

Penentuan Komposisi dan Jenis Alga pada Tambak Udang menggunakan Metode YOLOv5 dan *Transfer Learning*

Yosua Wijaya^{1*}, Kenneth Angelo Tanjaya²

^{1,2}Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur
^{1*}s160419038@student.ubaya.ac.id, ²s160419144@student.ubaya.ac.id

Abstrak

Kualitas air adalah salah satu faktor pertimbangan yang penting ketika membudidayakan tambak udang. Fitoplankton atau mikroalga sebagai bioindikator kualitas air di tambak udang dapat meningkatkan pertumbuhan dan produktivitas udang di tambak. Namun, komposisi dan jenis fitoplankton di tambak udang bisa berubah secara cepat. Oleh karena itu, tidak menutup kemungkinan udang dapat terserang penyakit apabila komposisi alga dominan yang berdampak positif berubah. Penentuan komposisi dan jenis alga masih dilakukan secara tradisional dengan pemeriksaan mikroskopis menggunakan sampel air yang rumit dan memakan waktu. Penelitian ini dilakukan untuk mengidentifikasi komposisi dan jenis alga pada air tambak udang menggunakan sistem *computer vision*. Sistem dibuat dengan mengembangkan model *deep learning*, yaitu metode YOLOv5 sebagai model pendeteksi objek dan metode transfer learning, yaitu MobileNetV2, Xception, dan ResNet50, sebagai model klasifikasi jenis alga. Hasil penelitian menunjukkan model YOLOv5 mampu mendeteksi objek alga pada citra dengan nilai *precision* sebesar 92,2%, *recall* sebesar 91,7%, dan mAP50 sebesar 95,6%. Model ResNet50 memiliki performa terbaik dalam mengklasifikasikan jenis alga dengan nilai akurasi sebesar 97,7%, *precision* sebesar 98%, *recall* sebesar 98%, dan *f1-score* sebesar 98%.

Kata Kunci: tambak udang, alga, *computer vision*, YOLOv5, *transfer learning*.

Determination of Composition and Species of Algae in Shrimp Ponds using the YOLOv5 Method and Transfer Learning

Abstract

Water quality is one of the most important factor to consider when cultivating shrimp ponds. Phytoplankton or microalgae as bioindicators of water quality in shrimp ponds can increase the growth and productivity of shrimp in ponds. However, the composition and types of phytoplankton in shrimp ponds can change rapidly. Therefore, it is possible for the shrimp disease to happen if the composition of the dominant algae that has a positive impact changes. Determination of the composition and species of algae is still done traditionally by microscopic examination using water samples which is complicated and time consuming. This research was conducted to identify the composition and species of algae in shrimp pond water using a computer vision system. The system was created by developing deep learning models, such as the YOLOv5 method as an object detection model and transfer learning methods, namely MobileNetV2, Xception, and ResNet50, as an algae species classifier model. The results showed that the YOLOv5 model is able to detect algae objects in the image with 92,2% precision, 91,7% recall, and 95,6% mAP50. The ResNet50 model has the best performance in classifying algae with 97.7% accuracy, 98% precision, 98% recall, and a f1-score of 98%.

Keywords: shrimp ponds, algae, *computer vision*, YOLOv5, *transfer learning*.

I. PENDAHULUAN

Dalam membudidayakan tambak udang, ada beberapa hal yang perlu diperhatikan, seperti luas lahan, pakan, dan

kualitas air [1]. Kualitas air merupakan hal yang penting dan perlu diperhatikan karena sangat mempengaruhi produktivitas, pertumbuhan, dan kesehatan udang yang ada di

tambak. Penelitian yang dilakukan oleh Supriatna et al. [2] menunjukkan bahwa fitoplankton atau yang sering disebut sebagai mikroalga dapat digunakan sebagai bioindikator kualitas air tambak udang. Dapat disimpulkan bahwa terdapat hubungan antara keberadaan alga dengan kualitas air tambak udang.

Samadan [3] dalam penelitiannya menyebutkan bahwa jenis alga yang paling banyak ditemui di perairan tambak adalah fitoplankton. Hal tersebut terjadi karena faktor keberadaan nutrisi berupa nitrogen dan fosfor yang berasal dari pupuk organik dan sisa pakan. Keberadaan fitoplankton di tambak memberikan pengaruh yang signifikan terhadap pertumbuhan dan produktivitas udang yang ada di tambak tersebut. Jenis alga yang mampu memberikan dampak positif bagi udang adalah Diatom (*Bacillariophyta*) dan ganggang hijau (*Chlorophyceae*) karena dapat digunakan sebagai indikator kualitas air dan menjadi sumber makanan udang [4]. Sebaliknya, jenis alga yang memberikan dampak negatif adalah Dinoflagellata (*Dinophyta*) dan ganggang hijau biru (*Cyanophyta*) karena dapat menghasilkan senyawa beracun dan penyimpangan bau serta rasa pada udang.

Komposisi dan jenis fitoplankton yang ada di tambak udang mudah berubah dalam waktu yang cepat [4]. Perubahan tersebut dapat terjadi karena pengaruh kandungan nitrogen, fosfor, dan ketersediaan alkalinitas yang ada di perairan tambak. Oleh sebab itu, meskipun jumlah alga yang berdampak positif bersifat dominan, tidak menutup kemungkinan timbul penyakit yang dapat menyerang udang jika terjadi perubahan komposisi alga yang dominan [5].

Penentuan komposisi dan jenis alga di perairan tambak udang masih banyak dilakukan secara tradisional dengan mengamati sampel air di bawah mikroskop [6]. Jumlah dan jenis mikroalga pada sampel air tersebut kemudian dihitung dan diklasifikasikan secara manual oleh ahli. Namun, cara manual tersebut sangat membosankan dan menghabiskan waktu [7]. Selain itu, akurasi dalam melakukan perhitungan dan klasifikasi jenis mikroalga secara manual masih terbilang rendah dengan rata-rata di bawah 73,7% [8]. Berdasarkan fakta tersebut dibutuhkan sistem pendukung yang dapat mengetahui komposisi dan jenis alga yang tumbuh dan berkembang di perairan tambak udang agar produktivitas udang dapat tetap terjaga.

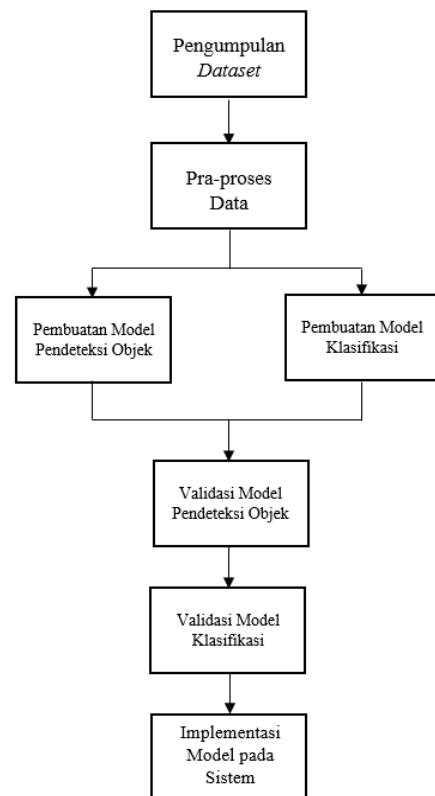
Rivas-Villar et al. [7] mengembangkan sistem pendeteksi dan pengklasifikasi jenis fitoplankton menggunakan metode *machine learning* terhadap citra sampel air tawar yang diambil menggunakan mikroskop. Objek alga pada citra sampel air tawar dideteksi dan dipisahkan menggunakan metode pemisahan *foreground-background*, diikuti algoritma Suzuki and Abe, lalu *colony merging*, dan perhitungan *similarity* tiap objek tetangga untuk mengetahui apakah kedua objek tersebut berbeda atau tidak. Proses identifikasi dan klasifikasi jenis fitoplankton menggunakan dua model *machine learning*, yaitu *Random Forest* (RF) dan *Support Vector Machine* (SVM). Algoritma pendeteksi dan pemisahan objek memberikan nilai *False Negative Rate* (FNR) sebesar 0,4%. Model SVM memberikan nilai *precision* sebesar 84,07% pada nilai *recall* 90% dalam mengklasifikasikan fitoplankton dari objek lainnya. Model RF memberikan nilai

akurasi sebesar 87,5% dalam mengklasifikasikan 4 jenis fitoplankton.

Correa et al. [9] menerapkan metode *deep learning* dengan menggunakan arsitektur *Convolutional Neural Network* (CNN) dalam mengklasifikasikan jenis mikroalga laut. Sebelum model CNN dilatih, *dataset* citra mikroalga dikenai proses augmentasi agar model memiliki kemampuan generalisasi yang baik. *Dataset* citra mikroalga dibagi menjadi 70% data latih dan 30% data validasi. Model CNN yang dibuat mampu memberikan akurasi sebesar 88,59%. Meskipun begitu, sistem-sistem tersebut tidak dibuat untuk mengenali alga yang ada di perairan tambak. Oleh karena itu, akan dibuat sistem *computer vision* untuk mengetahui komposisi dan jenis alga di perairan tambak udang. Sistem tersebut akan dibuat dengan mengembangkan model *deep learning* dengan menggunakan metode YOLOv5 sebagai model pendeteksi objek dan metode *transfer learning* sebagai model pengklasifikasi jenis alga. Dengan begitu, komposisi dan jenis alga akan lebih mudah diketahui tanpa menghabiskan waktu dan biaya.

II. METODOLOGI PENELITIAN

Penelitian ini terdiri dari beberapa tahapan, seperti yang ditunjukkan pada Gambar 1. Metode penelitian ini terdiri dari tahap pengumpulan *dataset*, pra-proses data, pembuatan model pendeteksi objek, pembuatan model klasifikasi, validasi model pendeteksi objek, validasi model klasifikasi, dan implementasi model pada sistem. Bagian ini akan menjelaskan proses-proses tersebut dengan lebih detail.



Gambar 1. Tahapan Proses Penelitian

A. Pengumpulan Dataset

Dataset yang digunakan pada penelitian ini adalah kumpulan citra alga mikroskopis yang diambil dari *website* Protist Information Server, yaitu <http://protist.i.hosei.ac.jp/> [10]. *Dataset* ini mengandung lima jenis alga yang umum tumbuh di tambak udang, yaitu *green algae*, *blue green algae*, *Chrysophyta*, *Dinoflagellata*, dan *Protozoa*. Total jumlah citra alga pada *dataset* ini adalah sebanyak 270 citra alga mikroskopis. Untuk jenis *green algae* terdiri atas 53 citra alga mikroskopis dengan detail jenis yang dipakai ditulis pada Tabel 1. Jenis *blue green algae* terdiri atas 33 citra alga mikroskopis dengan detail jenis yang dipakai ditulis pada Tabel 2. Lalu, untuk jenis *Chrysophyta* terdiri atas 73 citra alga mikroskopis dengan detail jenis yang dipakai ditulis pada Tabel 3. Kemudian, jenis *Dinoflagellata* terdiri atas 48 citra alga mikroskopis dengan detail jenis yang dipakai ditulis pada Tabel 4. Terakhir, jenis *Protozoa* terdiri atas 63 citra alga mikroskopis dengan detail jenis yang dipakai ditulis pada Tabel 5.

Tabel 1. Detail Gambar *Green Algae* berdasarkan Divisi

Nama Divisi	Jumlah
<i>Chlamydomonas</i>	10
<i>Chlorella</i>	6
<i>Chodatella</i>	10
<i>Scenedesmus</i>	27

Tabel 2. Detail Gambar *Blue Green Algae* berdasarkan Divisi

Nama Divisi	Jumlah
<i>Aphanocapsa</i>	7
<i>Chroococcus</i>	17
<i>Gomphosphaeria</i>	6
<i>Microcystis</i>	3

Tabel 3. Detail Gambar *Chrysophyta* berdasarkan Divisi

Nama Divisi	Jumlah
<i>Amphora</i>	9
<i>Cyclotella</i>	6
<i>Diploneis</i>	15
<i>Gyrosigma</i>	12
<i>Navicula</i>	7
<i>Ochromonas</i>	12
<i>Pinnularia</i>	12

Tabel 4. Detail Gambar *Dinoflagellata* berdasarkan Divisi

Nama Divisi	Jumlah
<i>Glenodinium</i>	11
<i>Gymnodinium</i>	13
<i>Gyrodinium</i>	3
<i>Peridinium</i>	21

Tabel 5. Detail Gambar *Protozoa* berdasarkan Divisi

Nama Divisi	Jumlah
<i>Chilomonas</i>	11
<i>Coleps</i>	20
<i>Colpoda</i>	9
<i>Euplotes</i>	16
<i>Strombidinopsis</i>	1
<i>Tetrahymena</i>	6

B. Pra-proses Data

Tiap gambar pada kelas alga akan dilakukan anotasi menggunakan *software* bernama LabelImg. LabelImg merupakan *library* Python yang berfungsi untuk memberikan anotasi gambar grafik dan memberikan label serta *bounding box* pada beberapa objek dalam kumpulan gambar [11]. Hasil yang dikeluarkan setelah memberikan *bounding box* pada tiap objek adalah *file* teks berisi koordinat dari setiap kotak per gambar. Total objek *bounding box* pada seluruh citra di *dataset* adalah sebanyak 1.942 gambar. Jumlah objek *bounding box* pada setiap kelas alga dapat dilihat pada Tabel 6.

Tabel 6. Jumlah Objek pada Setiap Kelas Alga

Kelas Alga	Jumlah
<i>Green Algae</i>	522
<i>Blue Green Algae</i>	673
<i>Chrysophyta</i>	279
<i>Dinoflagellata</i>	183
<i>Protozoa</i>	292

Setelah memberikan *bounding box* pada setiap gambar alga, *file* teks dan gambar akan dimasukkan ke dalam situs RoboFlow. RoboFlow merupakan situs yang membantu pengembang sistem untuk membangun sebuah aplikasi visi komputer dengan memberikan berbagai macam sarana yang berhubungan dengan topik tersebut, yaitu manajemen *dataset*. *Dataset* alga dengan *bounding box* akan dibagi oleh RoboFlow menjadi dua folder, yaitu folder untuk data latih dan folder untuk data validasi. Persentase isi untuk folder data secara berurutan sebesar 80% dan 20%. Kedua folder ini akan dipersiapkan untuk tahap selanjutnya.

C. Pembuatan Model Pendeteksi Objek

Pada penelitian ini diperlukan proses untuk mendeteksi objek yang terdapat pada citra karena sebuah citra mikroalga yang diamati menggunakan mikroskop dapat mengandung lebih dari satu objek alga. Oleh karena itu, pada penelitian kali ini akan digunakan metode YOLOv5 [12] sebagai model pendeteksi objek. Model ini dipilih karena memiliki performa yang cepat, akurat, dan komputasi yang lebih sederhana dibandingkan algoritma pendeteksi objek lainnya, seperti R-CNN (*Region-based Convolutional Neural Network*).

Model pendeteksi objek yang dibuat pada penelitian terdiri dari 2 macam, yaitu model YOLOv5 yang akan digunakan di sistem sekaligus sebagai tolak ukur performa dan model YOLOv5 yang digunakan untuk memotong objek-objek pada

citra alga mikroskopis agar dapat digunakan sebagai *dataset* untuk melatih model pengklasifikasi. Model pendeteksi objek yang pertama akan dilatih menggunakan perbandingan 80% data latih dan 20% citra sisanya akan digunakan untuk mengevaluasi performa model ini. Model pendeteksi objek yang kedua akan dilatih menggunakan 100% data latih citra alga mikroskopis karena model ini dibuat dengan tujuan agar setiap objek yang ada pada citra alga dapat dideteksi dan dipotong secara otomatis. Performa dari model ini akan dikalkulasi menggunakan data validasi yang berasal dari seluruh data latih. Parameter yang digunakan dalam melatih kedua model ini dapat dilihat pada Tabel 7.

Tabel 7. Parameter Model YOLOv5 yang Digunakan

Parameter	Value
Image size	320px
Batch size	16
Weight	yolov5s
Optimizer	SGD
Epoch	300

D. Pembuatan Model Klasifikasi

Pada penelitian ini, teknik yang akan digunakan adalah *transfer learning* dengan menggunakan *pre-trained* model MobileNetV2, Xception, dan ResNet50 yang disediakan oleh *library* Keras Tensorflow. Implementasi MobileNetV2, Xception, dan ResNet50 yang digunakan sama-sama membutuhkan parameter *input shape* dengan ukuran gambar 160 x 160 piksel yang memiliki 3 *channel*. Selain itu, parameter 'include_top' diberi nilai *false* karena menggunakan *output layer* yang dibuat sendiri sesuai dengan jumlah kelas alga. *Weights node* pada tiap *layer* menggunakan *weight* dari pelatihan data ImageNet yang merupakan standar dari klasifikasi gambar. ImageNet membandingkan gambar yang satu dengan gambar yang lain sehingga model memiliki kemampuan klasifikasi gambar dengan cocok. Untuk *optimizer* yang digunakan pada ketiga model ini adalah Adam *optimizer* dengan nilai *learning rate* sebesar 0,0005, kecuali model ResNet50 menggunakan *learning rate* sebesar 0,005. Parameter-parameter yang digunakan pada model klasifikasi jenis alga dapat dilihat pada Tabel 8.

Tabel 8. Parameter Model Klasifikasi yang Digunakan

Parameter	Value
Input Shape	(160, 160, 3)
Include Top	<i>false</i>
Optimizer	Adam
Learning Rate	0.005
Weights	ImageNet
Epoch	100

Tabel 9. Layer Model MobileNetV2 yang Dimodifikasi

Layer	Value	Fungsi Aktivasi
Input	(160, 160, 3)	-
Input (MobileNetV2)	(160, 160, 3)	-
Preprocess Input	Input	-
Global Average Pooling	-	-
Dense	(128)	ReLU
Dropout	0.2	-
Dense	(64)	ReLU
Dense	(5)	Softmax

Tabel 9 di atas menunjukkan detail *layer* pada model modifikasi MobileNetV2. Pengembangan MobileNetV2 dilakukan dengan menambahkan *input layer* dengan ukuran yang sama dengan ukuran pembacaan gambar dari *dataset* yaitu 160x160x3 (3 *channel*/RGB) dan *top layer* model MobileNetV2 tidak dipakai. Setelah *input layer*, terdapat *Preprocess Input*, yaitu *layer* yang bertugas untuk melakukan proses skala ulang piksel pada setiap gambar agar rentangan nilai berubah dari 0 hingga 255 menjadi -1 hingga 1 sehingga cocok dengan parameter input pada MobileNetV2. Selanjutnya, ada *GlobalAveragePooling2D layer* yang berfungsi untuk mengganti *fully-connected layers* menjadi *average pooling*. Selanjutnya, terdapat *dense layer* dengan 128 *node* menggunakan fungsi aktivasi *Rectified Linear Unit* (ReLU). Kemudian, terdapat *Dropout layer* dengan nilai 0,2 yang bertugas untuk menyisihkan beberapa persen *node layer* secara acak sesuai dengan perhitungan yang ada pada setiap *epoch*. Berikutnya, terdapat *dense layer* lagi dengan 64 *node* menggunakan fungsi aktivasi ReLu. Terakhir, *Dense layer* dengan fungsi aktivasi *Softmax* memiliki 5 *node* yang berfungsi sebagai *output layer*. Hasil pada *layer* ini adalah probabilitas deteksi pada setiap kelas.

Tabel 10. Layer Model Xception yang Dimodifikasi

Layer	Value	Fungsi Aktivasi
Input	(160, 160, 3)	-
Input (Xception)	(160, 160, 3)	-
Preprocess Input	Input	-
Global Average Pooling	-	-
Dropout	0.2	-
Dense	(128)	ReLU
Dropout	0.2	-
Dense	(64)	ReLU
Dense	(5)	Softmax

Tabel 10 di atas menunjukkan detail *layer* pada model modifikasi Xception. Pengembangan model Xception pada empat *layer* pertama sama seperti yang dilakukan pada model MobileNetV2. Setelah *GlobalAveragePooling2D layer*, terdapat *Dropout layer* sebesar 0,2 yang bertugas untuk menyisihkan beberapa persen *node layer* secara acak sesuai dengan perhitungan yang ada pada setiap *epoch*. Kemudian, terdapat *dense layer* dengan 128 *node* menggunakan fungsi aktivasi ReLu. Selanjutnya, terdapat *Dropout layer* lagi

dengan nilai 0,2. Berikutnya, terdapat *dense layer* lagi dengan 64 *node* menggunakan fungsi aktivasi ReLu. Terakhir, *Dense layer* dengan fungsi aktivasi *Softmax* memiliki 5 *node* yang berfungsi sebagai *output layer*.

Tabel 11. Layer Model ResNet50 yang Dimodifikasi

Layer	Value	Fungsi Aktivasi
Input	(160, 160, 3)	-
Input (ResNet50)	(160, 160, 3)	-
Preprocess Input	Input	-
Global Average Pooling	-	-
Dropout	0.4	-
Dense	(5)	Softmax

Tabel 11 di atas menunjukkan detail *layer* pada model modifikasi ResNet50. Pengembangan model ResNet50 pada empat *layer* pertama juga masih sama seperti yang dilakukan pada model MobileNetV2. Setelah *GlobalAveragePooling2D layer*, terdapat *Dropout layer* dengan nilai 0,4 yang bertugas untuk menyisihkan beberapa persen *node* *layer* secara acak sesuai dengan perhitungan yang ada pada setiap *epoch*. Terakhir, *Dense layer* dengan fungsi aktivasi *Softmax* memiliki 5 *node* yang berfungsi sebagai *output layer*.

E. Validasi Model Pendeteksi Objek

Model pendeteksi objek yang sudah dilatih akan divalidasi dengan melihat performanya terhadap *validation dataset*. Performa model dapat diketahui dengan melihat nilai yang terdapat pada *evaluation metrics*. Pada kasus deteksi objek, *evaluation metrics* digunakan untuk mengukur seberapa dekat *bounding box* hasil prediksi dengan *bounding box ground truth* [13]. *Evaluation metrics* yang akan digunakan untuk mengukur performa model pendeteksi objek adalah *Precision*, *Recall*, dan *Mean Average Precision* (mAP).

mAP merupakan metrik yang digunakan untuk mengukur akurasi dari model pendeteksi objek untuk semua label yang digunakan [13]. Nilai mAP didapatkan dari rata-rata nilai *Average Precision* (AP) terhadap semua label objek yang dievaluasi (N). AP sendiri merupakan metrik yang didapatkan dengan menghitung area di bawah kurva *precision x recall* untuk setiap label secara independen. Nilai AP didapatkan dengan menghitung terlebih dahulu nilai rata-rata dari 11 poin *precision* terinterpolasi ($P_{interp}(R)$) yang didapatkan dengan menginterpolasi nilai *precision* maksimum ($\max P(\tilde{R})$) terhadap nilai *recall*-nya (\tilde{R}) pada saat nilai *recall* tersebut lebih besar atau sama dengan nilai *recall* saat ini (R), seperti yang ditunjukkan pada persamaan (1). Nilai *precision* diinterpolasi melewati 11 nilai *recall* (R) dengan jarak yang sama ($[0, 0.1, 0.2, \dots, 1]$). Perlu diketahui bahwa nilai AP dikaitkan dengan nilai *threshold Intersection Over Union* (*IoU*) yang merupakan batas persentase seberapa besar nilai daerah *bounding box* prediksi yang beririsan dengan *bounding box ground truth*. Oleh karena itu, hanya area objek dengan nilai $IoU \geq \text{threshold}$ saja yang dianggap terdeteksi dengan benar (termasuk *true positive*). Rumus untuk menghitung AP

dapat dilihat pada persamaan (2), sedangkan rumus untuk menghitung mAP dapat dilihat pada persamaan (3).

$$AP_{interp}(R) = \max_{\tilde{R}: \tilde{R} \geq R} P(\tilde{R}) \quad (1)$$

$$AP = \frac{1}{11} \sum_{R \in \{0, 0.1, \dots, 0.9, 1\}} P_{interp}(R) \quad (2)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

F. Validasi Model Klasifikasi

Setelah selesai melakukan pelatihan dan pengujian data, performa model klasifikasi dihitung dengan menghitung skor akurasi, *precision*, *recall*, dan *f1-score*. Model dengan performa terbaik akan didapatkan dengan melihat keempat aspek tersebut. Rumus untuk menghitung *accuracy model* dapat dilihat pada persamaan (4), *precision* pada persamaan (5), *recall* pada persamaan (6), dan *f1-score* pada persamaan (7). Perlu diketahui bahwa sebutan *True Positive* (TP) merupakan data kelas positif yang hasil prediksinya diklasifikasikan sebagai positif. *True Negative* (TN) merupakan data kelas negatif yang hasil prediksinya diklasifikasikan sebagai negatif. *False Positive* (FP) merupakan data kelas negatif yang hasil prediksinya diklasifikasikan sebagai positif. *False Negative* (FN) merupakan data kelas positif yang hasil prediksinya diklasifikasikan sebagai negatif.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 \text{ Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

G. Implementasi Model pada Sistem

Model pendeteksi objek dan model klasifikasi jenis alga yang sudah dilatih selanjutnya akan diimplementasikan ke sistem *computer vision*. Dari sisi *backend*, model pendeteksi objek akan dimuat menggunakan *weight* terbaik berformat '.pt' yang didapatkan dari hasil pelatihan model YOLOv5, sedangkan model klasifikasi jenis alga dimuat dari model hasil pelatihan yang telah disimpan sebelumnya menggunakan format '.pb'. Kedua model tersebut disimpan pada folder bernama 'saved_model'. Folder tersebut kemudian dipindah ke dalam folder *models* pada aplikasi *backend*. Selanjutnya, model pendeteksi objek dimuat dengan *library* Pytorch, sedangkan model klasifikasi akan dimuat dengan *library* Tensorflow, pada *framework* FastAPI.

FastAPI merupakan *framework* Python yang dapat digunakan untuk menjalankan server HTTP. Proses dilanjutkan dengan melakukan pembuatan suatu *endpoint* untuk metode POST untuk mendapatkan satu citra dari HTTP. Setelah itu, dilakukan sebuah *pre-processing* citra yang telah dikirimkan melalui HTTP, sehingga memiliki format *numpy*. *Numpy array* citra tersebut kemudian dideteksi dan dipotong setiap objek alga yang ditemui menggunakan model YOLOv5 dan diberi anotasi *bounding box*. Setiap objek yang sudah dipotong kemudian dikenai proses *dimension expansion* untuk dapat menjadikan citra sebagai parameter untuk dapat diprediksi jenisnya menggunakan model klasifikasi. Setelah melakukan prediksi, dilakukan perhitungan jumlah alga beserta jenisnya. Jumlah alga tersebut akan disusun bentuknya menjadi *dictionary* dengan *key* nya adalah jenis alga, dan *value* nya adalah jumlah dari alga jenis terkait. Selain itu, terdapat juga *value* citra alga mikroskopis yang telah diberi anotasi *bounding box* pada tiap objek yang dideteksi dan diubah formatnya menjadi *base64 image*. Pesan *return* HTTP yang akan dikirim adalah *dictionary* tersebut.

Dari sisi *frontend*, *framework* yang digunakan adalah React yang merupakan salah satu *frontend framework* berbasis Javascript. Pada *frontend*, pengguna dapat mengambil citra dengan mikroskop atau melakukan *upload* citra dari *file* lokal. Setelah citra diunggah, pengguna dapat menekan tombol *submit* untuk mengirimkan citra tersebut ke aplikasi *backend*. Kemudian, sistem akan menampilkan hasil prediksi yang telah dikirim dari *backend*.

III. HASIL DAN PEMBAHASAN

Hasil Pelatihan model YOLOv5 sebagai model pendeteksi objek alga mikroskopis yang dievaluasi menggunakan 54 citra alga dari *validation dataset* memberikan performa yang cukup memuaskan, seperti yang ditunjukkan pada Tabel 12. Pelatihan model ini membutuhkan waktu selama 16 menit 14 detik menggunakan *Graphics Processing Unit* (GPU) yang disediakan oleh Google Colab. Nilai mAP yang didapatkan pada IoU dengan *threshold* 50% beserta *precision* dan *recall* untuk semua jenis alga memberikan hasil yang baik dengan nilai di atas 90%. Hal tersebut menunjukkan bahwa model YOLOv5 yang dilatih mampu mendeteksi objek alga mikroskopis pada citra dengan cukup baik.

Tabel 12. Hasil Evaluasi Model YOLOv5

Metrik	Nilai (%)
<i>Precision</i>	92,2
<i>Recall</i>	91,7
mAP50	95,6

Pelatihan model YOLOv5 yang kedua, yaitu model pendeteksi objek yang digunakan untuk memotong objek alga mikroskopis pada citra agar dapat digunakan sebagai *dataset* model pengklasifikasi jenis alga membutuhkan waktu selama 26 menit 23 detik menggunakan GPU yang disediakan oleh Google Colab. Berdasarkan validasi yang juga dilakukan menggunakan data latih, diketahui bahwa model ini

memberikan nilai *precision* sebesar 98,1%, nilai *recall* sebesar 97,7%, dan mAP pada IoU 50% sebesar 98,9%. Model ini mampu mendeteksi dan memotong objek-objek alga mikroskopis pada citra dengan baik, meskipun jumlah objek alga yang dipotong berbeda sedikit dengan jumlah objek alga sebenarnya di data latih.

Pelatihan model pra-latih CNN yang dilakukan terhadap *dataset* citra alga mikroskopis melibatkan beberapa arsitektur, yaitu MobileNetV2, Xception, dan ResNet50. Ketiga model yang dilatih ini akan dibandingkan performanya dalam mengklasifikasikan jenis mikroalga. Pelatihan model ini dilakukan dengan memanfaatkan metode *transfer learning* yang mana ketiga model yang telah ditambah *layer*-nya tersebut menggunakan *weight* dari pelatihan *dataset* ImageNet. Selama proses pelatihan, *layer-layer* yang sudah memiliki *weight* tidak dilatih lagi, sedangkan *layer* tambahan yang diberikan dilatih menggunakan *dataset* citra alga mikroskopis. Berdasarkan beberapa percobaan yang telah dilakukan, didapatkan beberapa metrik berupa akurasi, *precision*, *recall*, dan *f1-score* terhadap *validation dataset*. Nilai-nilai metrik untuk setiap model pra-latih CNN untuk mengklasifikasikan jenis alga dapat dilihat pada Tabel 13.

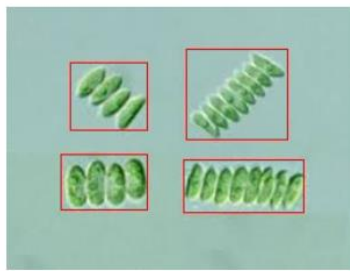
Tabel 13. Hasil Evaluasi Ketiga Model Klasifikasi

Model	Akurasi (%)	<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F1-score</i> (%)
MobileNetV2	97,2	97	97	97
Xception	96,1	96	96	96
ResNet50	97,7	98	98	98

Dalam proses pelatihan model pengklasifikasi jenis alga, ketiga model yang dibandingkan ini dilatih dengan iterasi sebanyak 100 *epoch*. Selain itu, selama pelatihan juga diterapkan metode *early stopping* dengan mengamati *validation accuracy* pada setiap *epoch*-nya dengan nilai *patience* sebanyak 25 *epoch*. Saat melatih model MobileNetV2, iterasi berhenti pada *epoch* ke-61 karena tidak terjadi kenaikan *validation accuracy* sejak *epoch* ke-36. Model MobileNetV2 mampu memberikan *validation accuracy* sebesar 97,2% dengan nilai *loss* sebesar 0,073. Pada model Xception, pelatihan model berhenti pada *epoch* ke-59 karena tidak terjadi kenaikan *validation accuracy* sejak *epoch* ke-34. Setelah dilatih, model Xception mampu memberikan *validation accuracy* sebesar 96,1% dengan nilai *loss* sebesar 0,187. Pada model ResNet50, pelatihan model berhasil berjalan penuh selama 100 *epoch* dan tidak terjadi penurunan *validation accuracy* yang signifikan. Model ini mampu memberikan *validation accuracy* sebesar 97,7% dengan nilai *loss* sebesar 0,079.

Berdasarkan hasil metrik masing-masing model pengklasifikasi jenis alga yang sudah disebutkan sebelumnya, diketahui bahwa model ResNet50 memberikan performa yang paling baik dibandingkan model MobileNetV2 maupun Xception. Dari segi akurasi, *precision*, *recall*, dan *f1-score*, model ResNet50 lebih unggul dibandingkan kedua model lainnya. Setelah model YOLOv5 dan ResNet50 diimplementasikan pada sistem maka input citra alga

mikroskopis dapat ditentukan komposisi dan jenisnya secara otomatis. Gambar 2 menunjukkan contoh *output* dari sistem.



Green Algae 4
Gambar 2. Contoh *Output* Sistem

IV. KESIMPULAN

Penelitian ini berhasil menghasilkan sebuah sistem *computer vision* yang digunakan untuk mengetahui komposisi dan jenis alga yang tumbuh dan berkembang di perairan tambak udang berdasarkan citra alga mikroskopis. Implementasi model pendeteksi objek serta model pengklasifikasi jenis alga juga dapat berjalan dengan baik pada sistem yang dibuat. Sistem ini menerima input berupa citra alga mikroskopis dengan memberikan *output* berupa jumlah, jenis, dan anotasi *bounding box* pada citra alga yang telah diinputkan sebelumnya.

Pengumpulan *dataset* dilakukan dengan mengambil citra alga mikroskopis yang terdapat pada *website* <http://protist.i.hosei.ac.jp/>. *Dataset* ini terdiri atas 270 citra alga mikroskopis yang mengandung lima jenis alga yang umum tumbuh di tambak udang, yaitu *green algae*, *blue green algae*, *Chrysophyta*, *Dinoflagellata*, dan *Protozoa*.

Model pendeteksi objek menggunakan metode YOLOv5 berhasil digunakan untuk mendeteksi objek alga pada citra alga mikroskopis dengan nilai *precision* sebesar 92,2%, *recall* sebesar 91,7%, dan *mAP50* sebesar 95,6%. Berdasarkan pelatihan ketiga model pra-latih CNN dengan memanfaatkan metode *transfer learning*, didapati hasil bahwa model ResNet50 memberikan performa paling baik dibandingkan dua model yang lain, yaitu MobileNetV2 dan Xception. Model ResNet50 mampu mengklasifikasikan jenis alga mikroskopis dengan akurasi sebesar 97,7%, *recall* sebesar 98%, *precision* sebesar 98%, dan *f1-score* sebesar 98%.

Berdasarkan hasil temuan pada penelitian ini maka saran untuk penelitian ke depannya adalah menggunakan *dataset* citra alga mikroskopis yang lebih banyak dengan jumlah citra maupun objeknya yang seimbang antar jenisnya agar performa model pendeteksi objek dan juga model pengklasifikasi jenis alga dapat meningkat. Selain itu, disarankan juga untuk mencoba versi YOLO yang lebih baru, yaitu YOLOv7 yang mungkin dapat memberikan performa yang lebih baik dalam mendeteksi objek alga.

REFERENSI

- [1] Budi, N. A. R. S. (2019). *Analisis Faktor-Faktor yang Mempengaruhi Produksi Tambak Udang di Kabupaten Kulon Progo* (Doctoral dissertation, STIE YKPN).
- [2] Supriatna, Mahmudi, M., Musa, M., Martinah, A., Marsoedi. (2018). Hubungan Parameter Kualitas Air dengan Fitoplankton pada Tambak Intensif Udang Vanamei (*Litopenaeus vannamei*) di Banyuwangi Jawa Timur. <https://isfm.faperika.unri.ac.id/wp-content/uploads/2019/08/ISFM7th-min-1.pdf#page=92>
- [3] Samadan, G. M., Supyan, S., Andriani, R., & Juharni, J. (2020). Kelimpahan plankton pada budidaya udang vaname (*Litopenaeus vannamei*) dengan kepadatan berbeda di tambak lahan pasir. *Jurnal Ilmu Kelautan Kepulauan*, 3(2). <http://dx.doi.org/10.33387/jikk.v3i2.2588>
- [4] Supono, S. (2018). *Manajemen Kualitas Air untuk Budidaya Udang*. Bandar Lampung: Anugrah Utama Raharja.
- [5] Yang, W., Zheng, Z., Lu, K., Zheng, C., Du, Y., Wang, J., & Zhu, J. (2020). Manipulating the phytoplankton community has the potential to create a stable bacterioplankton community in a shrimp rearing environment. *Aquaculture*, 520, 734789. <https://doi.org/10.1016/j.aquaculture.2019.734789>
- [6] Barsanti, L., Birindelli, L., & Gualtieri, P. (2021). Water monitoring by means of digital microscopy identification and classification of microalgae. *Environmental Science: Processes & Impacts*, 23(9), 1443-14457. <https://doi.org/10.1039/D1EM00258A>
- [7] Rivas-Villar, D., Rouco, J., Carballeira, R., Penedo, M. G., & Novo, J. (2021). Fully automatic detection and classification of phytoplankton specimens in digital microscopy images. *Computer Methods and Programs in Biomedicine*, 200, 105923. <https://doi.org/10.1016/j.cmpb.2020.105923>
- [8] Kelly, M. G., Bayer, M. M., Hürlimann, J., & Telford, R. J. (2002). Human error and quality assurance in diatom analysis. *Automatic diatom identification*, pp. 75-91. https://doi.org/10.1142/9789812777867_0005
- [9] Correa, I., Drews, P., Botelho, S., de Souza, M. S., & Tavano, V. M. (2017). Deep learning for microalgae classification. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 20-25. <https://doi.org/10.1109/ICMLA.2017.0-183>
- [10] Japan Science and Technology Corporation (2017, January 6), <http://protist.i.hosei.ac.jp/>
- [11] Lin, Tzutha (2017). LabelImg. Pypi.org. <https://pypi.org/project/labelImg/1.4.0/>.
- [12] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, ... & Mrinal Jain. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). *Zenodo*. <https://doi.org/10.5281/zenodo.7347926>
- [13] Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & Da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit.

Electronics, 10(3), 279.
<http://dx.doi.org/10.3390/electronics1003027>
