

# Deteksi Jamur Beracun dan Tidak Beracun Menggunakan CNN dan YOLO

Nurul Fajria<sup>1</sup>, Lifwarda<sup>2\*</sup>, Ramiati<sup>3</sup>, dan Yulindon<sup>4</sup>  
<sup>1,2,3,4</sup>Jurusan Teknik Elektro, Politeknik Negeri Padang, Jl. Limau Manih Padang, 25164, Indonesia  
\*Corresponding Author, email: [lifwarda@pnp.ac.id](mailto:lifwarda@pnp.ac.id)

**Abstrak** — Di Indonesia yang memiliki iklim tropis dan ada banyak kayu yang sangat cocok untuk pertumbuhan jamur. Namun, beberapa jamur ini ada yang beracun dan harus dihindari. Dengan memeriksa ciri-ciri morfologi jamur, seperti bentuk tudung jamur, warna, bau, dan ciri lainnya dimungkinkan untuk mengidentifikasi antara jamur beracun dan tidak beracun. Namun, beberapa jamur beracun memiliki sifat morfologis yang sama dengan jamur tidak beracun, sehingga sulit bagi kita untuk membedakannya jika dilihat dengan mata biasa. Sebagai solusi dari permasalahan tersebut, dibutuhkan pembelajaran mesin untuk mengidentifikasi jamur yang beracun dan tidak. Seperti pada penelitian kali ini digunakan Convolutional Neural Network (CNN) dan You Only Look Once (YOLO) sebagai metode untuk identifikasi. Tahapan pada penelitian ini antara lain pengumpulan dataset, lalu pemodelan dan data latih, dan proses yang terakhir yaitu deployment. Metode CNN mendapatkan nilai akurasi sebesar 55% dan berhasil mengidentifikasi 5 gambar benar dan 5 salah dari 10 gambar yang diberikan. Sedangkan menggunakan YOLOv5 mendapatkan nilai akurasi yang besar yaitu 91% dan berhasil mengidentifikasi 9 gambar benar dan 1 salah dari 10 gambar yang diberikan. Dari perbandingan 2 metode tersebut didapatkan bahwa untuk mendeteksi jamur beracun dan tidak beracun itu lebih bagus dengan menggunakan metode YOLO daripada metode CNN.

**Kata Kunci:** Convolutional Neural Network, Jamur Beracun, Jamur Tidak beracun, Training, YOLO

*Abstract— In Indonesia, which has a tropical climate there is a lot of wood which is very suitable for mushroom growth. However, some of these mushrooms are poisonous and should be avoided. By examining the morphological characteristics of the mushroom, such as the shape of the mushroom cap, color, smell, and other characteristics it is possible to identify between poisonous and non-poisonous mushrooms. However, some poisonous mushrooms have the same morphological characteristics as non-poisonous mushrooms, making it difficult for us to differentiate them when seen with the naked eye. As a solution to this problem, machine learning is needed to identify which mushrooms are poisonous and which are not. As in this research, Convolutional Neural Network (CNN) and You Only Look Once (YOLO) were used as methods for identification. The stages in this research include collecting datasets, modeling and training data, and the final process, namely deployment. CNN Method got an accuracy value of 55% and succeeded in identifying 5 correct and 5 incorrect images from the 10 images given. Meanwhile, using YOLOv5, we got a high accuracy value of 91% and succeeded in identifying 9 correct and 1 incorrect image out of the 10 images provided. From a comparison of the 2 methods, it was found that detecting poisonous and non-toxic mushrooms was better using the YOLO method than the CNN method.*

**Keywords:** Convolutional Neural Network, Poisonous Mushrooms, Edible Mushrooms, Training, YOLO

© 2024 Elektron Jurnal Ilmiah

## I. PENDAHULUAN

Di Indonesia yang memiliki iklim tropis dan ada banyak kayu yang sangat cocok untuk pertumbuhan jamur, jamur merupakan salah satu tumbuhan liar yang banyak tersebar di alam bebas. Namun, beberapa jamur ini ada yang beracun dan harus dihindari. Jamur yang tersebar di dunia saat ini diperkirakan sedikitnya ada 45.000 jenis dan 2.000 jenis diantaranya tidak beracun. Ada banyak spesies jamur mematikan yang ada di dunia. Ada sekitar 70 hingga 80 spesies yang paling beracun dan mematikan yang bisa berakibat fatal jika dikonsumsi [1]. Dengan memeriksa ciri-ciri morfologi jamur, seperti bentuk payung atau tudung jamur, warna, tekstur, bau, dan ciri lainnya dimungkinkan untuk mengidentifikasi antara jamur beracun dan tidak beracun. Namun, beberapa jamur beracun memiliki sifat morfologis yang sama dengan jamur tidak beracun, sehingga sulit bagi kita untuk membedakannya jika dilihat dengan mata biasa.

Pembelajaran mesin diperlukan untuk membedakan antara jamur beracun dan tidak pada masalah ini. Untuk mengidentifikasi dengan hasil yang lebih baik dan lebih akurat diperlukan pendekatan algoritma pembelajaran mesin khususnya *deep learning*, dengan memanfaatkan kumpulan data berupa gambar.

Beberapa penelitian sebelumnya yang membahas tentang teknik untuk membedakan antara jenis jamur beracun dan tidak beracun. Seperti penelitian yang telah dilakukan oleh Nuke L. Chusna, dkk [2], mereka telah melakukan klasifikasi terhadap jamur dengan menggunakan Algoritma *Multiclass Support Vector Machin* (SVM). Aplikasi untuk melakukan klasifikasi terhadap jamur dibuat menggunakan Matlab dengan tingkat akurasi dari klasifikasi yang dihasilkan sebesar 83%. Kemudian penelitian oleh Khoiru Nurfitri, dkk [3], melakukan klasifikasi jenis jamur yang beracun dan tidak beracun dengan menerapkan algoritma PCA dan KNN. Aplikasi ini dibuat menggunakan matlab

dengan nilai akurasi 92% dari 25 data uji. Penelitian lain oleh Yohannes, dkk [4], yang melakukan klasifikasi jenis jamur menggunakan SVM dengan fitur HSV dan HOG. Pada penelitian itu didapatkan nilai akurasinya sebesar 82,69%. Penelitian oleh Elok Iedfitra Haksoro dan Abas Setiawan [5], tentang pengenalan jamur yang dapat dikonsumsi menggunakan metode *transfer learning* pada *convolutional neural network*. Akurasi yang didapatkan sebesar 92,19%. Kemudian penelitian lain dilakukan oleh Sofika Enggari, dkk [6], penelitiannya tentang peningkatan *digital image processing* dalam mendeskripsikan tumbuhan jamur dengan segmentasi warna, deteksi tepi dan kontur. Pada penelitian ini menggunakan matlab untuk membuat aplikasinya. Lima penelitian tersebut tidak mengidentifikasi objek secara real time, perlu memfoto objek dulu baru upload ke aplikasi dan keluarlah hasil identifikasi.

Pemilihan penggunaan metode *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO) dikarenakan kelebihan dari metode tersebut. Kelebihan dari metode CNN adalah metode ini dapat secara otomatis mengekstrak detail kecil dari setiap gambar. Selain itu, metode CNN lebih efektif daripada pendekatan jaringan saraf lainnya, khususnya dalam hal kompleksitas dan memori. Pendekatan CNN memiliki kelemahan yaitu membutuhkan banyak data *training*, proses pelatihan yang memakan waktu, dan *overfitting*. Pendekatan YOLO memiliki manfaat kecepatan pemrosesan yang cepat dan kesederhanaan dalam beradaptasi dengan foto *input* baru. Meskipun YOLO cepat mengenali item dalam foto, melokalisasi beberapa hal dengan tepat, terutama yang kecil, membutuhkan banyak usaha.

Dari beberapa penelitian yang telah disebutkan sebelumnya, maka pada penelitian kali ini digunakan CNN dan YOLO sebagai metode untuk mengenali jenis jamur yang beracun dan tidak beracun. Yang membedakan metode ini dari penelitian yang sudah ada sebelumnya adalah aplikasi ini digunakan secara real time sehingga pengguna bisa langsung mengarahkan *device* ke objek lalu muncul hasil identifikasinya.

Perbedaan antara jamur beracun dan tidak beracun memiliki konsekuensi serius terhadap keamanan pangan dan kesehatan manusia, yang memperkuat urgensi dari penelitian tentang identifikasi yang akurat. Penelitian ini memiliki tujuan dalam meningkatkan efisiensi dan akurasi proses identifikasi jamur. Metode ini menggunakan teknologi *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO) untuk mengidentifikasi jamur secara real-time berdasarkan citra visual. Implementasi teknologi ini tidak hanya mempercepat proses identifikasi, tetapi juga dapat digunakan sebagai alat bantu dalam mendukung pengawasan keamanan pangan dan pencegahan keracunan jamur[7].

## II. METODE

### A. Dataset

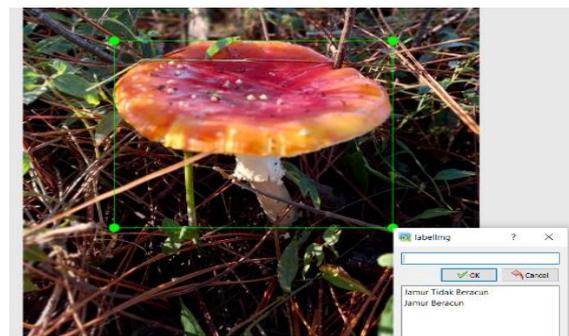
Data yang penulis kumpulkan pada tahap ini adalah citra jamur beracun dan tidak beracun yang di dapat dari *website* Kaggle.com, sebuah platform yang menyediakan berbagai dataset untuk keperluan penelitian dan pengembangan.. Sebanyak 1.234 citra jamur dengan berbagai jenis jamur digunakan sebagai dataset pada penelitian ini. Jamur yang tidak beracun adalah jamur tiram, jamur kuping, jamur kancing, dan jamur enoki yang total gambarnya ialah 595 gambar. Lalu jenis jamur yang beracun adalah jamur *death cap*, jamur *conocybe filaris*, jamur *webcaps*, jamur *autumn skullcap*, jamur *destroying angels*, jamur *deadly dapperling* dan *magic mushroom* yang total gambarnya ialah 639 gambar.

Sebelum dataset dapat digunakan untuk pengujian, setiap gambar harus melalui proses pelabelan. Metode YOLO (*You Only Look Once*) digunakan untuk pelabelan ini, dimana setiap gambar diberi label secara manual satu per satu, menghasilkan file .txt yang menyertai setiap gambar. Pelabelan dilakukan untuk memastikan bahwa model YOLO dapat mengenali dan mengklasifikasikan jamur dengan benar. Proses ini melibatkan penandaan posisi dan jenis jamur dalam setiap gambar.

### B. Labelling Dataset

Labelling atau Anotasi, sering dikenal sebagai pelabelan, adalah proses pemberian detail tentang banyak hal yang diperlukan seperti jamur. Untuk menentukan koordinat kotak pembatas *ground-truth* yang akan dibandingkan dengan kotak pembatas yang diprediksi, koleksi gambar yang terkumpul diberi label satu per satu. Nilai *Intersection over Union* (IoU) akan ditentukan dengan membandingkan dua kotak *ground*[8]. Proses labelling ini dilakukan pada *dataset* yang menggunakan metode YOLO. Untuk metode CNN tidak perlu melakukan labelling pada gambar jamurnya.

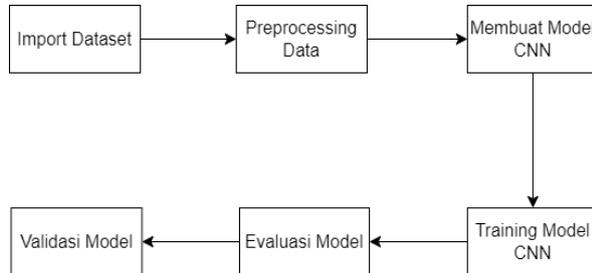
Proses memberikan label pada gambar yang dibutuhkan digunakan aplikasi LabelImg. Setelah memberi label nantinya akan disimpan kedalam file .xml/.txt dengan format YOLO. Proses untuk membuat anotasi gambar jamur yang dilakukan pada LabelImg terdapat pada Gambar 1.



Gambar 1. Labelling dataset

### C. Pemodelan dan Training Metode CNN

*Dataset* yang sudah dikumpulkan sebelumnya akan dibuat dalam 2 kelas yaitu jamur beracun dan jamur tidak beracun. Setelah itu dilakukan proses *training* untuk membuat sistem identifikasi jamur beracun dan tidak beracun dengan menggunakan Google Colaboratory. Proses yang dilalui saat pemodelan terdapat pada Gambar 2.



Gambar 2. Blok diagram pemodelan CNN

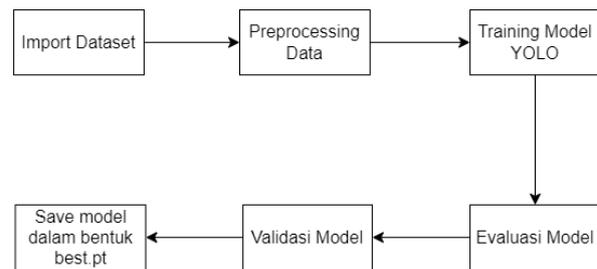
1. *Preprocessing* data: Pada tahap *preprocessing* data dilakukan seperti *resizing* gambar ke ukuran yang seragam, normalisasi nilai piksel, serta pemisahan data menjadi train set dan validation set pada *dataset* yang telah kita kumpulkan tadi.
2. Bangun model CNN: Gunakan framework deep learning seperti TensorFlow atau PyTorch untuk membangun arsitektur model CNN dengan menggunakan lapisan-lapisan konvolusi, *pooling*, *dropout*, dan lain-lain yang terbaik sehingga model tersebut mendapatkan akurasi yang baik.
3. *Training* model: Latih model CNN menggunakan train set yang telah dipersiapkan sebelumnya. Selama pelatihan (*training*), pastikan untuk memantau performa model di validation set agar dapat melakukan evaluasi secara berkala. *Training* model dilakukan dengan mengatur *epoch* yang baik agar menghasilkan nilai akurasi yang tinggi.
4. Evaluasi Model : Setelah melakukan *training* selama beberapa *epoch* yang diberikan atau hingga mencapai tingkat akurasi optimal. Selanjutnya dilakukan evaluasi pada performa model dengan memperhatikan matrik akurasi (*accuracy*), presisi (*precision*), *recall*, *f1-score*, dll.
5. Validasi Model : Langkah terakhir adalah menguji model dengan menggunakan gambar jamur yang belum pernah dilihat sebelumnya untuk memeriksa keakuratan dan kinerja keseluruhan dari model tersebut.

Tahapan pertama yang harus dilakukan pada identifikasi jamur beracun dan tidak beracun menggunakan metode CNN yaitu pembacaan dataset citra. Selanjutnya pada proses klasifikasi pertama akan dilakukan split dataset dimana akan dibagi antara data training dan validasinya dan pada tahap ini juga dilakukan perubahan resolusi seluruh gambar menjadi 224x224 pixel. Setelah dibagi didapatkan 988 gambar untuk data training dan 246 data validasi. Selanjutnya

dilakukan pembuatan layer-layer pada modelnya, menggunakan beberapa layer seperti Convolution Layer, Maxpolling layer, dropout layer, flatten layer, dan dense layer.

Selanjutnya melakukan training model menggunakan Google Colaboratory dengan mengatur epoch dan steps per epoch. Epoch digunakan untuk mengukur seberapa banyak kali model dilatih dengan dataset yang sama, steps per epoch untuk mengatur banyaknya batch yang akan dieksekusi pada setiap epoch. Lama waktu yang dibutuhkan untuk training tersebut sekitar 30 menit.

### D. Pemodelan dan Training Metode YOLO



Gambar 3. Blok diagram pemodelan dengan YOLO

Pada Gambar 3 menunjukkan diagram blok proses *training* model YOLO yang diajukan pada penelitian ini. Dimulai dari import *dataset* lalu setelah itu dilanjutkan *preprocessing* data, yang dilakukan pada saat *preprocessing* yaitu *split dataset* atau pemisahan data adalah metode membagi data menjadi dua bagian. Citra dibagi kedalam folder train dan test untuk dilakukan proses *training* secara terpisah. Selanjutnya dilakukan *training* model menggunakan versi yolov5m6. Yolov5m6 adalah salah satu versi atau varian dari arsitektur YOLO versi kelima (YOLOv5). Varian ini memiliki model dengan tingkat keakuratan yang lebih tinggi daripada varian yang lebih kecil seperti YOLOv5s (*small*) [9]. Setelah itu dilakukan evaluasi akurasi dari model, apakah sudah benar hasil deteksinya dan sudah bagus nilai akurasinya. Selanjutnya menguji model dengan menggunakan gambar jamur yang belum pernah dilihat sebelumnya untuk memeriksa keakuratan dan kinerja keseluruhan dari model tersebut. Terakhir, dilakukan save model YOLO dalam bentuk best.pt yang nanti akan digunakan pada proses *deployment*.

Dataset yang sudah di labelling pada tahap sebelumnya akan dilatih sehingga membentuk sebuah pola yang hasilnya berbentuk Bobot. Bobot tersebut akan digunakan untuk mendeteksi objek di dalam citra. Training akan dilakukan menggunakan metode YOLO. Pertama kali yang dilakukan untuk training adalah mengimport library PyTorch, library ini utamanya digunakan untuk aplikasi yang menggunakan GPU dan CPU [10]. Selanjutnya juga mengimport library yolov5 yang akan kita gunakan. Selanjutnya import file dataset yang telah dilabelling sebelumnya. Dataset tersebut akan dilakukan pembagian dataset menjadi data latih

dan validation data. Selanjutnya perlu mengatur epoch dan batch size seperti

*Confusion matrix* yang terdapat pada Tabel 1 dapat digunakan untuk menentukan hasil akurasi klasifikasi model. Nilai *precision* dan nilai *recall* setiap kelas pada setiap model dapat digunakan untuk menghitung kebaikan klasifikasi. Hasil dalam masalah klasifikasi dapat dikategorikan menjadi empat kategori berdasarkan kombinasi kelas *ground truth* dan kelas prediksi yaitu *true positive* (TP), *false positive* (FP), *true negative* (TN), dan *false negative* (FN) [11]. Ketentuan untuk kelas prediksi berdasarkan letak yang ada pada data *confusion matrix* setelah dilakukan *training*.

Tabel 1. Confusion Matrix

		Actual Class	
		Positive	Negative
Prediction Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Negative (FN)	True Positive (TN)

Persamaan 1,2, dan 3 berikut dapat digunakan untuk menentukan *accuracy*, *precision* dan *recall* berdasarkan data *confusion matrix*.

*Accuracy* adalah rasio hasil aktual terhadap total prediksi sistem. Untuk mencari nilai akurasi dapat dilihat pada persamaan (1)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

*Precision* atau presisi adalah rasio item yang diprediksi terhadap semua hasil yang diprediksi sistem sebagai positif. Untuk mencari nilai presisi dapat dilihat pada persamaan (2)

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

*Recall* merupakan Rasio objek yang diprediksi dengan hasil akhir [12]. Untuk mencari nilai recall dapat dilihat pada persamaan (3).

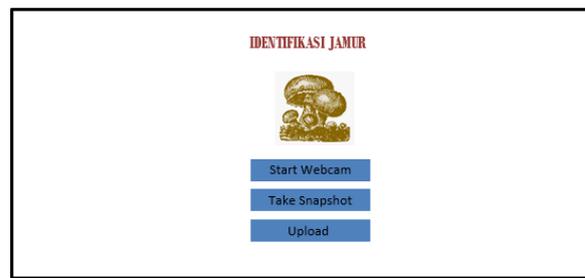
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

#### E. Deployment

Pada *deployment* diperlukan 2 file yaitu file python untuk membuat dengan flask dan file html untuk membuat tampilan *website* yang diinginkan.

*Microframework* web berbasis Python yang disebut Flask disebut sebagai web *framework* [13]. Langkah pertama dalam *deployment* adalah membangun virtual environment menggunakan anaconda command kemudian menginstall flask dan ngrok ke dalam virtual environment setelah menyimpan model mengikuti proses *training*. Langkah selanjutnya adalah mendownload file *template* dari GitHub, dimana gambar tersebut terdapat pada bagian statis, setelah flask dan ngrok sudah berhasil terinstall.

Gambar 4 memberikan ilustrasi desain *website* identifikasi jamur. Ada tiga button yang berlabel 'start webcam' yang berfungsi untuk memulai deteksi melalui webcam dari perangkat yang digunakan. Button 'take snapshot' berfungsi untuk meng *capture* atau mengambil gambar secara langsung melalui webcam yang nantinya format berupa .jpg dan langsung keluar hasil deteksinya. Pada button 'upload' berfungsi untuk mengupload foto yang akan digunakan untuk deteksi.



Gambar 4. Perancangan website

### III. HASIL DAN PEMBAHASAN

Algoritma CNN dan YOLO digunakan dalam penelitian ini untuk membuat web real-time yang membedakan antara jamur beracun dan tidak beracun. Data yang digunakan terdiri dari 1.234 foto, 988 diantaranya digunakan untuk *training* dan 246 untuk validasi. Foto-foto ini dibagi menjadi dua kelas, jamur tidak beracun dan beracun.

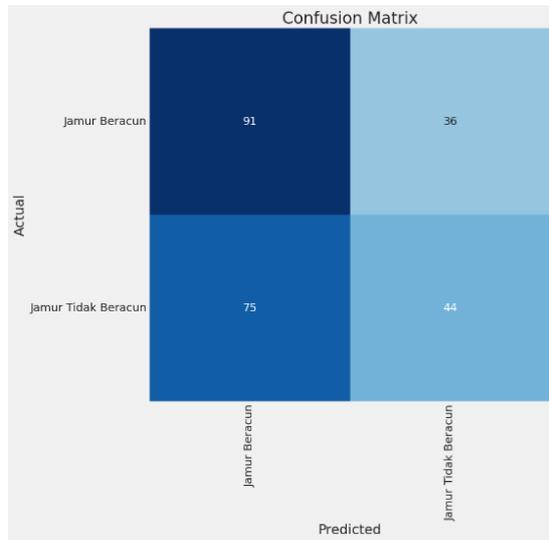
#### A. Hasil Metode CNN

Tabel 2 menunjukkan bahwa model akhir untuk *training* menggunakan metode ini mendapatkan nilai loss 73,5% dan nilai akurasi sebesar 69,1%. Nilai tersebut menunjukkan bahwa akurasi deteksi objek model CNN pada identifikasi jamur beracun dan tidak beracun memiliki nilai yang rendah.

Tabel 2. Hasil Training CNN

Class	Precision	Recall	F1-score	Accuracy
Jamur Beracun	0.55	0.72	0.62	
Jamur Tidak Beracun	0.55	0.37	0.44	
Semua				0.55

Gambar 5 merupakan hasil *confusion matrix* yang didapat. Menggunakan persamaan 1,2, dan 3 maka dilakukanlah perhitungan untuk mencari akurasi.



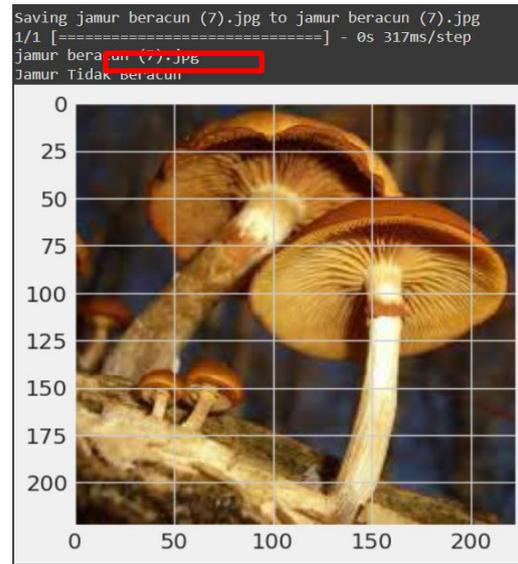
Gambar 5. Confusion matrix CNN

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Accuracy = \frac{91+44}{91+44+36+75} = \frac{135}{246} = 0,55 \times 100\% = 55\%$$

Setelah dilakukan perhitungan, didapatkan nilai akhir akurasi deteksi pada metode CNN sebesar 55%. Pada Gambar 6 terdapat hasil deteksi yang dilakukan setelah raining model. Hasil deteksi tersebut salah karena ia mendeteksi jamur yang beracun menjadi jamur yang tidak beracun. Ternyata menggunakan metode CNN dalam mendeteksi jamur akurasinya rendah dan hasil deteksi tidak akurat. Kemudian saat dilakukan evaluasi model, diberikan 5 gambar jamur beracun diluar dari dataset yang telah ada ia mendeteksi semua gambar jamur beracun tersebut sebagai jamur tidak beracun. Diberikan 5 buah gambar jamur tidak beracun ia mendeteksi benar semua gambar tersebut jamur tidak beracun. Rendahnya akurasi pada metode CNN dalam identifikasi jamur beracun dan tidak beracun bisa dipengaruhi oleh beberapa faktor. Pertama, keterbatasan jumlah data pelatihan yang cukup besar dan representatif dapat menyebabkan kurangnya generalisasi model terhadap variasi jamur dalam kondisi nyata. Kedua, proses tuning parameter yang tidak optimal atau penerapan arsitektur CNN yang kompleks dapat memicu overfitting, di mana model terlalu beradaptasi dengan data pelatihan dan kehilangan kemampuan untuk mengenali pola umum. Terakhir, kurangnya kerjasama antara lapisan-lapisan konvolusi dan dropout yang tidak memadai dapat mempengaruhi kemampuan model untuk mengekstrak fitur secara efektif[14]. Umumnya pada model CNN menggunakan dataset yang banyak, disarankan menggunakan dataset sebanyak puluhan ribu

jumlahnya dengan jumlah per kelas ribuan. Kualitas citra yang diperbaiki bisa berupa resolusi, kontras, ukuran citra, dan sebagainya.



Gambar 6. Hasil deteksi dengan metode CNN

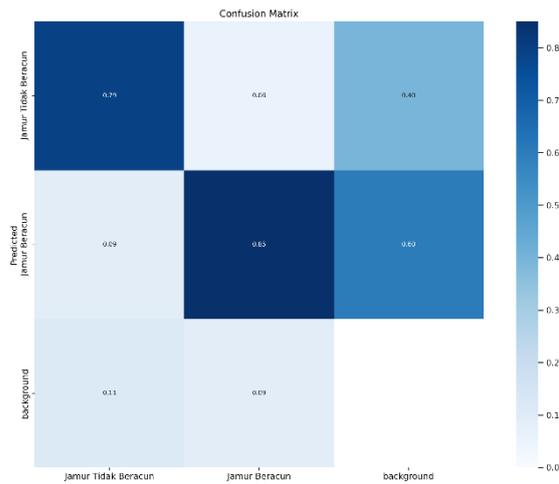
## B. Hasil Metode YOLO

Tabel 3. Menunjukkan bahwa model akhir untuk jamur tidak beracun memiliki precision sebesar 85,3%, Recall sebesar 80,4% dan peringat validasi mAP sebesar 88,5%. Untuk jamur beracun mendapat precision 79,7%, recall 85,6%, dan mAP sebesar 86,6%, skor tersebut menunjukkan bahwa akurasi deteksi objek model YOLOv5 pada identifikasi jamur beracun dan tidak beracun bagus dan akurat.

Tabel 3. Hasil Training YOLOv5

Class	Precision	Recall	mAP
Semua	0.825	0.83	0.875
Jamur Beracun	0.853	0.804	0.885
Jamur Tidak Beracun	0.797	0.856	0.866

Gambar 7 merupakan hasil *confusion matrix* yang didapat. Menggunakan persamaan 1,2, dan 3 maka dilakukanlah perhitungan untuk mencari akurasi.



Gambar 7. Confusion matrix YOLO

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Accuracy = \frac{79+85}{79+85+6+9} = \frac{164}{179} = 0,91 \times 100\% = 91\%$$

Setelah dilakukan perhitungan, didapatkan nilai akhir akurasi deteksi pada metode YOLO sebesar 91%.

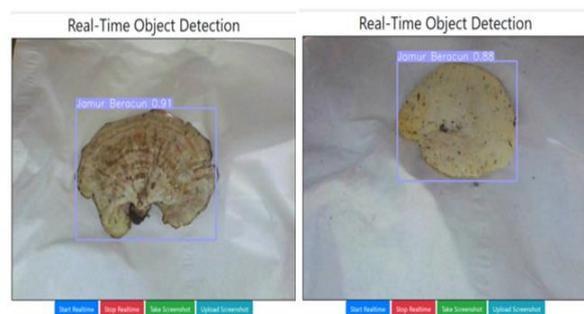
Setelah proses *training* selesai, langkah pertama dalam melakukan pendeteksian citra adalah memasukkan gambar atau video jamur yang akan dites. Keluaran sistem mencakup tingkat *confidence*, nama kelas, dan lokasi koordinat kotak pembatas tempat jamur ditemukan. Seperti yang terlihat pada gambar 8, disana *output* yang diperoleh yaitu hasil deteksi berupa jamur beracun dan nilai *confidence* nya 0,53 atau 53% yang berarti nilai tersebut adalah nilai yang bagus dan metode yang digunakan akurat untuk mendeteksi jamur. Kemudian saat dilakukan evaluasi model, diberikan 5 gambar jamur beracun diluar dari dataset yang telah ada ia mendeteksi 1 gambar sebagai jamur tidak beracun. Diberikan 5 buah gambar jamur tidak beracun ia mendeteksi benar semua gambar tersebut jamur tidak beracun. Pada Google Colab untuk meningkatkan akurasi dapat dilakukan dengan mengubah nilai epoch, batch size, pengaturan versi yang digunakan, meningkatkan kualitas citra, menambah jumlah dataset, dataset diproses menggunakan roboflow, dan juga menggunakan training function yang sesuai.



Gambar 8. Hasil deteksi jamur beracun

### C. Pengujian Secara Realtime

Pengujian dilakukan dengan 2 jenis jamur, 1 jamur beracun dan 1 jamur tidak beracun. Pengujian secara realtime dilakukan dengan kamera dari laptop yang nantinya jamur akan dihadapkan ke kamera dan kamera akan mendeteksi apakah itu jamur beracun atau tidak. Seperti yang terdapat pada gambar 9 dan 10 hasil deteksi berupa bounding box pada bagian objek jamur dan muncul hasil deteksi dan nilai *confidence* nya pada jamur beracun 90%, pada jamur tidak beracun 69%.



Gambar 9. Pengujian pada jamur beracun secara realtime



Gambar 10. Pengujian pada jamur tidak beracun secara realtime

Kemudian dilakukan juga pengujian secara realtime pada jamur tidak beracun, lalu jamur tersebut dikeringkan sehingga membuat tampilan dan warnanya berubah. Seperti pada gambar 11 menampilkan hasil deteksi pada jamur tersebut.



Gambar 11. Pengujian pada jamur yang dibuat menjadi beracun

#### D. Pembahasan

Dapat dilihat pada Tabel 4 perbandingan pada kedua metode yang digunakan untuk nilai *precision*, *recall*, *F1-score* metode CNN lebih rendah dari metode YOLO. Nilai-nilai tersebut mempengaruhi hasil akurasi. Akurasi pada metode CNN hanya mendapatkan nilai 0.55 saja sedangkan untuk YOLO berada di nilai yang tinggi yaitu 0.91. Akurasi pada metode CNN tersebut mendapatkan nilai yang kurang baik bisa disebabkan oleh beberapa faktor seperti jumlah *dataset* yang tidak banyak, resolusi gambar pada *dataset* yang tidak bagus, dan terjadinya *overfitting* pada model. Maka sebaiknya untuk meningkatkan nilai akurasi pada metode yang digunakan bisa dengan melakukan pengambilan *dataset* secara mandiri dan langsung.

Tabel 4. Hasil Metode CNN dan YOLO

	CNN		YOLO	
	Jamur Beracun	Jamur Tidak Beracun	Jamur Beracun	Jamur Tidak Beracun
Precision	0.55	0.55	0.853	0.797
Recall	0.72	0.37	0.804	0.856
F1-Score / mAP	0.62	0.44	0.885	0.866
Accuracy	0.55		0.91	

Metode YOLO menunjukkan performa yang lebih baik, yang dibuktikan dengan nilai *precision*, *recall*, dan *F1-score* yang lebih tinggi. *Precision* pada YOLO untuk jamur beracun adalah 0.853 dan untuk jamur tidak beracun adalah 0.797, sementara *recall* masing-masing adalah 0.804 dan 0.856. Nilai-nilai ini menunjukkan bahwa YOLO mampu mengidentifikasi jamur beracun dan tidak beracun dengan tingkat kesalahan yang lebih rendah dibandingkan CNN. *F1-score* yang lebih tinggi pada YOLO juga menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall*. YOLO dapat mendeteksi jamur secara *real-time* dengan cukup akurat, yang terlihat dari pengujian pada

jamur yang tidak beracun tetapi telah dikeringkan sehingga bentuk dan warnanya berubah. YOLO mampu mendeteksi perubahan ini dan memberikan hasil yang tepat.

Meskipun metode You Only Look Once (YOLO) diyakini telah akurat dalam mendeteksi objek tapi beberapa keterbatasan masih ada. YOLO cenderung kurang efektif dalam mendeteksi objek yang sangat kecil atau memiliki perubahan ukuran yang signifikan dalam gambar. Selain itu, YOLO dapat mengalami kesulitan dalam mengidentifikasi objek yang tumpang tindih atau objek yang terlalu dekat satu sama lain. Variabilitas kondisi pencahayaan yang tinggi atau perbedaan latar belakang dapat memengaruhi performa deteksi. Meskipun telah ada pengembangan dan peningkatan pada model YOLO, pemahaman akan keterbatasan ini tetap penting untuk memastikan keandalan dan kesesuaian penggunaannya dalam berbagai konteks[15].

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa YOLO lebih unggul dibandingkan CNN dalam tugas deteksi jamur, dengan nilai *precision*, *recall*, *F1-score*, dan akurasi yang lebih baik. YOLO mampu mengatasi beberapa tantangan dalam deteksi objek dan memberikan hasil yang lebih akurat dan efisien dalam aplikasi *real-time*. Namun, pemahaman akan keterbatasan YOLO penting untuk memastikan bahwa model ini digunakan dengan tepat dalam berbagai kondisi dan konteks.

#### IV. KESIMPULAN

Penelitian ini melakukan identifikasi jenis jamur beracun dan tidak beracun dengan metode CNN dan YOLO. Setelah dilakukan penelitian, didapatkan bahwa pada metode CNN nilai akurasi sebesar 55% dan hasil deteksi yang dilakukan tidak akurat. Hal itu disebabkan oleh banyak faktor seperti jumlah *dataset* yang tidak banyak dan terjadinya *overfitting* pada model. Sedangkan menggunakan YOLOv5 mendapatkan nilai akurasi yang besar yaitu 91% dan hasil deteksi yang dilakukan akurat dan nilai *confidence* yang cukup bagus yaitu 53%. Dari perbandingan 2 metode tersebut didapatkan bahwa untuk mendeteksi jamur beracun dan tidak beracun itu mengidentifikasi dengan lebih baik menggunakan metode YOLO daripada metode CNN.

Implikasi dari temuan ini sangat relevan dengan tujuan penelitian untuk meningkatkan efisiensi dan akurasi proses identifikasi jamur, menunjukkan bahwa penggunaan YOLO dalam konteks ini dapat memberikan solusi yang lebih efektif. Keunggulan YOLO dalam mendeteksi jamur beracun dan tidak beracun memiliki implikasi positif terutama dalam mendukung keamanan pangan dan kesehatan masyarakat, serta memberikan kontribusi pada pengembangan teknologi deteksi yang lebih canggih dalam bidang ini. Selain itu, temuan ini juga mengisi celah dalam literatur yang menunjukkan bahwa model

YOLO dapat diandalkan untuk tugas deteksi objek yang kompleks seperti identifikasi jamur, yang sebelumnya belum banyak dieksplorasi dalam penelitian. Hal ini menekankan pentingnya pemilihan metode yang tepat dalam analisis citra untuk aplikasi kritis, dan memberikan arahan yang jelas untuk penelitian lebih lanjut dalam meningkatkan teknik deteksi objek menggunakan dataset yang lebih besar dan beragam untuk mencapai hasil yang lebih optimal.

#### REFERENSI

- [1] Y. Suryani, O. Taupiqurrahman, and Y. Kulsum, *Buku Mikologi Dr. Yani Suryani\_Lengkap*, 2020.
- [2] N. L. Chusna, M. I. Shalahudin, U. Riyanto, and A. D. Alexander, Klasifikasi Citra Jenis Tanaman Jamur Layak Konsumsi Menggunakan Algoritma Multiclass Support Vector Machine, *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 1, Jun. 2022, doi: 10.47065/bits.v4i1.1624.
- [3] K. Nurfitri, A. D. Pradana, and I. Widaningrum, Jurnal Rekayasa Teknologi dan Komputasi Penerapan Algoritma Principal Component Analysis (PCA) Dan K-Nearest Neighbors (KNN) Pada Klasifikasi, 2021.
- [4] Y. Yohannes, D. Udjulawa, and T. Ivan Sariyo, Klasifikasi Jenis Jamur Menggunakan SVM dengan Fitur HSV dan HOG, *PETIR*, vol. 15, no. 1, pp. 113–120, Dec. 2021, doi: 10.33322/petir.v15i1.1101.
- [5] E. Iedfitra Haksoro and A. Setiawan, Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network, *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, Online, 2021.
- [6] S. Enggari, A. Ramadhanu, and H. Marfalino, Peningkatan Digital Image Processing Dalam Mendeskripsikan Tumbuhan Jamur Dengan Segmentasi Warna, Deteksi Tepi Dan Kontur, *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 4, no. 1, pp. 70–75, Jan. 2022, doi: 10.47233/jteksis.v4i1.358.
- [7] Li, W., Zhao, X., Li, S., & Niu, D. Real-time recognition of edible and poisonous mushrooms based on deep learning. *Computers and Electronics in Agriculture*, 174, 105450. 2020. doi: 10.1016/j.compag.2020.105450
- [8] Khairunnas, E. Mulyanto Yuniarno, and A. Zaininas, Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot, 2021.
- [9] R. Gelar Guntara, Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7, *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.
- [10] Algorit.ma. Mengenal Unstructured Data. 2022. [Online] Tersedia: <https://algorit.ma/blog/pytorch-adalah-2022/>. [30 Juli 2023]
- [11] A. F. Fandisyah, N. Irawan, and W. S. Winahju, Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3, 2021.
- [12] A. K. E. Lopian, S. R. U. A. , Sompie, and P. D. K. Manembu, You Only Look Once (YOLO) Implementation For Signature Pattern Classification, 2021.
- [13] Shrivastava, H. (2021). Symptomatic Assistance. *International Journal for Research in Applied Science and Engineering Technology*,9(VII).<https://doi.org/10.22214/ijraset.2021.37132>
- [14] Tan, M., & Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2019
- [15] Bochkovskiy, A., theNM22, & AlexeyAB. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020