteksi. Biasanya, untuk setiap aplikasi yang diberikan, keseluruhan tugas tidak dapat dilaksanakan pada sebuah tahapan tunggal. *Computer Vision* terdiri dari tahapan-tahapan seperti perolehan citra, *preprocessing,* pengekstraksian fitur, penyimpanan objek secara asosiatif, pengaksesan suatu basis pengetahuan, dan pengenalan. (Robert B.Fisher, 2013)



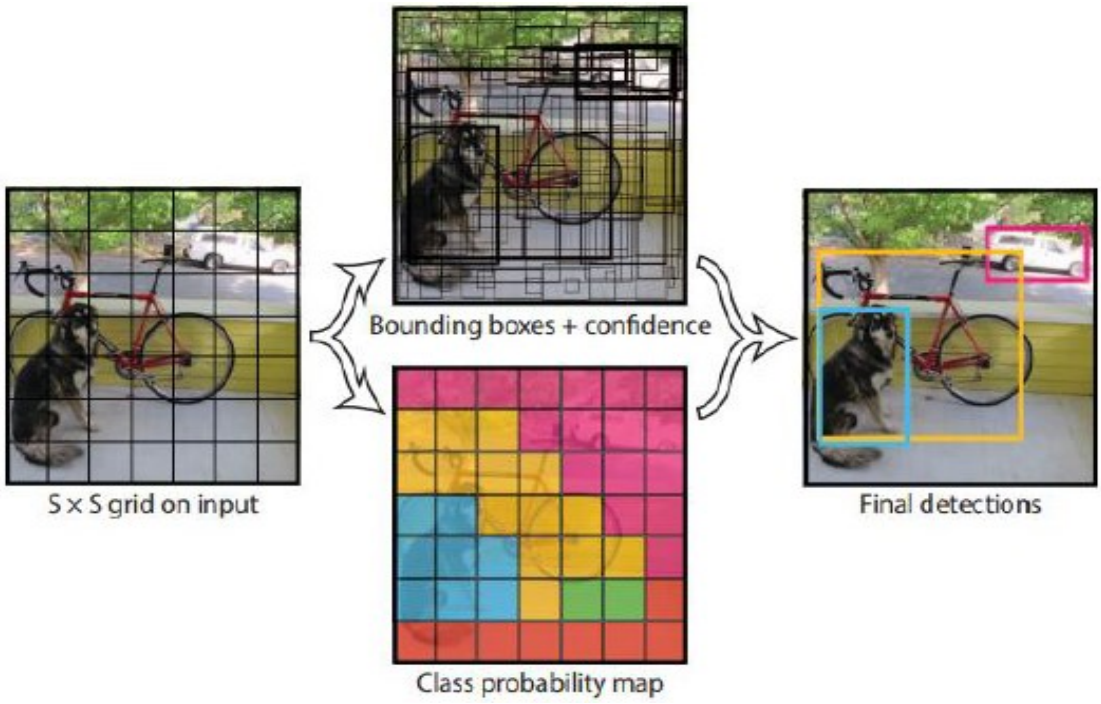
Gambar 4. Digital Image; (a) Gambar berwarna, (b) Komponen merah dalam gambar berwarna, (c) Komponen hijau dalam gambar bewarna, (d) Komponen biru dalam gambar bewarna, (e) Gambar bewarna dikonversi dalam 8bit grayscale, (f) Matriks dari gambar (b), (g) Matriks dari gambar (c), (h) Matriks dari gambar (d), (i) Matriks dari gambar (e).

4. YOLO

YOLO (You Only Look Once) merupakan salah satu algoritma *one-stage object detection*. YOLO memiliki kecepatan komputasi yang sangat tinggi dan dapat memproses gambar secara real time. Pada YOLO, komponen-komponen object detection yang terpisah disatukan dalam satu jaringan saraf tiruan. Sehingga YOLO memungkinkan pelatihan secara *end-to-end* dan memperoleh kecepatan yang tinggi dengan tetap mempertahankan presisi yang cukup tinggi (Redmon et al., 2016).

YOLO membagi input gambar menjadi grid berukuran S×S, dimana nilai S adalah 7 dengan input gambar berukuran 448 x 448. Untuk memperoleh *bounding box*, dilakukan konvolusi dari *input* gambar, sehingga hasil akhirnya akan mendapat ukuran *bounding box* sebesar S×S×(B*5+C) dimana B adalah banyaknya *bounding box* (umumnya 2) dalam 1 grid dan C adalah banyaknya class yang dapat diklasifikasi. Nilai B dikalikan dengan 5 karena sebuah *bounding box* memiliki 5 nilai yang perlu disimpan, koordinat x, koordinat y, lebar (width), tinggi (height), dan *confidence score* (nilai probabilitas bounding box pada sebuah objek).

Untuk semua atribut pada bounding box akan dilakukan normalisasi sehingga nilainya menjadi antara 0 hingga 1. Koordinat x dan y akan dinormalisasi menyesuaikan titik kiri atas dari grid yang bersangkutan. Tinggi dan lebar akan dinormalisasi sesuai dengan ukuran gambar. Nilai koordinat x dan y pada sebuah bounding box pada setiap grid merupakan titik tengah grid yang bersangkutan. Secara umum model YOLO dapat dilihat pada Gambar 5.



Gambar 5. YOLO (Redmon et al., 2016)

1. YOLOv2

YOLO menghasilkan *localization error* yang relatif tinggi dan *recall* yang relatif rendah dibandingkan dengan metode *object detection* berbasis region proposal. Selain itu, jumlah objek yang dapat dideteksi oleh YOLO terbatas karena setiap sel grid hanya dapat mendeteksi satu kelas objek. Sehingga YOLO tidak dapat mendeteksi lebih dari satu objek jika terdapat beberapa objek yang berdekatan dalam satu sel grid. Oleh karena itu, (Redmon & Farhadi, 2017) mencoba mengatasi permasalahan tersebut dengan menerapkan beberapa perbaikan pada YOLO dan menamai metode yang diperbaiki ini dengan YOLOv2. Adapun beberapa perubahan pada YOLOv2 di antaranya adalah sebagai berikut (Redmon & Farhadi, 2017).

a. Batch Normalization

Batch normalization dilakukan dengan melakukan normalisasi terhadap nilai bobot layer input pada *artificial neural networks* (ANN) Teknik ini digunakan untuk meningkatkan kecepatan, kinerja, dan kestabilan ANN. Dengan menambahkan *batch normalization* pada YOLO, YOLOv2 memperoleh peningkatan mAP lebih dari 2%.

b. Classifier Resolusi Tinggi

YOLO melatih model classifier dengan resolusi 224 × 224 kemudian meningkatkan resolusinya menjadi 448 untuk deteksi. Adapun untuk YOLOv2, sebelum tahap pelatihan model untuk *object detection*, terlebih dahulu dilakukan fine tune pada model *classifier* dengan resolusi 448 × 448 sebanyak 10 epoch dengan dataset *ImageNet*. Dengan jaringan *classifier* resolusi tinggi ini, YOLOv2 memperoleh peningkatan mAP mendekati 4%.

c. Anchor/Prior Box dan Dimension Clusters

Telah disebutkan sebelumnya bahwa pada YOLO jika titik tengah suatu objek jatuh pada suatu sel grid maka sel grid tersebut bertanggung jawab untuk memprediksi objek tersebut. Akan tetapi, setiap sel grid hanya dapat memprediksi satu objek. Hal ini menjadi kelemahan YOLO ketika terdapat lebih dari satu objek yang titik tengahnya jatuh pada sel grid yang sama. Oleh karena itu, untuk mengatasi masalah ini YOLOv2 mencoba untuk memungkinkan setiap sel grid memprediksi lebih dari satu objek dengan menggunakan k bounding box.

d. Arsitektur Jaringan

YOLOv2 menggunakan arsitektur jaringan Darknet-19 sebagai *backbone*. Darknet-19 merupakan model klasifikasi yang dikembangkan untuk basis YOLOv2. Darknet-19 hanya membutuhkan 5,58 miliar operasi untuk memproses suatu gambar namun memperoleh hasil yang baik. Pada dataset ImageNet, Darknet-19 menghasilkan nilai akurasi top-5 91,2% yang mana lebih baik dari VGG-16 yang memperoleh 90% dan model custom YOLO yang memperoleh 88%. Gambar 6 merupakan arsitektur jaringan Darknet-19.

Type	*Filters*	*Size/Stride*	*Output*
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2 / 2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2 / 2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2 / 2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2 / 2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2 / 2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Averagepool		Global	1000
Softmax			

Gambar 6. Darknet-19 (Redmon & Farhadi, 2017)

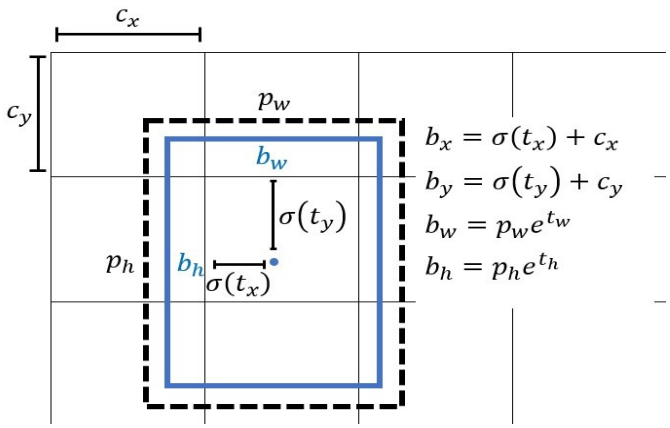
2. YOLOv3

YOLOv3 merupakan generasi ketiga dari YOLO yang dipublikasikan pada 2018 oleh Redmon dan Farhadi. Terdapat beberapa perubahan pada YOLOv3 dari generasi sebelumnya (YOLOv2) di antaranya adalah sebagai berikut (Redmon & Farhadi, 2018).

a. Prediksi Bounding Box

Sebagaimana YOLOv2, YOLOv3 menggunakan dimension clusters sebagai prior box. Yang berubah pada YOLOv3 adalah perhitungan fungsi cost. Setiap *bounding box* (Gambar 7) memiliki *objectness score* yang diprediksi menggunakan *logistic regression*. Jika *bounding box prior* (anchor) tumpang tindih (*overlap*) dengan *ground truth object* lebih dari *bounding box prior* yang lainnya (memiliki nilai IOU paling besar), maka *objectness scorenya* adalah 1. Untuk prior lainnya jika memiliki nilai IOU lebih besar dari ambang batas tertentu (misalnya

0,5) maka diabaikan. Sistem hanya menetapkan satu *bounding box prior* untuk setiap *ground truth object*. Jika suatu *bounding box prior* tidak ditetapkan pada ground truth object manapun, maka tidak ada *loss* untuk prediksi koordinat dan kelas, hanya *objectness*.



Gambar 7. Bounding box (Redmon & Farhadi, 2018)

b. Prediksi Kelas

Generasi YOLO sebelumnya menggunakan fungsi softmax untuk klasifikasi sehingga setiap objek hanya ditetapkan pada satu kelas. Pendekatan ini berjalan jika diasumsikan bahwa output objek bersifat mutually exclusive. Akan tetapi, jika suatu objek memiliki lebih dari satu kelas (multi-label), maka diperlukan pendekatan klasifikasi multi-label untuk menghasilkan model yang lebih baik. Misalnya, suatu objek dapat memiliki dua label yaitu “person” dan “woman”. Untuk melakukan klasifikasi multi-label, YOLOv3 mengganti fungsi softmax dengan independent logistic classifier. Adapun loss function untuk prediksi kelas yang digunakan selama pelatihan adalah binary cross-entropy.

c. Prediksi Multiskala

YOLO dan YOLOv2 memprediksi output pada layer terakhir, sedangkan YOLOv3 memprediksi bounding box pada beberapa skala yang berbeda. Pada percobaan yang dilakukan oleh (Redmon & Farhadi, 2018) menggunakan dataset COCO (Common Objects in Context), YOLOv3 memprediksi bounding box pada 3 skala yang berbeda. Setelah melakukan prediksi bounding box pada skala pertama, YOLOv3 mengambil feature map dari 2 layer terakhir dan

melakukan upsample 2 kali. YOLOv3 juga mengambil *feature map* dari layer sebelumnya kemudian menggabungkannya dengan feature map yang sudah dilakukan *upsample*. Selanjutnya deteksi kedua dilakukan pada gabungan feature map tersebut dengan skala kedua. Untuk deteksi terakhir dengan skala ketiga, dilakukan proses yang sama dengan desain skala kedua. Untuk menentukan *prior box*, (Redmon & Farhadi, 2018) menggunakan k-means dan dipilih sebanyak 9 cluster. Karena pada pelatihan YOLOv3 digunakan 3 skala, maka masing-masing skala menggunakan 3 prior box. Pada dataset COCO, 9 cluster tersebut adalah: (10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), (373×326).

d. *Feature Extractor*

Untuk ekstraksi fitur YOLOv3 menggunakan pendekatan hybrid antara jaringan Darknet-19 dan jaringan residual (Gambar 8). Jaringan ini memiliki 53 layer dan dinamai Darknet-53. Jaringan ini lebih powerful daripada Darknet-19 tetapi masih lebih efisien dari ResNet-101 atau ResNet-152 (Redmon & Farhadi, 2018).

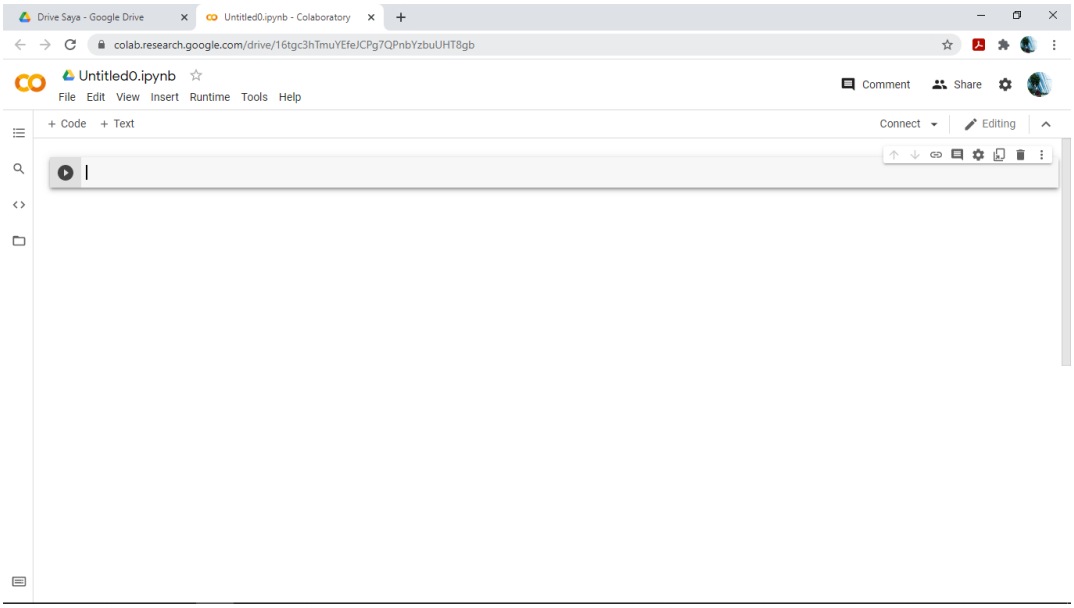
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3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

Gambar 8. Model layer network darknet 53 (Redmon & Farhadi, 2018)

5. Google Colaboratory

Google Colab adalah salah satu produk Google berbasis *cloud* yang bisa digunakan secara gratis. Google Colab dibuat khusus untuk membantu para peneliti dalam mengolah data, khususnya di bidang *Machine Learning*. *Tools* Colab ini menyediakan layanan GPU gratis kepada pengguna sebagai backend komputasi. Dengan Google Colab pengguna dapat mengakses sumber daya yang *powerful* melalui browser secara bebas selama 12 jam untuk mengeksekusi kode dan membangun aplikasi menggunakan berbagai *library* yang tersedia seperti *Numpy, OpenCV, Keras, TensorFlow,* dan lain-lain.

Google Colab adalah *coding environment* bahasa pemrograman *Python* dengan format *"notebook"* memiliki tampilan yang mirip dengan *Jupyter Notebook* (Gambar 9) dan tidak memerlukan pengaturan terlebih dahulu sebelum digunakan dan berjalan sepenuhnya pada *cloud* dengan memanfaatkan media penyimpanan Google Drive.



Gambar 9. Tampilan Google Colaboratory

Selain itu, terdapat beberapa perintah yang bisa digunakan di Google Colab seperti perintah *git, pip* dan sed. Git merupakan fasilitas version control system yang banyak digunakan oleh programmer untuk berbagi paket program dan membangun aplikasi secara

bersama-sama dengan menggunakan perintah dasar seperti git init (membuat repositori/tempat penyimpanan paket program), git add (menambah file baru pada repositori), git clone (menyalin repositori), dll (*Git*, n.d.).

Pip adalah salah satu pengelola paket *python*. Perintah pip merupakan baris program yang dapat dieksekusi di command prompt/penerjermah baris perintah yang ada di Windows ataupun python interpreter. Adapun sed merupakan stream editor dari Unix yang digunakan untuk manipulasi script tanpa harus membuka file sumbernya.

6. *Image Annotation*

Image Annotation adalah teknik utama yang digunakan untuk membuat data pelatihan untuk computer vision. Image Annotation pada dasarnya adalah proses pelabelan data dalam berbagai media gambar, teks, atau video (Low, 2020). Pada tesis ini pemberian anotasi dilakukan pada objek larva udang. Terdapat beberapa macam anotasi gambar yang biasa dipakai dalam visi komputer, di antaranya yaitu: *bounding boxes* (Gambar 10) , 3D (Gambar 11), *polygon* (Gambar 12), *semantic segmentation* (Gambar 13), *key-point* (Gambar 14), Lines (Gambar 15), dan *image classification* (Gambar 16).

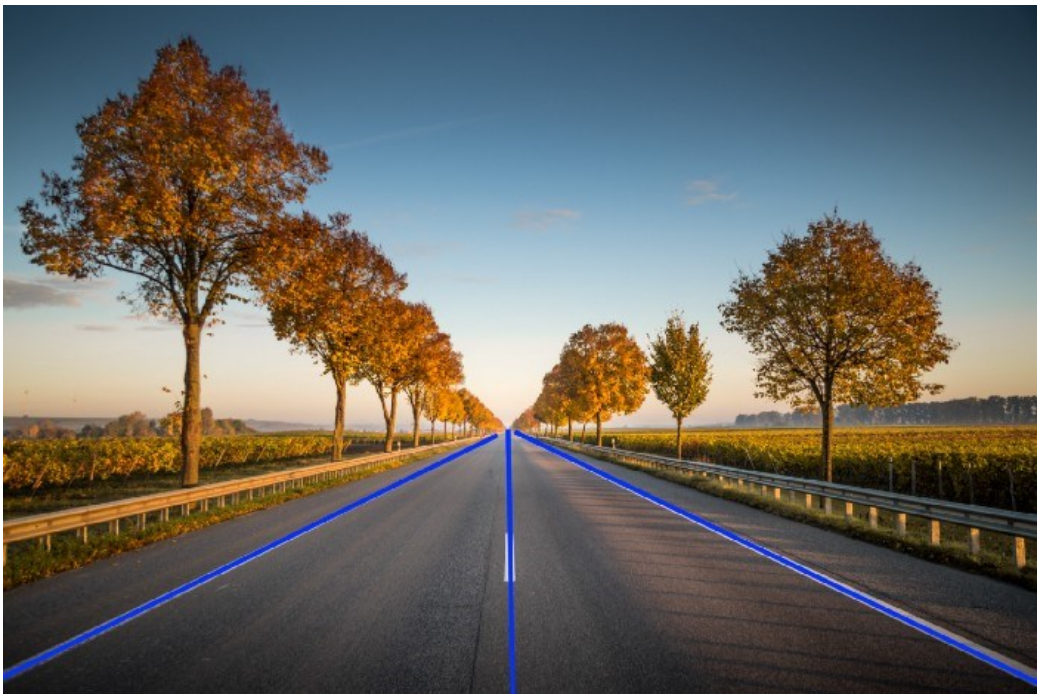
Gambar 10. Bounding Box (Pokhrel, 2020)

Gambar 11. 3D cuboids (Pokhrel, 2020)

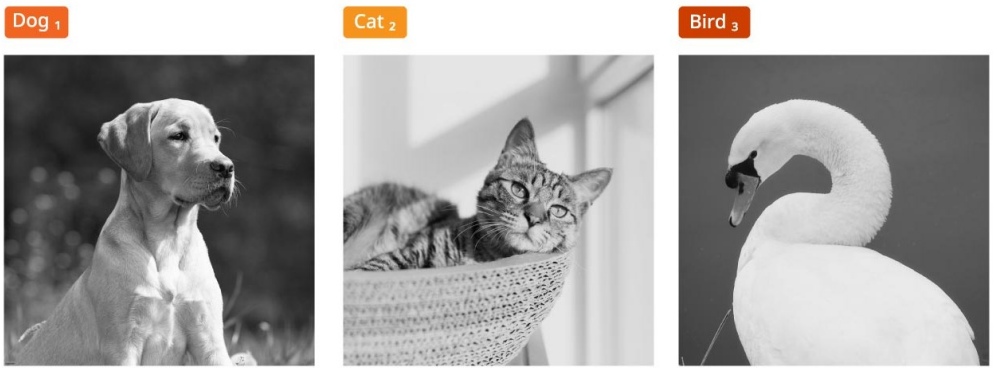
Gambar 12. Polygonal segmentation (Ambalina, 2019)

Gambar 13. Semantic segmentation (Pokhrel, 2020)

Gambar 14. Key-point annotation (Pokhrel, 2020)



Gambar 15. Line (Pokhrel, 2020)



Gambar 16. Image Classification (Ambalina, 2019)

B. PENELITIAN TERKAIT

Beberapa penelitian terkait pendeteksian dan penghitungan larva udang di antaranya sebagai berikut:

a. (Kaewchote et al., 2018) melakukan penghitungan post larva udang menggunakan teknik pengenalan citra. Penghitungan dilakukan dengan menggunakan metode *Local Binary Pattern* (LBP) dan klasifikasi *Random Forest.* Menilai hasil dengan validasi K-fold cross (k ¼ 5) menunjukkan bahwa metode LBP akurat 98,50%. Pada penelitian ini didapatkan *root mean square error* (RMSE) yang relatif besar sebesar 14,43 karena tumpang tindih udang.

b. (Awalludin et al., 2019), menggunakan analisis blob dan teknik component connected untuk menghitung larva udang. Berdasarkan hasilnya, metode yang diusulkan menyajikan *Root Mean Square Error* (RMSE) kurang dari 6% yang lebih baik dari beberapa pendekatan penghitungan manual.

c. (Nguyen et al., 2020) menerapkan segmentasi instance berbasis Mask R-CNN dua fase meningkat dengan margin maksimum 16,1% untuk mensegmentasi larva udang untuk tujuan perhitungan. Pendekatan ini memiliki hasil dengan akurasi mulai dari 92,2% hingga 95,4% untuk gambar tumpang tindih sedang.

d. (Solahudin et al., 2018) menggunakan teknik pengolahan citra dalam menghitung benih udang. Proses pengolahan citra yang digunakan terdiri dari pemotongan citra, thresholding, dilasi, dan pelabelan. Gambar diambil menggunakan kamera CCD dengan sistem *backlight.* Hasil akurasi dari penelitian ini sebesar 98.49%. rata-rata waktu penghitungan benih udang adalah 1.70 detik/citra dengan standar deviasi 0.15 detik.

e. (Khantuwan & Khiripet, 2012) mengusulkan penggunaan *Co-Occurrence Color Histogram* untuk menghitung larva udang secara akurat dan otomatis. Penghitungan jumlah larva udang dilakukan dalam wadah berisi air. Gambar wadah diambil dari kamera

di atas wadah. Gambar-gambar diolah terlebih dahulu untuk menghilangkan dan meningkatkan *noise*. Gambar yang dihasilkan kemudian diproses untuk menghitung jumlah larva udang dengan akurasi yang dapat diterima. Hasil percobaan menunjukkan akurasi sebesar 97% dalam kondisi terkendali.

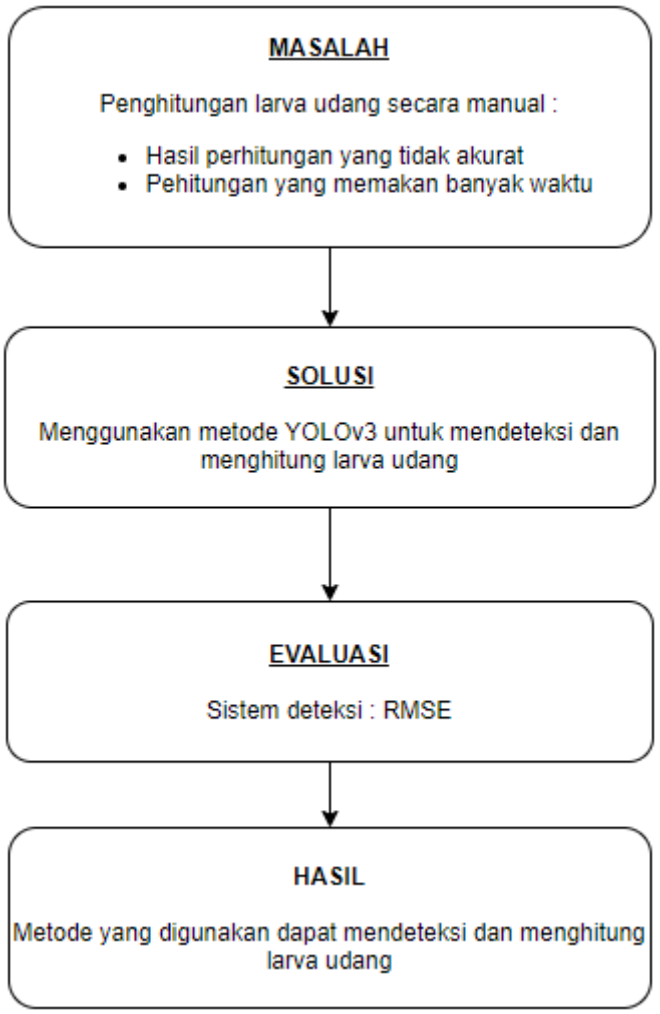
C. STATE OF THE ART

Penelitian sebelumnya oleh Kaewchote et al, melakukan penghitungan post larva udang menggunakan teknik klasifikasi. Awalludin et al, menggunakan analisis blob dan teknik *component connected* untuk menghitung larva udang. Nguyen et al, menerapkan segmentasi instance berbasis *Mask R-CNN* dua fase meningkat dengan margin maksimum 16,1% untuk mensegmentasi larva udang. Solahudin et al, melakukan deteksi larva udang dengan menggunakan teknik pengolahan citra. Khantuwan & Khiripet, mengusulkan penggunaan *Co-Occurrence Color Histogram* untuk menghitung larva udang secara akurat dan otomatis.

Tabel 1. *State of the art*

No	Judul	Penerbit, Penulis & Tahun	Metode	Hasil	Jumlah Data
1.	*Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp* (10.1016/j.anres.2018.10.007)	*Agriculture and Natural Resources,* Kaewchote et al, 2018	• LBP • RF classifier	Sistem menunjukkan persentase yang tinggi dalam menghitung post larva udang	100 gambar
2.	*Combination of Canny Edge Detection and Blob Processing Techniques for Shrimp Larvae Counting* (10.1109/ICSIPA45851.2019.8977746)	*Proceedings of the* 2019 IEEE *International Conference on Signal and Image Processing Applications*, ICSIPA 2019, Awalludin et al, 2019	• *Canny Edge Detection* • *Blob* analysis	metode yang diusulkan menyajikan Root Mean Square Error (RMSE) kurang dari 6% yang lebih baik dari beberapa pendekatan penghitungan manual	90 gambar
3.	*Two-Phase Instance Segmentation for Whiteleg Shrimp Larvae Counting* (10.1109/ICCE46568.2020.9043075)	*Digest of Technical Papers* - IEEE *International Conference on Consumer Electronics,* Nguyen et al, 2020	• Mask R-CNN	memiliki hasil dengan akurasi mulai dari 92,2% hingga 95,4% untuk gambar udang tumpang tindih sedang	-
4.	*Vaname (Litopenaeus vannamei) Shrimp Fry Counting Based on Image Processing Method* (10.1088/1755-1315/147/1/012014)	IOP *Conference Series: Earth and Environmental Science,* Solahudin et al, 2018	• Operasi *Thresholding*	Akurasi 98.49% dalam menghitung benih udang dalam waktu rata-rata 1.70 detik.	-
5.	*Live Shrimp Larvae Counting Method Using Co-occurrence Color Histogram*	2012 *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology,* Khantuwan & Khiripet, 2012	• *Co-Occurrence color Histogram*	Hasil percobaan menunjukkan akurasi sebesar 97%	100 gambar

D. KERANGKA PIKIR



Gambar 17. Kerangka Pikir

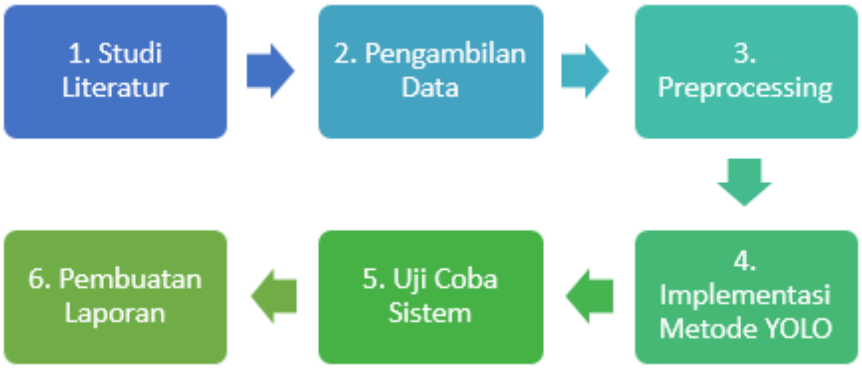
Pada Gambar 17 menunjukkan kerangka pikir pada penelitian ini. Pada tahap pertama menjelaskan permasalahan dalam menghitung larva udang secara manual, yaitu hasil perhitungan yang tidak akurat serta memakan banyak waktu. Kemudian, solusi yang ditawarkan ialah dengan menggunakan sebuah sistem berbasis deteksi objek yang dapat mendeteksi serta menghitung objek secara otomatis sehingga digunakan metode *deep learning* yaitu YOLOv3. Evaluasi dilakukan dengan menghitung nilai *root mean square error* (RMSE).

BAB III

METODOLOGI PENELITIAN

A. TAHAPAN PENELITIAN

Sistem yang dibuat pada penelitian ini menggunakan algoritma You Only Look Once (YOLO) generasi ketiga, yang terdiri dari *convolutional neural network layer* untuk proses ekstraksi fitur dari input serta proses *localization* objek, dan *fully connected layer* untuk mengklasifikasikan jenis larva udang. Adapun tahapan-tahapan yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 18.



Gambar 18. Diagram tahapan penelitian

Tahapan penelitian pada Gambar 18 dijelaskan sebagai berikut.

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahapan ini dilakukan untuk mengumpulkan penelitian terkait penghitungan larva udang dan penggunaan algoritma yolo dalam melakukan pendeteksian terhadap objek disekitar maupun objek pada benur.
2. Pengambilan data dilakukan dengan menggunakan kamera handphone yang memiliki kualitas kamera minimum 5 pixel.
3. *Preprocessing* data berupa proses analisis data yang layak digunakan, melakukan penyortiran dan *resize* pada data, melakukan labelling terhadap data training, untuk memperbaiki akurasi sistem yang dibuat.

4. Dalam penelitian ini, metode yang digunakan untuk melakukan pendeteksian serta penghitungan objek adalah YOLO generasi ketiga dengan menggunakan Bahasa pemrograman *python* dan *framework Darknet* untuk proses *training*. Pada tahap ini juga dilakukan pembuatan *flowchart* terkait alur kerja sistem.
5. Uji coba sistem dilakukan untuk mengetahui seberapa akurat sistem yang dibuat.
6. Pembuatan laporan. Tahapan akhir yang dilakukan adalah melakukan penulisan laporan secara menyeluruh sebagai bahan publikasi dan laporan akhir magister.

B. WAKTU DAN LOKASI PENELITIAN

Penelitian ini dimulai pada saat persetujuan proposal diterima. Penelitian dilakukan di dua tempat, yaitu balai perikanan budidaya air payau dan laboratorium. Proses pengambilan data berlokasi di Kabupaten Takalar, Sulawesi Selatan, sedangkan proses pengolahan data dan perancangan sistem dikerjakan di Kampus Teknik Gowa Universitas Hasanuddin.

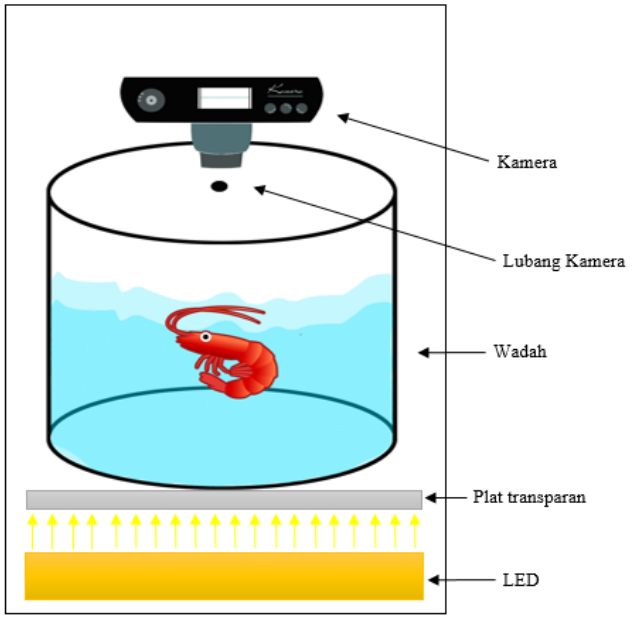
C. INSTRUMEN PENELITIAN

Instrumen penelitian yang digunakan pada penelitian ini meliputi:

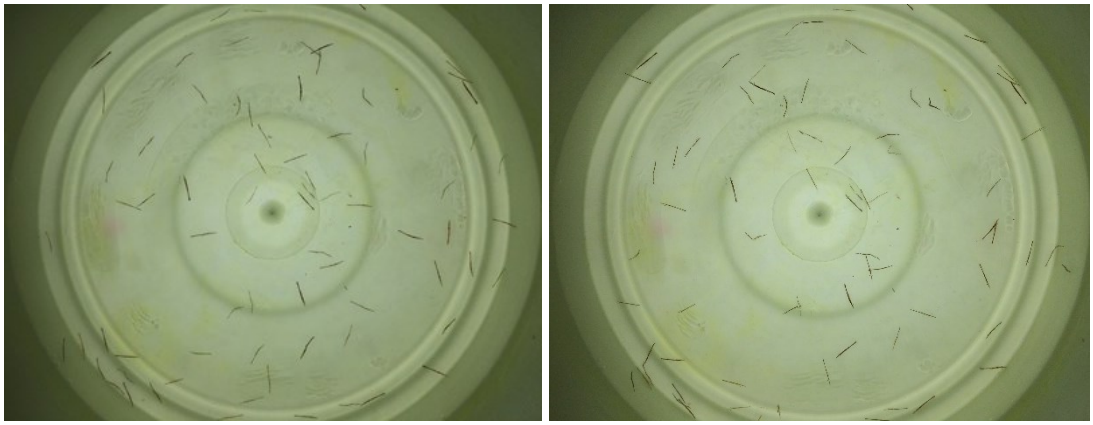
1. *Software*
   a. Sistem Operasi Windows 10, 64 bit
   b. Visual Studio Code
   c. Google Colab
   d. Anaconda prompt
   e. OpenCV 4.4.0
2. *Hardware*
   a. Laptop ASUS, Prosesor Intel® Core™ i5-4210U, up to 2.39 GHz , HDD 1 TB NVIDIA GEFORCE, RAM 8GB.
   b. *Smartphone Redmi Note* 8 dan *iPhone* 5S.

D. TEKNIK PENGAMBILAN DATA

Data yang digunakan pada penelitian ini berupa data citra dan video dengan larva udang stadia PL 8, PL 10, dan PL 12. Jenis udang yang digunakan pada penelitian ini adalah jenis udang windu. Pengambilan data dilakukan dengan mengambil citra udang dalam variasi jumlah dan bentuk. Pada Gambar 19 citra diambil menggunakan kamera *Smartphone* yang tersimpan diatas wadah putih berisi air 2 cm serta benih udang dengan menggunakan sistem *backlight* agar tidak ada pantulan cahaya dari dalam air karena cahaya diberikan dari bawah. Gambar 20 merupakan contoh larva udang yang dijadikan data latih.



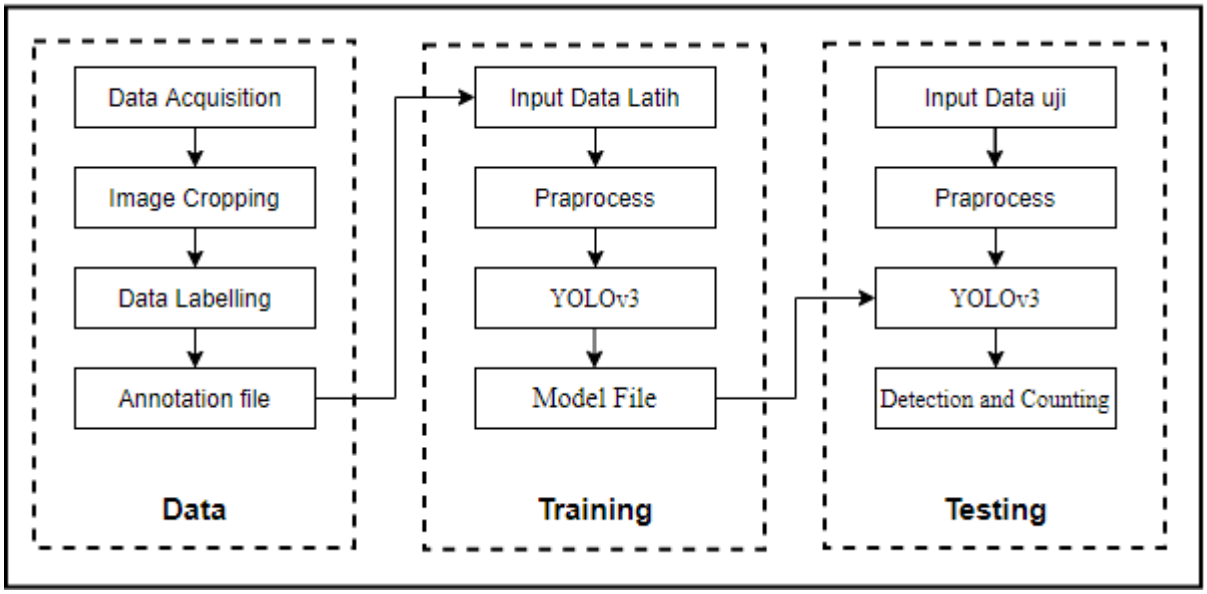
Gambar 19. Pengambilan data citra



Gambar 20. Data latih

E. PERANCANGAN SISTEM

Sistem ini secara umum dibagi menjadi tiga bagian, yaitu akuisisi data, pelatihan dan pengujian seperti yang diperlihatkan pada Gambar 21.

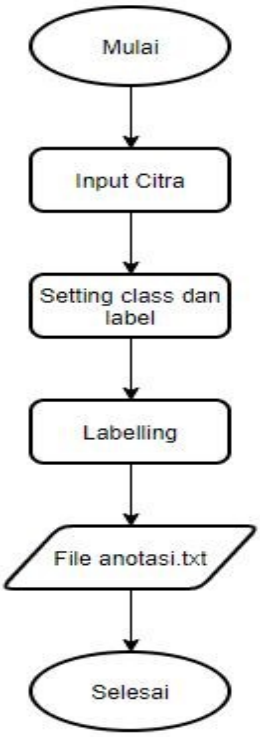


Gambar 21. Desain alur sistem

Berikut ini penjelasan mengenai proses yang terjadi pada masing-masing blok sistem deteksi larva udang pada Gambar 21.

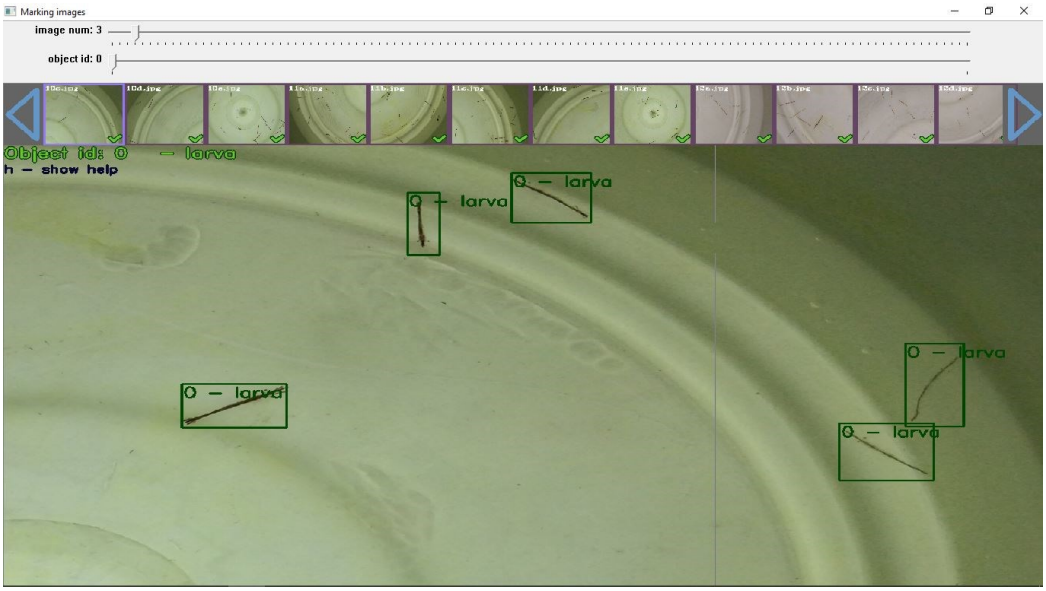
1. Pelabelan Data

Pada penelitian ini, data latih yang digunakan sebanyak 325 data citra sedangkan data uji sebanyak 99 data. Data latih yang digunakan pada penelitian ini terlebih dahulu akan di*cropping* menjadi empat bagian pada satu citra. Hal ini bertujuan untuk meningkatkan performa YOLO dalam mengenal objek larva udang. Sedangkan data uji tetap dengan ukuran aslinya yaitu 3264 x 244 piksel. Pada metode *deep learning* berbasis YOLO, data yang akan digunakan untuk proses *training* perlu diberi label terlebih dahulu. Proses pelabelan dilakukan menggunakan yolo_mark (AlexeyAB, 2019).



Gambar 22. Flowchart proses labelling yolo_mark

Pada Gambar 22 menampilkan tahapan proses pelabelan menggunakan yolo_mark. Pertama, masukkan citra pada direktori x64/Release/data/img, selanjutnya dilakukan pengaturan jumlah kelas dan label yang akan digunakan. Untuk penelitian ini hanya menggunakan satu label yaitu ‘larva’.



Gambar 23. Pelabelan menggunakan yolo_mark

Gambar 23 menampilkan data yang telah diberi label menggunakan yolo_mark. Selanjutnya, saat proses pelabelan, setiap objek larva udang yang terdapat pada gambar diberi label sesuai dengan kelas yang telah ditentukan. Pada saat proses pelabelan berlangsung, setiap objek yang telah diberi label akan tersimpan kedalam file anotasi berekstensi .txt yang meyimpan nilai anotasi masing-masing citra seperti nama file, label dan koordinat *bounding box*. File anotasi dan data citra kemudian diunggah ke Google Drive untuk memudahkan akses data saat proses *training* di Google Colab.

2. Training

Setelah semua dataset gambar diberi label langkah selanjutnya adalah melakukan *training* untuk menghasilkan model yang akan dipakai untuk pendeteksian objek. Lamanya proses training tergantung dari seberapa besar dataset yang dibuat. Semakin banyak jumlah gambar semakin lama juga proses *training*nya, tetapi akurasi yang dihasilkan akan semakin baik.

Sebelum proses *training* perlu untuk mengkonfigurasi proses konvolusi. File untuk konfigurasi adalah file dengan ekstensi .cfg. yolo sudah menyediakan konfigurasi default. Untuk membuat konfigurasi baru dapat menyalin konfigurasi yang disediakan kemudian ubah sesuai dengan kebutuhan. Pada penelitian ini menggunakan konfigurasi sebagai berikut:

```
[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=1
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

Gambar 24. File konfigurasi

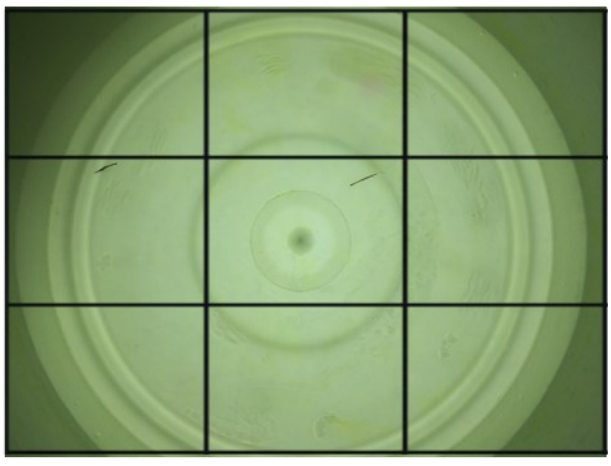
Konfigurasi yang perlu disesuaikan ada di bagian [convolutional] dan [region] yaitu filters dengan menggunakan rumus filter=(classes + 5)x3, dan classes yang ada penelitian ini memiliki 1 kelas. Konfigurasi diatas akan menghasilkan proses konvolusi sebagai berikut:

Tabel 2. Proses konvolusi

Layer	Size	stride	Output
Input	-	-	416 x 416 x 3
Conv 1	3 x 3	1	416 x 416 x 16
Max Pool 1	2 x 2	2	208 x 208 x 16
Conv 2	3 x 3	1	208 x 208 x 32
Max Pool 2	2 x 2	2	104 x 104 x 32
Conv 3	3 x 3	1	104 x 104 x 64
Max Pool 3	2 x 2	2	52 x 52 x 64
Conv 4	3 x 3	1	52 x 52 x 128
Max Pool 4	2 x 2	2	26 x 26 x 128
Conv 5	3 x 3	1	26 x 26 x 256
Max Pool 5	2 x 2	2	13 x 13 x 256
Conv 6	3 x 3	1	13 x 13 x 512
Max Pool 6	2 x 2	1	13 x 13 x 512
Conv 7	3 x 3	1	13 x 13 x 1024
Conv 8	3 x 3	1	13 x 13 x 1024
Conv 9	1 x 1	1	13 x 13 x 45

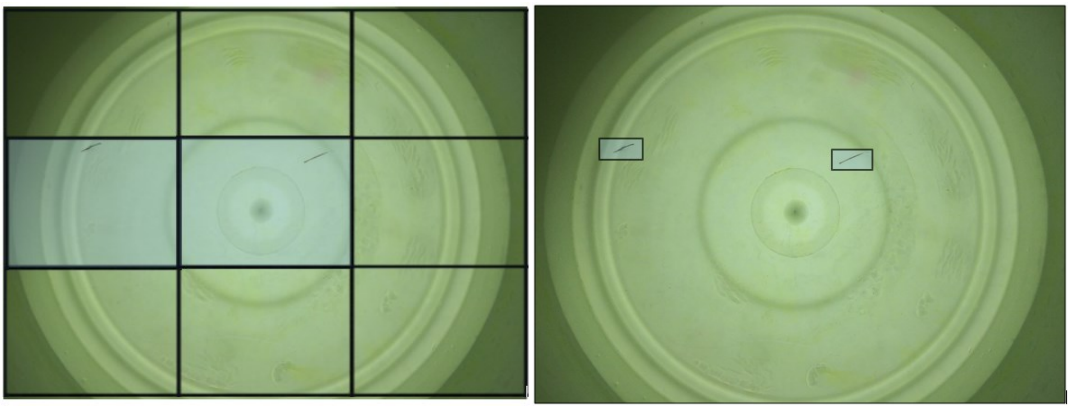
Berdasarkan arsitektur yang digunakan YOLO berikut proses prediksi terhadap dataset yang digunakan dengan konfigurasi diatas:

- *Input* gambar akan diubah menjadi 416 x 416 kemudian dibagi menjadi *grid* sesuai konfigurasi yang ditentukan yaitu S x S dimana $S = 3$ seperti Gambar 25 berikut:



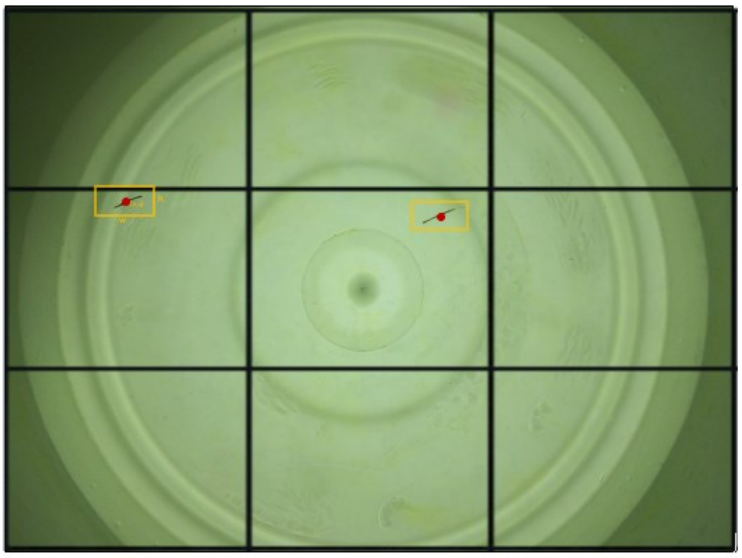
Gambar 25. Grid 3x3

- Kemudian setiap *grid* akan memprediksi *B Bounding Box* dan *C class probabilities* dimana nilai $B = 2$ dan $C = 1$ (Jumlah kelas yang digunakan). Prediksi *bounding box* ditunjukan pada Gambar 26 berikut:



Gambar 26. Class probabilities

- Hasil prediksi *Bounding Box* akan menghasilkan 5 komponen yaitu x, y, w, h dan *confidence.* x dan y adalah koordinat yang ada ditengah *box* dan bersifat relatif terhadap ukuran gambar yang ditunjukan dengan berikut.



Gambar 27. Komponen bounding box

Proses *training* YOLO membutuhkan *pretrained weight* dari YOLO serta sumber daya yang besar, untuk itu proses ini dilakukan di Google Colab. *Tools* ini digunakan agar proses pelatihan menjadi lebih cepat dengan memanfaatkan sumber daya GPU yang disediakan oleh

Google Colab. Tahap pertama adalah menghubungkan Google Drive dengan Google Colab, setelah terhubung, kemudian *download* folder data *training* dari *github* https://github.com/siskarmlv/trainingYOLO.git menggunakan perintah seperti Gambar 22. Data training yang telah di*download* akan tersimpan dalam file *darknet* yang berada di Google Drive.

```
% cd /content/drive/My Drive/darknet/
! git clone https://github.com/siskarmlv/trainingYOLO.git
% cd trainingYOLO
! make
```

Gambar 28. Perintah *download* data *training*

Selanjutnya men*download* model YOLO yang akan digunakan untuk training menggunakan perintah seperti pada Gambar 23.

```
! wget http://pjreddie.com/media/files/darknet53.conv.74
```

Gambar 29. Perintah *download* model YOLO

Setelah itu, melakukan *training* menggunakan file konfigurasi cfg menggunakan perintah seperti pada Gambar 30. Hasil dari proses pelatihan ini akan menghasilkan file bobot akhir dalam ekstensi .weights, yang akan digunakan saat proses pengujian.

```
! ./darknet detector train data/obj.data cfg/small-
larva.cfg darknet53.conv.74 -dont_show
```

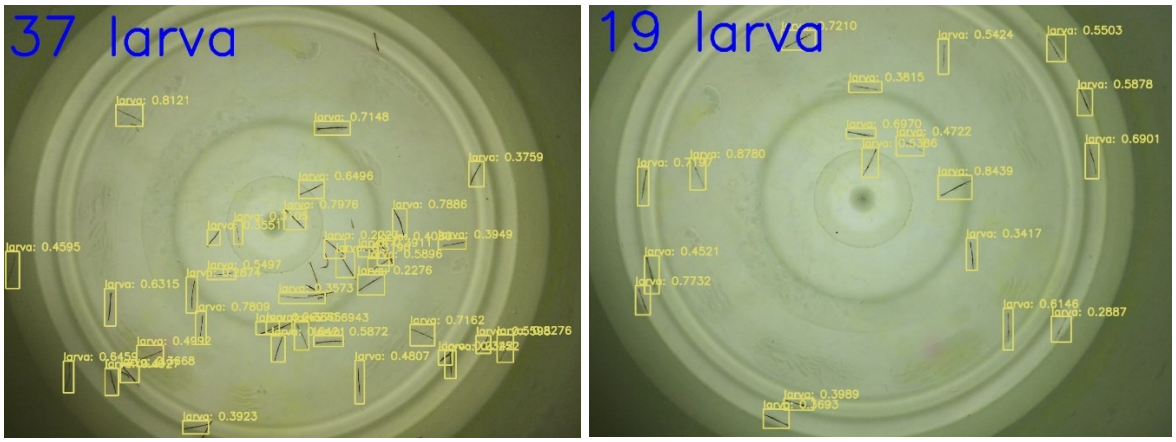
Gambar 30. Perintah *training*

3. Testing

Pada saat testing akan dilakukan proses yang menampilkan kemampuan model dalam melakukan penalaran dengan menggunakan pengetahuan yang telah diberikan saat proses *training* untuk menghasilkan suatu kesimpulan dalam mendeteksi larva udang. Data yang digunakan untuk proses testing sebanyak 99 gambar. Proses testing dilakukan menggunakan

anaconda *prompt*. Proses testing diawali dengan memasukkan data uji ke dalam sistem seperti pada Gambar 31. Kemudian dilakukan proses deteksi dengan menggunakan file kompilasi darknet, file .cfg yang digunakan saat pelatihan, dan file model .weight yang dihasilkan pada proses *training.* Hasil dari proses ini menghasilkan gambar yang sama dengan bounding box yang terdapat disetiap objek, *class object, confidence score,* serta jumlah objek yang terdeteksi sebagai larva udang seperti terlihat pada Gambar 32.

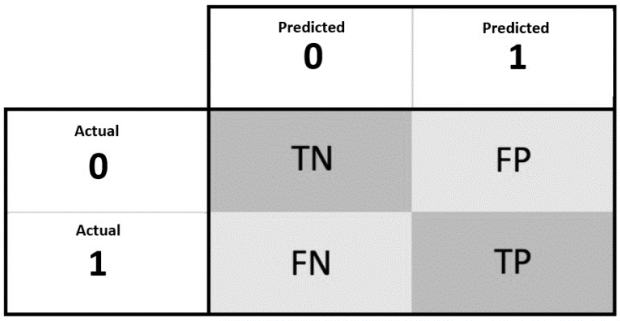
Gambar 31. Proses testing menggunakan YOLO



Gambar 32. Contoh hasil deteksi

F. ANALISIS KERJA SISTEM

Analisis kerja sistem deteksi larva menggunakan YOLOv3 dapat dilakukan dengan menghitung ketepatan prediksi yang dilakukan sistem terhadap data citra inputan pada proses *testing* dengan menggunakan bantuan *confussion matrix*. *Confusion Matrix* merupakan tabel yang didalamnya menyimpan informasi perbandingan antara klasifikasi oleh sistem dan klasifikasi yang sebenarnya dari data. Cara perhitungan confusion matrix dapat dilihat pada Gambar 33.



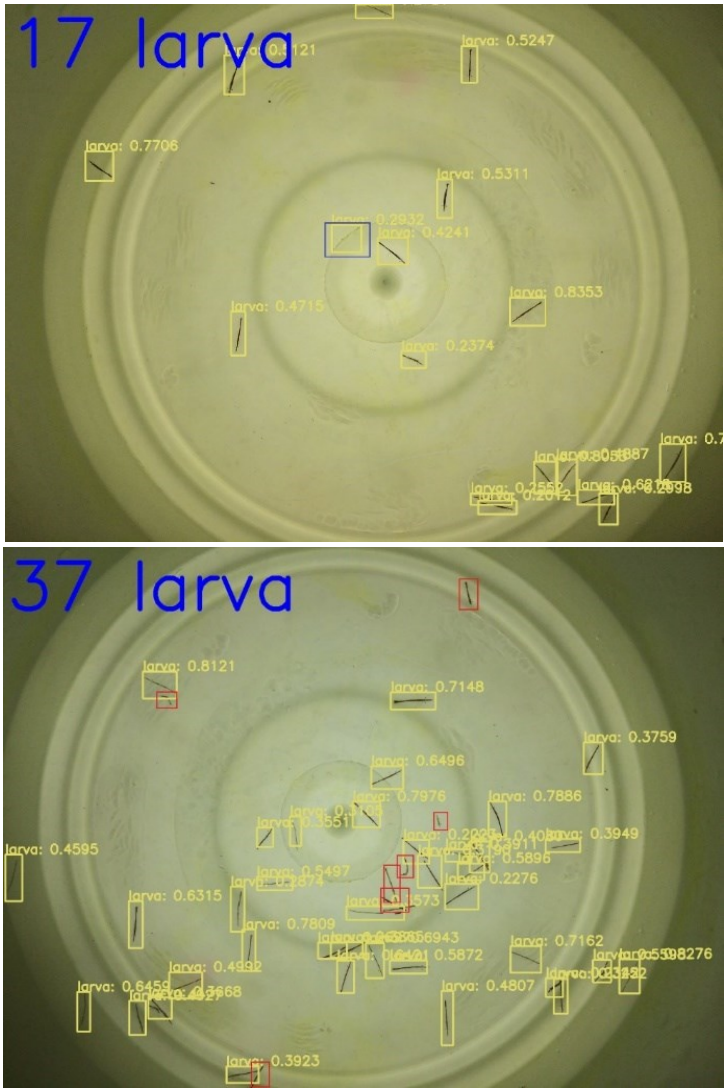
Gambar 33. Confusion Matrix

Confusion matrix terdiri atas empat bagian, yaitu:

1. *True positive* : yaitu jumlah prediksi benar dari data positif.
2. *False Positive* : yaitu jumlah prediksi salah dari data positif.
3. *False Negative* : yaitu jumlah prediksi salah dari data negatif.
4. *True Negative* : yaitu jumlah prediksi benar dari data negatif.

Pada penelitian ini, hanya terdapat nilai TP, FP dan FN. Hasil deteksi dianggap sebagai *True positive* (TP) jika objek larva udang (aktual positif) terdeteksi sebagai larva udang oleh sistem (prediksi positif). Jika objek larva udang (aktual positif) tidak terdeteksi sebagai larva udang oleh sistem (prediksi negatif), maka hasil deteksi dianggap sebagai *False Negative* (FN). Jika sistem mendeteksi objek sebagai larva udang (prediksi positif) padahal objek tersebut sebenarnya bukanlah larva udang (aktual negatif), maka hasil deteksi dimasukkan sebagai *False Positive* (FP), dan jika sistem mendeteksi sebagai objek bukan larva udang (prediksi

negeatif) dan objek tersebut memang bukan larva udang (actual negatif), maka hasil deteksi dianggap sebagai *True Negative* (TN). Pada Gambar 34 merupakan contoh *True Positive* (TP) dengan *bounding box* berwarna kuning, sedangkan *box* berwarna merah merupakan nilai *False Negative* (FN) pada hasil deteksi sistem.



Gambar 34. Contoh TP, FP dan FN

Ket:

: True Positive (TP)

: False Negative (FN)

: False Positive (FP)

Dari hasil perhitungan *Confusion Matrix* kemudian dapat dihitung nilai *Accuracy,* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Nilai akurasi dapat diperoleh dengan persamaan (1) berikut.

$$\text{Akurasi} = \left(\frac{TP+TN}{TP+FP+FN+TN}\right) \text{ x } 100\% \quad (1)$$

Setelah nilai akurasi didapatkan, kemudian dilakukan perhitungan nilai *root mean square* (RMSE) untuk mengestimasi besarnya kesalahan deteksi. Nilai RMSE dapat diperoleh dengan persamaan (2) berikut.

$$\text{RMSE} = \frac{\sqrt{\sum(x-y)^2}}{n} \quad (2)$$

Ket:

x = Nilai data Aktual

y = Nilai hasil peramalan

n = banyaknya data

$\sum$ = Summation (Jumlahkan keseluruhan nilai)

Nilai *root mean square error* (RMSE) merupakan besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) nilai RMSE maka hasil prediksi akan semakin akurat.

BAB IV

HASIL DAN PEMBAHASAN

A. HASIL PENELITIAN

Pada bab ini, disajikan hasil kerja sistem deteksi larva udang pada video dan citra menggunakan YOLOv3. Adapun jumlah data yang digunakan adalah sebanyak 424 citra, yang dibagi menjadi 325 data latih dan 99 data uji. Tabel 2 merupakan hasil dari perhitungan *confusion matrix* yang didapat dari 99 data uji.

Tabel 3. Confusion Matrix

TP	FP	FN	TN
4240	**1**	**660**	**0**

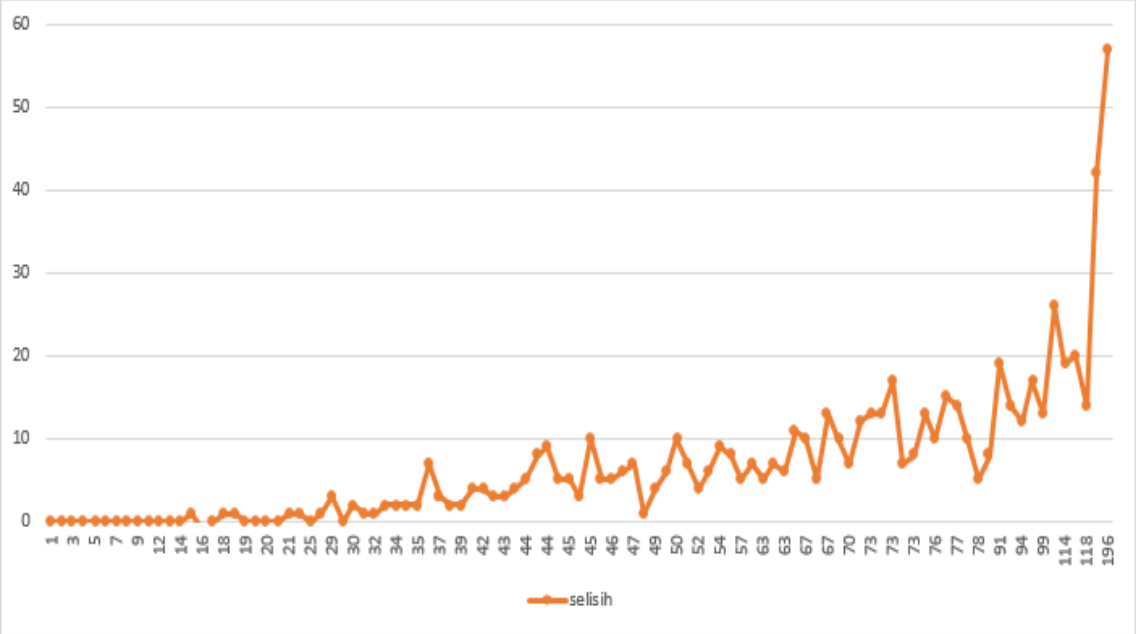
Pada tabel diatas menunjukkan bahwa dari 99 data uji didapatkan nilai FN sebanyak 660 dan nilai FP yaitu 1. Untuk lebih jelasnya dapat dilihat pada Tabel 3 yang merupakan hasil perbandingan antara perhitungan manual dengan perhitungan sistem.

Tabel 4. Hasil Perbandingan

Image	Manual	TP	FN	FP	TN	Image	Manual	TP	FN	FP	TN
1	1	1	0	0	0	**51**	45	35	10	0	0
2	2	2	0	0	0	**52**	46	41	5	0	0
3	3	3	0	0	0	**53**	46	41	5	0	0
4	4	4	0	0	0	**54**	46	40	6	0	0
5	5	5	0	0	0	**55**	47	40	7	0	0
6	6	6	0	0	0	**56**	48	47	1	0	0
7	7	7	0	0	0	**57**	49	45	4	0	0
8	8	8	0	0	0	**58**	49	43	6	0	0
9	9	9	0	0	0	**59**	50	40	10	0	0
10	10	10	0	0	0	**60**	50	43	7	0	0
11	12	12	0	0	0	**61**	52	48	4	0	0
12	13	13	0	0	0	**62**	53	47	6	0	0
13	14	14	0	0	0	**63**	54	45	9	0	0
14	15	14	1	0	0	**64**	54	46	8	0	0
15	16	17	0	1	0	**65**	57	52	5	0	0
16	17	17	0	0	0	**66**	60	53	7	0	0
17	18	17	1	0	0	**67**	63	58	5	0	0
18	19	18	1	0	0	**68**	63	56	7	0	0
19	19	19	0	0	0	**69**	63	57	6	0	0

20	19	19	0	0	0	**70**	65	54	11	0	0
21	20	20	0	0	0	**71**	67	57	10	0	0
22	21	21	0	0	0	**72**	67	62	5	0	0
23	21	20	1	0	0	**73**	67	54	13	0	0
24	24	23	1	0	0	**74**	69	59	10	0	0
25	25	25	0	0	0	**75**	70	63	7	0	0
26	28	27	1	0	0	**76**	71	59	12	0	0
27	29	26	3	0	0	**77**	73	60	13	0	0
28	30	30	0	0	0	**78**	73	60	13	0	0
29	30	28	2	0	0	**79**	73	56	17	0	0
30	32	31	1	0	0	**80**	73	66	7	0	0
31	32	31	1	0	0	**81**	73	65	8	0	0
32	33	31	2	0	0	**82**	75	62	13	0	0
33	34	32	2	0	0	**83**	76	66	10	0	0
34	34	32	2	0	0	**84**	77	62	15	0	0
35	35	33	2	0	0	**85**	77	63	14	0	0
36	35	28	7	0	0	**86**	78	68	10	0	0
37	37	34	3	0	0	**87**	78	73	5	0	0
38	39	37	2	0	0	**88**	81	73	8	0	0
39	39	37	2	0	0	**89**	91	72	19	0	0
40	39	35	4	0	0	**90**	92	78	14	0	0
41	42	38	4	0	0	**91**	94	82	12	0	0
42	42	39	3	0	0	**92**	96	79	17	0	0
43	43	40	3	0	0	**93**	99	86	13	0	0
44	43	39	4	0	0	**94**	109	83	26	0	0
45	44	39	5	0	0	**95**	114	95	19	0	0
46	44	36	8	0	0	**96**	116	96	20	0	0
47	44	35	9	0	0	**97**	118	104	14	0	0
48	45	40	5	0	0	**98**	125	83	42	0	0
49	45	40	5	0	0	**99**	196	139	57	0	0
50	45	42	3	0	0	**TOTAL**	4899	4240	660	1	0

Pada tabel diatas menunjukkan hasil perbandingan perhitungan larva udang antara perhitungan sistem dengan perhitungan manual serta selisih dari perbandingan tersebut. Dapat dilihat pada data ke-15, hasil perhitungan sistem melebihi dari hasil perhitungan manual sehingga data inilah yang disebut sebagai False Positive (FP). Berikut grafik hasil perbandingan ini dapat dilihat pada Gambar 35.



Gambar 35. Grafik selisih

Gambar 35 memperlihatkan grafik selisih dari perbandingan antara perhitungan manual dan perhitungan sistem. Pada kepadatan < 50 menunjukkan selisih (FN) ≤ 20. Sedangkan pada kepadatan > 50 selisihnya mencapai > 50. Hal ini disebabkan karena banyaknya objek yang tumpang tindih. Selanjutnya akan dihitung nilai *root mean square error* (RMSE)nya dengan 2 kondisi, yaitu pada kepadatan < 50 dan pada kepadatan > 50 dengan menggunakan 25 data uji. Pada tabel 4 merupakan hasil perhitungan RMSE dengan kepadatan > 50.

Tabel 5. Perhitungan RMSE kepadatan < 50

Image	Prediction	Corrected Prediction	Manual	Error	Error^2	Accuracy
1	1	1	1	0	0	100%
2	2	2	2	0	0	100.00%
3	3	3	3	0	0	100.00%
4	4	4	4	0	0	100.00%
5	14	15	15	0	0	100.00%
6	26	29	29	0	0	100.00%
7	30	33	30	3	9	91.67%
8	28	31	35	-4	16	88.57%
9	34	37	37	0	0	100.00%
10	35	39	39	0	0	100.00%
11	38	42	42	0	0	100.00%
12	39	43	43	0	0	100.00%
13	39	43	44	-1	1	97.73%

14	36	40	44	-4	16	90.91%
15	35	39	44	-5	25	88.64%
16	40	44	45	-1	1	97.78%
17	40	44	45	-1	1	97.78%
18	35	39	45	-6	36	86.67%
19	41	45	46	-1	1	97.83%
20	41	45	46	-1	1	97.83%
21	40	44	46	-2	4	95.65%
22	40	44	47	-3	9	93.62%
23	43	47	49	-2	4	95.92%
24	40	44	50	-6	36	88.00%
25	43	47	50	-3	9	95.92%
					Rata-rata akurasi	96.18%
					RMSE	2.60

Pada tabel diatas merupakan hasil perhitungan pada kepadatan < 50, rata-rata akurasi yang didapatkan sebesar 96.18%, nilai ini cukup tinggi untuk sebuah deteksi objek, serta menghasilkan nilai RMSE sebesar 2.60 dengan menggunakan faktor koreksi 1.1, nilai ini cukup rendah sehingga menunjukkan bahwa metode YOLO dapat mendeteksi objek kecil seperti larva udang. Selanjutnya dilakukan perhitungan RMSE pada kepadatan > 50.

Tabel 6. Perhitungan RMSE kepadatan > 50

Image	**Prediction**	**Corrected Prediction**	**Manual**	**Error**	**Error^2**	**akurasi**
1	48	58	52	6	36	90.63%
2	47	56	53	3	9	94.92%
3	45	54	54	0	0	100.00%
4	46	55	54	1	1	98.21%
5	52	62	57	5	25	92.54%
6	53	64	60	4	16	94.12%
7	56	67	63	4	16	94.37%
8	57	68	63	5	25	93.15%
9	54	65	65	0	0	100.00%
10	57	68	67	1	1	98.55%
11	62	74	67	7	49	91.36%
12	59	71	69	2	4	97.26%
13	63	76	70	6	36	92.68%
14	59	71	71	0	0	100.00%
15	66	79	73	6	36	92.94%
16	65	78	73	5	25	93.98%
17	66	79	76	3	9	96.34%
18	68	82	78	4	16	95.35%
19	73	88	81	7	49	92.63%
20	78	94	92	2	4	97.92%
21	82	98	94	4	16	96.08%
22	86	103	99	4	16	96.26%

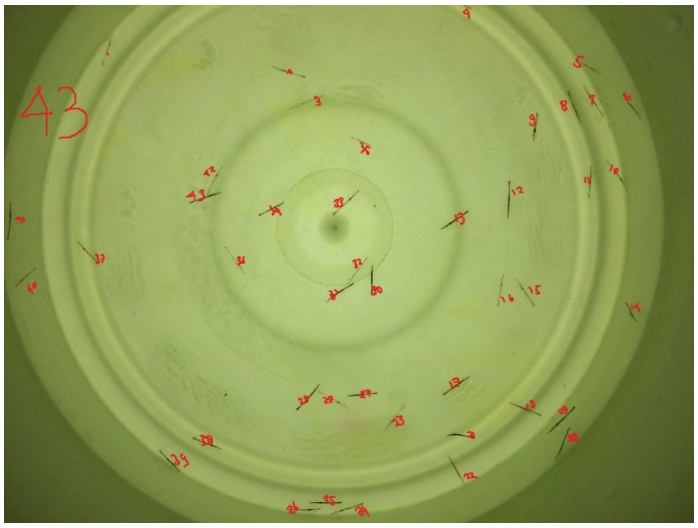
23	95	114	114	0	0	100.00%
24	96	115	116	-1	1	99.14%
25	104	125	118	7	49	94.70%
					Rata-rata akurasi	95.72%
					RMSE	4.19

Pada tabel diatas merupakan hasil perhitungan pada kepadatan > 50, rata-rata akurasi yang didapatkan sebesar 95.72%, serta menghasilkan nilai RMSE sebesar 4.19 dengan menggunakan faktor koreksi 1.2, nilai yang dihasilkan cukup besar dikarenakan tumpang tindih udang.

B. PEMBAHASAN

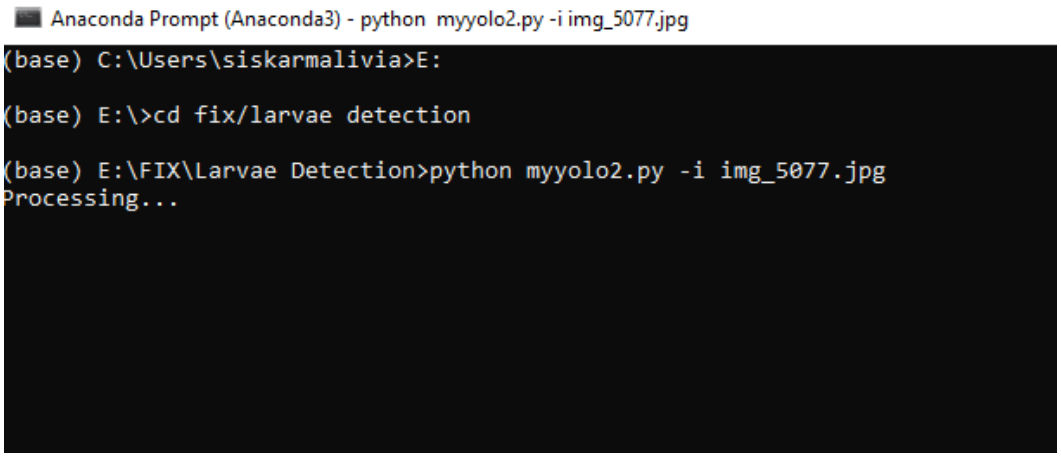
Proses perhitungan larva udang dimulai dengan mengambil objek dalam hal ini larva udang, di Balai Perikanan Budidaya Air Payau, Takalar. Selanjutnya, objek tersebut dimasukkan kedalam wadah putih dengan berisikan air setinggi 2 cm. kemudian objek tersebut akan difoto menggunakan kamera *smartphone* yang disimpan diatas wadah. Proses pengambilan foto dapat dilihat pada **Lampiran 1**.

Selanjutnya dilakukan perhitungan secara manual pada foto yang telah diambil. Perhitungan manual pada penelitian ini dilakukan dengan menghitung objek satu per satu menggunakan aplikasi *paint* seperti pada Gambar 36.



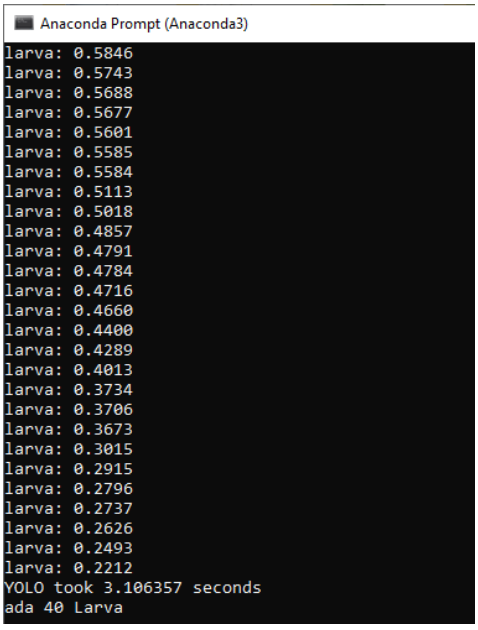
Gambar 36. Perhitungan manual menggunakan aplikasi *paint*

Setelah dilakukan perhitungan manual, selanjutnya sistem diuji coba, menggunakan *anaconda prompt* dengan gambar yang sama seperti pada gambar Gambar 37.



Gambar 37. Proses penghitungan larva udang

Setelah proses perhitungan selesai, maka akan muncul jumlah deteksi serta nilai *confidence score* untuk setiap objek yang terdeteksi. Hasil perhitungan dapat dilihat pada Gambar 38 dengan waktu deteksi adalah 3.10 per detik dengan hasil deteksi sebanyak 40 larva. Data lengkap waktu perhitungan menggunakan program ditampilkan pada **Lampiran 2** dengan rata-rata waktu perhitungan sebesar 2.67 detik, jika dibandingan pada penelitian sebelumnya, maka metode ini cukup memakan waktu dalam proses deteksi objek yang sangat tipis.



Gambar 38. Hasil perhitungan pada sistem

Pada penelitian ini, dilakukan percobaan dengan menggunakan parameter *batch, subdivision, momentum, width, height, channel, learning rate*, dan *iterations default* yang telah diatur dalam file .cfg. Nilai *default* untuk masing-masing parameter adalah *batch* 64, *subdivision* 16, *momentum* 0.09, *width* 416, *height* 416, *channel* 3, *learning rate* 0.001. Sistem yang dibuat menggunakan parameter-parameter berikut dalam proses pelatihan.

a. Batch

Parameter *batch* menunjukkan ukuran *batch* yang digunakan dalam proses training. Data training biasanya terdiri dari ratusan bahkan ribuan citra. Dalam sistem yang dibuat ini, nilai *batch size* adalah 64. Artinya, dalam satu iterasi terdapat 64 citra yang digunakan untuk mengupdate parameter dari *neural network*.

b. Subdivision

Meskipun jumlah citra yang diproses dibatasi menjadi 64 citra dalam satu iterasi, hal ini belum cukup karena bisa saja GPU menjadi out of memory saat proses training. Dengan memberikan nilai subdivision menjadi 16, maka data yang akan diproses adalah 64/16 atau 4 citra dalam satu waktu. Hal ini dilakukan untuk menghindari error akibat GPU kekurangan memori. Saat testing, nilai batch dan subdivision akan di set menjadi 1.

c. *Width, height*, dan *Channel*

Parameter-parameter ini menunjukkan ukuran citra *inputan*. Nilai untuk *width* x *height* yang digunakan adalah 416 x 416 piksel. Hasil deteksi bisa saja lebih baik jika ukuran *input* diubah menjadi 832 x 832 piksel misalnya, akan tetapi akan membutuhkan waktu training yang jauh lebih lama dan sistem pendeteksian akan menjadi lebih lambat. Sementara nilai channel = 3 untuk menunjukkan bahwa citra yang akan diproses adalah citra 3-channel RGB (*Red, Green, Blue*).

d. Momentum

Nilai *weights* yang terdapat pada sebuah *neural network* di *update* menggunakan *batch* citra tidak dengan keseluruhan data citra secara bersamaan. Hal ini akan menyebabkan fluktuasi pada proses *update weigh*t. Nilai momentum 0.9 diberikan sebagai pembatas bahwa perubahan nilai *weights* tidak boleh melebihi 0.9 dari *weight* sebelum proses *update.*

e. *Learning Rate, Steps*, dan *Scale*

Learning rate mengontrol seberapa agresif sistem akan belajar berdasarkan *batch* data inputannya. Biasanya nilai *learning rate* antara 0.01 hingga 0.0001. Pada sistem yang dibuat, nilai *learning rate* yang digunakan adalah 0.001, di awal proses *training, learning rate* diset tinggi kemudian semakin lama semakin kecil untuk mengurangi agresifitas proses *learning*. *Steps* adalah parameter yang digunakan sebagai *threshold* untuk *learning rate*, nilai *steps* pada sistem ini adalah 4000, sehingga nilai *learning rate* akan diperkecil dengan cara mengalikannya dengan parameter *scale* yang memiliki nilai 0.1 apabila iterasi sudah mencapai 4000 *steps*.

f. *Max Batches* (Iterations)

Parameter *max batches* menspesifikasikan seberapa banyak iterasi yang harus dilakukan untuk menyelesaikan proses training. Dalam sistem yang dibuat ini nilai *max batches* diset pada 4000 iterasi.

Dengan konfigurasi parameter-parameter diatas, sistem yang telah dibuat dilatih dengan 4.000 iterasi dengan jumlah data latih sebanyak 325 citra, kemudian dilakukan proses uji yang memberikan hasil akurasi sebesar 96.84%. Pada pengujian dengan menggunakan citra yang memiliki objek lain selain larva udang, sistem yang dibuat terkadang memprediksi objek lain seperti lidi sebagai salah satu jenis larva udang. Hal ini dikarenakan sistem dengan algoritma YOLO akan memprediksi kemungkinan suatu objek dalam sebuah input yang diterima kemudian akan mengklasifikasikan kelas atau jenis objek tersebut. Proses klasifikasi pada YOLOv3 menggunakan *logistic regression* sehingga kelas dan *score* tertinggi akan langsung

di*assign* ke dalam objek yang terdeteksi. Sistem yang dibuat hanya memiliki 1 kelas yaitu larva. Sehingga objek yang terdeteksi hanya memiliki pilihan untuk diklasifikasikan ke dalam kelas tersebut.

Untuk mencegah terjadiya kesalahan dalam klasifikasi objek, sistem yang dibuat diberi *threshold* 0.25, artinya objek yang dideteksi hanya akan divisualisasikan *bounding box*nya apabila *score* kelasnya diatas 0.25, akan tetapi dalam beberapa kasus, objek lain seperti lidi bisa jadi memiliki skor objek diatas 0.25, hal ini dipengaruhi oleh objek larva udang yang memiliki bentuk panjang dan tipis, sehingga kemungkinan objek lain dapat terprediksi sebagai larva cukup tinggi.

Pada Gambar 39 dapat dilihat bahwa sistem terkadang tidak mendeteksi objek larva udang, dikarenakan score yang diperoleh tidak mencapai score threshold minimum objeknya. Cara lain untuk mencegah terjadinya kesalahan pendeteksian adalah dengan cara melakukan pelatihan ulang, dimana sistem dilatih dengan tambahan objek lain selain larva udang.

Gambar 39. Hasil deteksi sistem

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan analisis yang dilakukan terhadap hasil deteksi larva udang menggunakan metode *YOLOv3* dapat ditarik kesimpulan antara lain:

1. Metode YOLOv3 dapat digunakan untuk mendeteksi dan menghitung jumlah larva sehingga membantu para petambak udang menghitung larva dalam jumlah yang besar, yaitu ribuan hingga ratusan ekor larva udang, serta menghindari kesalahan hasil agar tidak ada pihak yang dirugikan.
2. Dataset yang digunakan pada penelitian ini sebanyak 325 data gambar sebagai data latih dan 99 data uji.
3. Pada kepadatan 1 hingga 14 sistem dapat mendeteksi larva dengan akurat sehingga menghasilkan nilai *root mean square error* (RMSE) untuk kepadatan < 50 sebesar 2.60 serta akurasi sebesar 96.18%. Pada kepadatan > 50 menghasilkan nilai *root mean square error* (RMSE) sebesar 4.19 dikarenakan tumang tindih udang, serta akurasi sebesar 95.72%.

B. SARAN

Adapun saran dari penulis adalah sebagai berikut:

1. Sistem yang telah dibuat ini dapat dikembangkan lebih lanjut terkait jenis dan panjang larva yang dapat dideteksi oleh sistem dan meningkatkan akurasi sistem yang sudah ada. Sistem yang telah dibuat juga dapat dimodifikasi untuk mendeteksi objek lain selain larva udang dengan mengganti data latih pada sistem untuk mengatasi permasalahan lainnya.

2. Untuk mencegah terjadinya kesalahan pendeteksian objek sebaiknya ditambahkan data negatif berupa objek lain saat proses pelatihan agar sistem yang dibuat dapat membedakan antara objek larva udang dengan objek lainnya.
3. Untuk penelitian selanjutnya dapat dicoba perhitungan benih dengan sudut pengambilan gambar yang berbeda serta dilakukan berulang kali untuk meningkatkan akurasi perhitungan.

DAFTAR PUSTAKA

Admin. (2019). *siklus hidup udang vaname*. https://agrikan.id/siklus-hidup-udang-vaname/

AlexeyAB. (2019). *AlexeyAB/Yolo_mark: GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2*. https://github.com/AlexeyAB/Yolo_mark

Ambalina, L. (2019). *What is Image Annotation? – An Intro to 5 Image Annotation Services*. https://hackernoon.com/what-is-image-annotation-an-intro-to-5-image-annotation-services-yt6n3xfj

Awalludin, E. A., Mat Yaziz, M. Y., Abdul Rahman, N. R., Yussof, W. N. J. H. W., Hitam, M. S., & T Arsad, T. N. (2019). Combination of Canny Edge Detection and Blob Processing Techniques for Shrimp Larvae Counting. *Proceedings of the 2019 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2019*, *2011*, 308–313. https://doi.org/10.1109/ICSIPA45851.2019.8977746

Awalludin, E. A., Wan Muhammad, W. N. A., Arsad, T. N. T., & Wan Yussof, W. N. J. H. (2020). Fish Larvae Counting System Using Image Processing Techniques. *Journal of Physics: Conference Series*, *1529*(5). https://doi.org/10.1088/1742-6596/1529/5/052040

Boyd, C. E., & Clay, J. (2002). Evaluation of Belize Aquaculture Ltd : A Superintensive Shrimp Aquaculture System E V A L U A T I O N O F B E L I Z E A Q U A C U L T U R E L T D : *By the Consortium*, 17.

Clarke, H., Horine, B., & Thomas-Hall, P. L. (2018). Image processing pipeline for automated larva counting. *I2MTC 2018 - 2018 IEEE International Instrumentation and Measurement Technology Conference: Discovering New Horizons in Instrumentation and Measurement, Proceedings*, 1–5. https://doi.org/10.1109/I2MTC.2018.8409867

Duraiappah, A. K., Israngkura, A., & Sae-hae, S. (2000). *Sustainable Shrimp Farming : Estimations of a Survival Function*. *31*.

Fast, A. W. ., & Lester, L. J. (1992). *Marine Shrimp Culture*. Elsevier. https://doi.org/10.1016/C2009-0-01033-2

Flores, A., Crisóstomo, P., & López, J. (2008). Peruvian scallop larvae counting system using image processing techniques. *Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, ICCDCS*, 8–11. https://doi.org/10.1109/ICCDCS.2008.4542660

Git. (n.d.). https://git-scm.com/

Haliman, R. W. (2005). *Udang Vannamei*. Penebar Swadaya. https://onesearch.id/Record/IOS6976.slims-6987#details

Hendarajat, E. A., Mangampa, M., & Suryanto, H. (2007). *BUDI DAYA UDANG VANNAMEI (Litopenaeus vannamei) POLA TRADISIONAL PLUS DI KABUPATEN MAROS, SULAWESI SELATAN*. *2*, *NO. 2*. https://doi.org/http://dx.doi.org/10.15578/ma.2.2.2007.67-70

Kaewchote, J., Janyong, S., & Limprasert, W. (2018). Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp. *Agriculture and Natural Resources*, *52*(4), 371–376. https://doi.org/10.1016/j.anres.2018.10.007

Khantuwan, W., & Khiripet, N. (2012). Live Shrimp Larvae Counting Method Using Co-occurrence Color Histogram. *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 1–4.

Lainez, S. M. D., & Gonzales, D. B. (2019). Automated fingerlings counting using convolutional neural network. *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, 67–72. https://doi.org/10.1109/CCOMS.2019.8821746

Loh, B. C. S., Raman, V., & Then, P. H. H. (2011). First Prototype of Aquatic Tool Kit: Towards Low-Cost Intelligent Larval Fish Counting in Hatcheries. *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, 192–195. https://doi.org/10.1109/DASC.2011.53

Low, J. (2020). *What is Image Annotation*. https://medium.com/supahands-techblog/what-is-image-annotation-caf4107601b7

Menristek. (2003). *Budidaya Udang Windu*. http//:agricta.org.com

Nguyen, K. T., Nguyen, C. N., Wang, C. Y., & Wang, J. C. (2020). Two-phase instance segmentation for whiteleg shrimp larvae counting. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, *2020-Janua*, 1–3. https://doi.org/10.1109/ICCE46568.2020.9043075

Novrihansa, R., Karnila, R., & Suparmi. (2016). *PENGARUH PENAMBAHAN KONSENTRASI GARAM BERBEDA SELAMA PEREBUSAN TERHADAP KANDUNGAN KOLESTEROL UDANG PUTIH (Penaeus indicu)*.

Nurlaela, N., Niswar, M., Nurtanio, I., Fujaya, Y., Kashihara, S., & Fall, D. (2019). Detection of Megalopa Phase Crab Larvae Using Digital Image Processing. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 269–272. https://doi.org/10.1109/ISRITI48646.2019.9034609

OEMUJATI, B. S. (1990). *Taksonomi avertebrata: pengantar praktikum laboratorium*. U.I. PRESS , 1990. https://onesearch.id/Record/IOS3239.slims-61356

Oshrimp. (2020). *5 Daerah Penghasil Udang Terbesar di Indonesia*. https://www.oshrimpseafood.com/2020/01/5-daerah-penghasil-udang-terbesar-di.html

Pokhrel, S. (2020). *Image Data Labelling and Annotation*. https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). Yolo V2.0. *Cvpr2017*, *April*, 187–213. http://www.worldscientific.com/doi/abs/10.1142/9789812771728_0012

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv*.

Robert B.Fisher, W. (2013). *Dictionary of Computer Vision and Image Processing, 2nd Edition*.

SEMINAR, K. (2000). The Design of Baby Fish Counter with Parallel Sensors. *Buletin Keteknikan Pertanian*, *14 NO.2*.

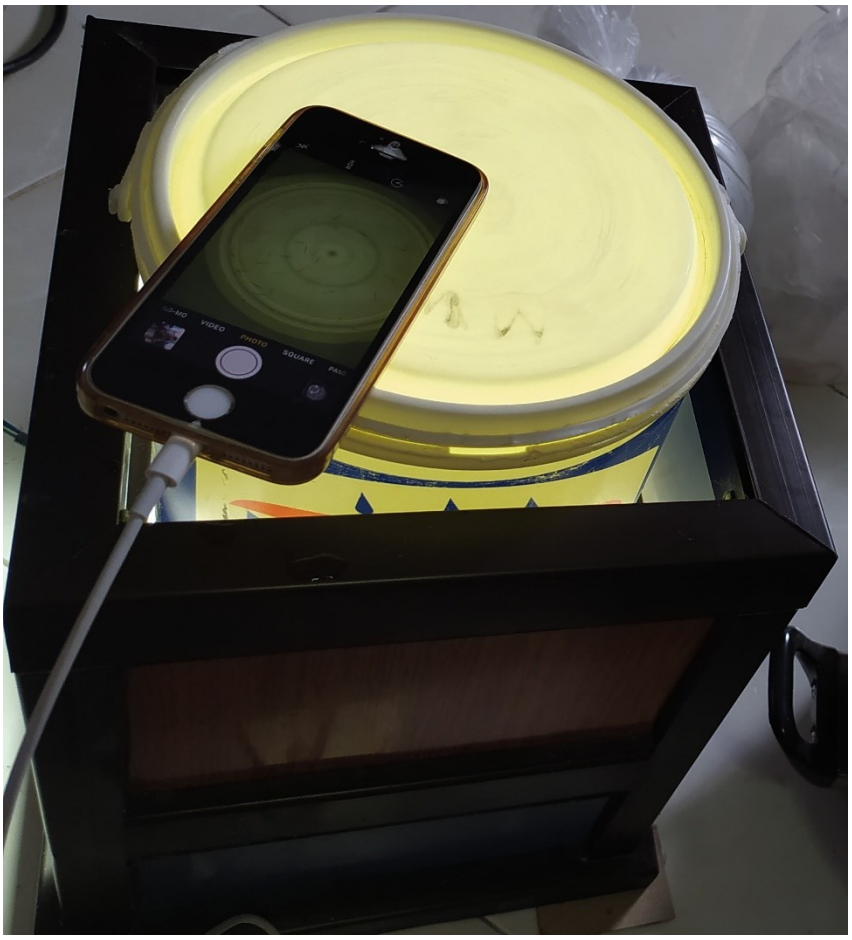
Solahudin, M., Slamet, W., & Dwi, A. S. (2018). Vaname (Litopenaeus vannamei) Shrimp Fry Counting Based on Image Processing Method. *IOP Conference Series: Earth and Environmental Science*, *147*(1). https://doi.org/10.1088/1755-1315/147/1/012014

Sterrer, W. (1986). Marine Fauna and Flora of Bermuda. A Systematic Guide to the Identification of Marine Organisms . *The Quarterly Review of Biology*, *61*(4), 566–567. https://doi.org/10.1086/415223

Tyagi, V. (2018). Understanding Digital Image Processing. *Understanding Digital Image Processing*, *November*. https://doi.org/10.1201/9781315123905

Zulfikar, wildan gayuh. (2020). *Kondisi Terkini Tambak Udang Indonesia: Trend Harga, Ekspor, dan Efek Pandemi*. https://app.jala.tech/kabar_udang/kondisi-terkini-tambak-udang-indonesia-trend-harga-ekspor-dan-efek-pandemi?redirect=https%3A%2F%2Fapp.jala.tech%2Fkabar_udang%2Fwabah-virus-corona-covid-19-tidak-mengganggu-budidaya-udang-indonesia

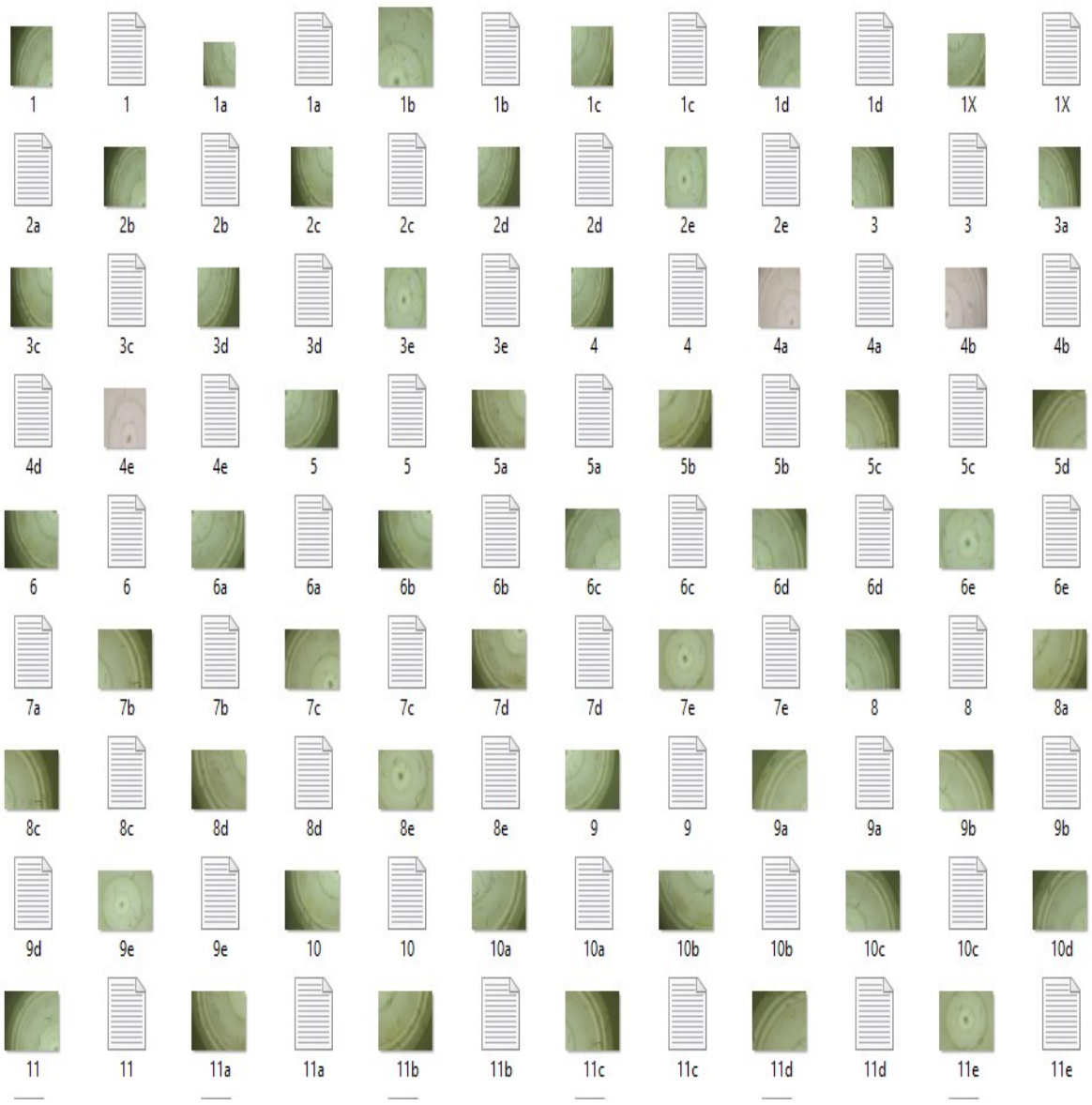
Lampiran 1. Proses Pengambilan Foto Larva Udang



Lampiran 2. Waktu Perhitungan Menggunakan Program Kepadatan < 50

Jumlah benih udang	Waktu perhitungan
1	2.54
2	2.49
3	2.6
4	2.35
15	2.7
29	2.99
30	2.61
35	2.72
37	2.65
39	3.08
42	3.14
43	2.73
44	2.88
44	2.7
44	2.9
45	3.19
45	2.85
45	2.65
46	2.34
46	2.31
46	2.3
47	2.84
49	2.98
50	2.15
50	2.15
Rata- rata	2.67

Lampiran 3. Data Training Sistem



Lampiran 4. File Konfigurasi Training small-larva-test.cfg

```
[net]
# Testing
batch=1
subdivisions=1
# Training
#batch=64
#subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches =4000
policy=steps
steps=1600,1800
scales=.1,.1

filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,
116,90,  156,198,  373,326
classes=1
num=9
```

Lampiran 5. Kode Program Pengujian Sistem

```
# import packages
import numpy as np
import argparse
import time
import cv2
import os

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
    help="minimum probability to filter weak detections")
ap.add_argument("-t", "--threshold", type=float, default=0.5,
    help="threshold when applying non-maxima suppression")
args = vars(ap.parse_args())

# load the YOLO class labels
labelsPath = "data/obj.names"
LABELS = open(labelsPath).read().strip().split("\n")

# paths to the YOLO weights and model configuration
weightsPath = "newweight/small-larva_final.weights"
configPath = "cfg/small-larva-test.cfg"

# initialize a list of colors to represent each possible class label
COLORS = (103, 220, 225)
# np.random.seed(42)
# COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
#   dtype="uint8")

# load our YOLO object detector
print("Processing...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

# load input image and grab its spatial dimensions
image = cv2.imread(args["image"])
# image = cv2.resize(image, (0,0), fx=0.7, fy=0.7)
(H, W) = image.shape[:2]

# determine only the *output* layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

# construct a blob from the input image and then perform a forward
# pass of the YOLO object detector, giving us our bounding boxes and
```

```
# associated probabilities
blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),swapRB=True,
crop=False)
net.setInput(blob)
start = time.time()
layerOutputs = net.forward(ln)
end = time.time()
print("Nilai blob: {}".format(blob.shape))

# initialize our lists of detected bounding boxes, confidences, and
# class IDs, respectively
boxes = []
confidences = []
classIDs = []

# loop over each of the layer outputs
for output in layerOutputs:
    # loop over each of the detections
    for detection in output:
        # extract the class ID and confidence (i.e., probability) form Image
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]
        # print ("Nilai Confidence dari object ialah :", confidence)
        # print ("Nilai ID dari Class ialah :", classID)

        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > args["confidence"]:
            # scale the bounding box coordinates back relative to the size of
the image
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")
            # print ("Nilai dari B.Box ialah :", centerX," ", centerY, " ",
width, " ", height)

            # use the center (x, y)-coordinates to get the top and and left
            #corner of the bounding box
            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))
            print ("Nilai x dan y dari B.Box ialah :", x," ", y)

            # update our list of bounding box coordinates, confidences, and
class IDs
            boxes.append([x, y, int(width), int(height)])
            confidences.append(float(confidence))
            classIDs.append(classID)
```

```
# apply non-maxima suppression to suppress weak, overlapping bounding boxes
idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
args["threshold"])

# ensure at least one detection exists
if len(idxs) > 0:
    # loop over the indexes
    for i in idxs.flatten():
        # extract the bounding box coordinates
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])

        # draw a bounding box rectangle and label on the image
        # color = [int(c) for c in COLORS[classIDs[i]]]
        cv2.rectangle(image, (x, y), (x + w, y + h), COLORS, 8)
        text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])
        print(text)
        cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,2,
COLORS, 8)
        # cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,0.2,
COLORS, 2)

# Font type
font = cv2.FONT_HERSHEY_SIMPLEX
# Font Coordinate
org = (20, 280)
# Font Size
fontScale = 10
# Font color with format (B,G,R)
color = (255, 0, 0)
# Font Thickness
thickness = 20

image = cv2.putText(image, '{} larva '.format(len(idxs)), org, font,
fontScale, color, thickness, cv2.LINE_AA)

# Write total of larva in the frame into larvaNumber.txt
with open("jumlahlarva.txt", "a") as myfile:
    myfile.write("Ada {} Larva yang terdeteksi\n".format(len(idxs)))

# show timing information on YOLO
print("YOLO took {:.6f} seconds".format(end - start))

# Save the output image
# cv2.imshow("Image.jpg", image)
cv2.imwrite("Image.jpg", image)
print ("ada", len(idxs), "Larva")
cv2.waitKey(0)
```