

## Implementasi *Machine Learning* Pada Deteksi Kecacatan *Printed Circuit Board*

Ahnaf Zainul Muttaqin<sup>1\*</sup>, Ibrahim Nawawi<sup>2</sup>, Johan Pamungkas<sup>3</sup>

<sup>1,2,3</sup>Universitas Tidar, Indonesia

Jl. Kapten Suparman No.39, Potrobangsari, Kec. Magelang Utara, Kota Magelang, Jawa Tengah, Indonesia

\*Corresponding author e-mail : ahnafzainul@gmail.com

### ABSTRAK

Kualitas suatu rangkaian elektronik sangat bergantung pada kualitas *Printed Circuit Board* (PCB) untuk mendukung proses berjalannya semua komponen. Meskipun dapat menggunakan inspeksi manual terhadap kecacatan pada PCB, namun metode tersebut memiliki keterbatasan terhadap ketelitian dan waktu. Oleh karena itu, penelitian ini dilakukan untuk mengimplementasikan *Machine Learning* dalam deteksi kecacatan pada PCB. Dalam pengaplikasiannya, penelitian ini menggunakan algoritma deteksi objek YOLOv5 (*You Look Only Once* versi 5), yang merupakan pengembangan dari *convolutional neural network*. Fokus penelitian ini adalah membangun sistem yang dapat mengidentifikasi kecacatan pada PCB menggunakan algoritma YOLOv5. Dataset yang digunakan terdiri dari 600 gambar, dengan 480 data latih dan 120 data uji. Jenis kecacatan PCB yang diidentifikasi meliputi *open circuit*, *missing hole*, *mouse bite*, *short*, *spur*, dan *spurious copper*. Hasil penelitian menunjukkan bahwa penggunaan YOLOv5m menghasilkan nilai *mean Average Precision* (mAP) sebesar 95,3%. Selain itu, dalam pengujian dengan 120 data uji, model berhasil mencapai akurasi sebesar 93,83%, presisi 98,13%, *recall* 95,43%, dan *error* 6,17%. Selain itu peningkatan spesifikasi *hardware* juga berpengaruh dalam kecepatan deteksi objek kecacatan pada PCB.

**Kata kunci** : PCB, kecacatan, *Machine Learning*, *convolutional neural network*, YOLOv5

### ABSTRACT

*The quality of an electronic circuit depends significantly on the quality of the Printed Circuit Board (PCB) supporting its operation. While manual inspection of defects on PCBs has been carried out, this method has limitations in terms of precision and time. Therefore, this research aims to implement Machine Learning in detecting defects on PCBs. In its application, this study utilizes the YOLOv5 (You Look Only Once version 5) object detection algorithm, which is an advancement of the convolutional neural network. The focus of this research is to build a system capable of identifying defects on PCBs using the YOLOv5 algorithm. The dataset used consists of 600 images, with 480 training data and 120 test data. Types of PCB defects identified include open circuit, missing hole, mouse bite, short, spur, and spurious copper. The research results indicate that using YOLOv5m yields a mean Average Precision (mAP) value of 95.3%. Additionally, in testing with 120 test data, the model achieved an accuracy of 93.81%, precision of 98.13%, recall of 95.43%, and an error rate of 6.17%. Furthermore, the improvement of hardware specifications also influences the speed of object detection for defects on PCBs.*

**Keywords:** PCB, defect, *Machine Learning*, *convolutional neural network*, YOLOv5

## I. PENDAHULUAN

Kualitas *Printed Circuit Board* (PCB) memiliki signifikansi penting dalam memastikan kinerja optimal suatu rangkaian elektronik. Dalam beberapa kasus, proses pembuatan jalur dan penyolderan dapat menyebabkan adanya cacat atau kecacatan pada jalur PCB atau solder tersebut. Inspeksi visual manual umumnya menjadi salah satu aspek pengeluaran

terbesar dalam proses pembuatan PCB [1]. Selain itu, karena keterbatasan kemampuan manusia, proses pemeriksaan PCB secara manual menjadi efektif karena melibatkan pengecekan setiap jalur dan solder secara individual, yang dapat memakan waktu yang cukup signifikan. Meskipun inspeksi manual PCB berguna, namun memiliki kelemahan seperti kebutuhan waktu yang dapat menyebabkan kelelahan

dan hasil inspeksi yang bervariasi jika terjadi pergantian inspektor [2].

Salah satu metode yang digunakan dalam melakukan inspeksi kecacatan pada PCB adalah melalui pengolahan citra digital. Dalam pendekatan ini, citra dari PCB digunakan sebagai media yang diolah melalui komputer. Dengan kata lain, informasi mengenai kecacatan PCB dapat diekstrak melalui manipulasi citra digital. Sebelumnya, telah dilakukan penelitian terkait penggunaan pengolahan citra digital untuk mendeteksi kecacatan pada PCB, khususnya melalui implementasi *canny edge detection* untuk mengidentifikasi kecacatan pada solder. Meskipun penelitian tersebut telah memberikan kontribusi, namun karena masih mengadopsi metode pengolahan citra konvensional, penelitian ini memiliki keterbatasan dengan fokus yang terbatas pada deteksi tepi. Sebagai dampaknya, penelitian ini tidak mampu menangani tugas-tugas yang lebih kompleks dalam konteks inspeksi kecacatan pada PCB [3].

Dalam perkembangan, pengolahan citra digital juga menjadi bagian langsung dari teknologi *Artificial Intelligence* (AI). Salah satu pendekatan yang berhubungan dengan AI yaitu *machine learning* yang mana merupakan suatu metode di mana algoritma digunakan untuk mengembangkan sistem atau model komputer yang dapat belajar dari data atau pengalaman. Dengan demikian maka model tersebut dapat membuat prediksi atau keputusan berdasarkan informasi yang telah dipelajari dari data dan pengalaman selama proses pelatihan. Dalam implementasinya, *machine learning* dapat digunakan untuk mengidentifikasi objek atau benda pada citra dengan memanfaatkan algoritma deteksi objek.

Umumnya, algoritma deteksi objek menggunakan *Convolutional Neural Network* (CNN) sebagai dasar, di mana gambar diolah sebagai *array* terstruktur untuk mengidentifikasi pola tertentu. CNN sendiri termasuk ke dalam *supervised learning* yang mana membutuhkan data yang diberi label [4] [5]. Proses pengenalan pola pada CNN bergantung pada data latih yang telah digunakan untuk melatih model tersebut. CNN dapat digunakan atau diimplementasikan untuk pengenalan citra dengan akurasi yang menyaingi manusia pada suatu dataset tertentu dan CNN dapat mempelajari jenis fitur yang menghasilkan akurasi yang lebih tinggi dan karenanya menggunakannya dalam proses klasifikasi [6].

Implementasi *Machine Learning* sebagai metode untuk melakukan deteksi kecacatan PCB juga telah dilakukan yang mana pengolahan citra digital dipadukan dengan *neural network* sehingga bisa menangani tugas yang kompleks. Metode yang digunakan pada penelitian ini yaitu Fast R-CNN dengan *backbone* ResNet-101 yang didesain ulang.

Hasil dari penelitian ini menunjukkan algoritma TDD-Net bisa mencapai nilai *mean Average Precision* (mAP) sebesar 98.90%. Namun penggunaan Fast R-CNN agak lambat dalam melakukan deteksi dan memiliki komputasi yang cukup berat. Hal tersebut dikarenakan Fast R-CNN memerlukan tahap pra-pemrosesan yang memakan waktu, seperti ekstraksi fitur dan pencocokan gambar dengan *bounding box* [7].

Disamping itu, terdapat algoritma deteksi objek yang dikembangkan berdasarkan *Convolutional Neural Network* (CNN) yang dikenal dengan nama *You Only Look Once* (YOLO). YOLO merupakan metode deteksi objek yang menggunakan jaringan saraf konvolusi (CNN) untuk menganalisis citra atau video secara keseluruhan dalam satu langkah, menghasilkan deteksi objek dengan kecepatan tinggi. YOLO memiliki berbagai versi, salah satunya adalah YOLOv5. Kecepatan inferensi YOLOv5 ternyata 10 kali lebih tinggi dibandingkan dengan metode *Faster R-CNN* [8]. YOLOv5 dirancang dengan perhatian khusus pada efisiensi. Dalam arti ini, model yang dihasilkan dari pelatihan YOLOv5 lebih kecil dalam ukuran, memerlukan lebih sedikit sumber daya komputasi, dan dapat dijalankan dengan cepat pada perangkat dengan daya terbatas.

Berdasarkan uraian penjelasan sebelumnya, dilakukan penelitian terkait penerapan *Machine Learning* dalam deteksi kecacatan pada *printed circuit board* (PCB). Dalam implementasinya, metode *machine learning* yang diadopsi adalah algoritma deteksi objek YOLOv5, yang merupakan pengembangan dari *convolutional neural network*. Dalam konteks penelitian ini, model YOLOv5 akan dilatih menggunakan dataset gambar PCB yang telah diberi label, sehingga mampu mengidentifikasi objek yang berupa kecacatan pada PCB.

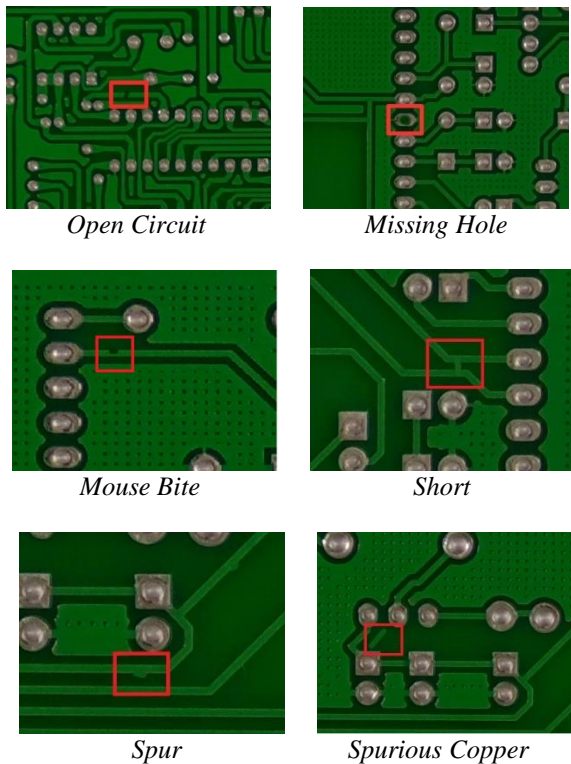
## II. METODE

Pada penelitian ini diimplementasikan algoritma *You Only Look Once* versi 5 (YOLOv5) dalam mendeteksi objek kecacatan pada *Printed Circuit Board* (PCB). Adapun objek kecacatan pada PCB terbagi menjadi 6 kelas yaitu : *open circuit*, *missing hole*, *mouse bite*, *short*, *spur*, dan *spurious copper*. Ketika model objek deteksi selesai dibangun maka selanjutnya model akan dilakukan pengujian untuk mengetahui tingkat keberhasilannya dalam mendeteksi objek kecacatan PCB dengan menggunakan matriks evaluasi.

### A. Pengambilan dan Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan citra PCB sekunder yang berasal dari sumber terbuka, dirilis oleh Lab Interaksi Robot

Manusia Universitas Peking. Sebanyak 600 citra PCB dipilih dari dataset ini, di mana setiap citra mencakup enam kelas atau parameter objek yang perlu dideteksi. Dengan kata lain, setiap kelas memiliki jumlah yang sama, yakni 100 citra PCB. Parameter yang menjadi fokus dalam analisis sistem deteksi kecacatan PCB ini mencakup objek berupa:



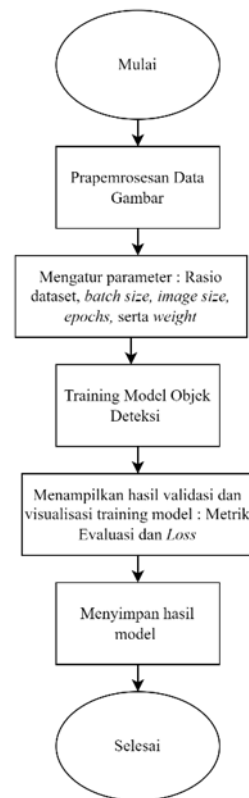
Gambar 1. Citra PCB dan Objek Kecacatannya

Dari total 600 citra data yang tersedia, akan diambil sampel untuk dijadikan sebagai data uji. Pembagian data pada penggunaan *machine learning* biasanya mengalokasikan sekitar 20% data untuk pengujian [9]. Dalam konteks penelitian ini, sampel data uji dipilih sejumlah 20%. Proses pengambilan data uji dilakukan melalui metode *stratified sampling*, di mana populasi dibagi menjadi beberapa sub-kelompok atau strata berdasarkan karakteristik yang relevan. *Stratified sampling* dianggap sebagai metode pengambilan sampel yang efisien karena mempertimbangkan variasi dalam populasi dan memastikan representasi proporsional dari setiap kelas. Dataset yang digunakan terdiri dari enam kelas yang terdistribusi secara merata, sehingga setiap kelas akan diambil sejumlah 20 citra secara acak. Mengingat adanya variasi dalam populasi, penggunaan *stratified sampling* dianggap lebih efisien dibandingkan dengan pengambilan sampel acak biasa [10]. Pendekatan *stratified sampling* diaplikasikan untuk memastikan bahwa setiap kelas dalam keseluruhan dataset diwakili secara proporsional dalam data uji. Data uji ini akan

digunakan untuk mengevaluasi hasil akhir dari sistem.

## B. Perancangan dan Pembuatan Sistem *Machine Learning*

Ada beberapa langkah yang terlibat dalam pengembangan sistem deteksi pada kecacatan PCB, meliputi pra-pemrosesan data hingga penyimpanan hasil dari model deteksi objek. Desain sistem ini dibangun berdasarkan tinjauan literatur yang mendalam. Diagram alir keseluruhan cara kerja sistem yang akan dikembangkan dapat dilihat pada Gambar 2 berikut ini:



Gambar 2. Diagram Alir Perancangan dan Pembuatan Sistem

Berdasarkan gambar 2 dapat diuraikan tahapan perancangan dan pembuatan sistem sebagai berikut:

### 1. Pra-pemrosesan Data

Proses pra-pemrosesan data yang terjadi di sini adalah pemberian label pada gambar. Dalam konteks penelitian ini, label yang diterapkan bertujuan untuk mengenali cacat pada PCB. Proses pelabelan ini dilakukan secara manual menggunakan perangkat lunak LabelIMG. Pelabelan dilakukan dengan menempatkan kotak pembatas (*bounding box*) pada gambar, yang menunjukkan lokasi objek, area, atau fitur khusus dalam gambar dengan label yang sesuai. Melalui pemberian label ini, area atau elemen yang mengandung cacat dalam gambar PCB dapat diidentifikasi dan dibedakan dengan jelas.

Dalam proses perancangan sistem model deteksi objek, ada beberapa parameter yang perlu ditentukan sebelum proses pelatihan model dimulai. Penetapan nilai parameter ini dilakukan secara eksperimental dengan mencoba berbagai nilai pada setiap *hyperparameter* melalui serangkaian eksperimen dan evaluasi performa model. Berikut adalah parameter yang diatur sebelum melaksanakan proses pelatihan:

- a. Rasio dataset : Pembagian data latih menjadi data *train* dan data *validation*. Rasio yang digunakan yaitu 80% untuk data *train* dan 20% untuk data *validation*.
- b. *Batch Size* : *Batch size* yang terlalu kecil dapat membuat pelatihan memakan waktu lebih lama, sedangkan *batch size* yang terlalu besar dapat menghabiskan sumber daya GPU atau RAM yang dimiliki. Pada penelitian ini ukuran *batch size* diatur secara bawaan yaitu bernilai 16.
- c. *Image Size* : Ukuran gambar masukan yang terlalu kecil dapat menyebabkan model kehilangan detail objek, sedangkan ukuran gambar masukan yang terlalu besar dapat memperlambat waktu pelatihan. *Image size* yang umum digunakan adalah 640x640.
- d. *Epochs* : Jumlah *epoch* yang terlalu sedikit dapat membuat model *underfitting* (model tidak mempelajari pola dengan baik) , sedangkan jumlah *epoch* yang terlalu banyak dapat menyebabkan *overfitting* (model terlalu fokus mempelajari pola). Oleh karena itu jumlah *epoch* diatur sebanyak 1000 namun jika dalam 50 iterasi tidak ada kenaikan maka proses pelatihan dihentikan.
- e. *Weight* : *Pretrained model* yang digunakan untuk *transfer learning* sebagai dasar untuk melatih model baru. Pada penelitian menggunakan *weight yolo5m.pt* yang merupakan *pretrained model medium* yang tidak terlalu membutuhkan komputasi yang berat.

## 2. Training Model

Dalam proses pelatihan model deteksi objek, digunakan algoritma YOLOv5 yang merupakan model deteksi objek berbasis *Convolutional Neural Network* (CNN). Arsitektur YOLOv5 terdiri dari tiga komponen utama, yaitu *backbone*, *neck*, dan *head*. Komponen *backbone* menggunakan jaringan syaraf konvolusi untuk mengekstraksi fitur dari gambar input, bertanggung jawab dalam menghasilkan fitur-fitur tingkat tinggi dari gambar. Setelah itu, komponen *neck* digunakan untuk menggabungkan dan memproses fitur-fitur yang dihasilkan oleh

*backbone*. Bagian terakhir adalah *head*, yang menghasilkan prediksi deteksi objek. *Head* pada YOLOv5 terdiri dari lapisan konvolusi yang menghasilkan kotak pembatas (*bounding box*) dan probabilitas kelas objek.

## C. Pengujian

Pengujian pada sistem ini dilakukan dengan tujuan untuk mengetahui tingkat keberhasilan rancangan sistem yang dibuat dalam mendeteksi kecacatan pada PCB sehingga dapat dipastikan sistem tersebut berfungsi dengan baik atau tidak. Pengujian sistem dilakukan melalui dua tahap yaitu, sebagai berikut:

### 1. Pengujian Model Deteksi

Uji coba dilakukan dengan menganalisis hasil dari pelatihan model menggunakan data gambar latih dan data gambar validasi yang sebelumnya telah dipisahkan. Untuk mengevaluasi model yang telah dilatih, digunakan metode perhitungan akurasi berupa *mean Average Precision* (mAP). Dalam konteks deteksi objek, mAP dihitung dengan membandingkan prediksi model dengan *ground truth* atau label objek sebenarnya. Dalam YOLO, perhitungan mAP melibatkan *Intersection over Union* (IoU), presisi, dan *recall*.

IoU digunakan untuk mengukur sejauh mana prediksi objek tumpang tindih dengan *ground truth* (objek sebenarnya) pada dataset pengujian. IoU membantu menentukan kebenaran atau ketidakbenaran suatu prediksi. Nilai IoU di atas ambang batas menunjukkan bahwa suatu prediksi dapat dianggap benar. IoU digunakan untuk menilai keakuratan *bounding box*. *Bounding box* dengan nilai IoU di atas ambang batas dianggap sebagai *true positive*, sementara yang di bawahnya dianggap sebagai *false positive*. Persamaan untuk menghitung IoU adalah sebagai berikut:

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (1)$$

Setelah menentukan *bounding box* mana yang benar dan salah berdasarkan IOU maka dapat dihitung nilai AP (*Average Precision*). AP dihitung dari rata-rata presisi berdasarkan nilai *recall* yang bervariasi. Formula yang digunakan untuk menghitung mAP (*Mean Average Precision*) pada model deteksi objek seperti YOLO yaitu dengan dengan mengambil rata-rata dari nilai AP pada setiap kelas objek.

$$mAP = \frac{AP_1 + AP_2 + \dots + AP_n}{n} \quad (2)$$

Dimana :

$n$  = Jumlah kelas objek yang dideteksi

$AP$  = *Average Precision* pada kelas objek ke- $i$ .

## 2. Pengujian Sistem Deteksi Kecacatan PCB

Proses pengujian sistem dilakukan dengan menggunakan 120 data gambar uji. Setiap gambar diuji sebanyak tiga kali percobaan untuk mengamati kemungkinan perbedaan dalam deteksi objek. Hasil pengujian kemudian disusun dalam bentuk *confusion matrix*, seperti yang terlihat pada tabel 1.

Tabel 1. Confusion Martrix

|          |         | Nilai Aktual                  |                               |
|----------|---------|-------------------------------|-------------------------------|
|          |         | Positif                       | Negatif                       |
| Prediksi | Positif | <i>True Positive</i><br>(TP)  | <i>False Positive</i><br>(FP) |
|          | Negatif | <i>False Negative</i><br>(FN) | <i>True Negative</i><br>(TN)  |

*Confusion matrix* ini akan digunakan untuk menghitung akurasi, presisi dan *recall* dari sistem. Untuk menghitung berbagai jenis matriks evaluasi seperti akurasi, presisi, dan lain-lain dapat menggunakan keempat nilai dari *confusion matrix*. Rumus untuk menghitung nilai matriks evaluasi dapat dilihat di bawah ini:

- a. Akurasi merupakan rasio antara jumlah prediksi benar dengan dengan total kelas atau kategori. Secara matematis dapat dihitung dengan persamaan berikut :

$$\text{Akurasi} = \frac{TP}{TP + FP + FN} \quad (3)$$

- b. Presisi memberikan informasi mengenai berapa persentase dari hasil positif yang benar-benar benar, sehingga mengukur sejauh mana model mampu menghindari kesalahan *false positive*. Secara matematis dapat dihitung dengan persamaan berikut :

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (4)$$

- c. Sensitifitas atau juga dikenal sebagai *True Positive Rate* (TPR) atau *Recall*. Sensitivitas berguna untuk mengukur sejauh mana model dapat menghindari kesalahan *false negative*.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

Adapun untuk perhitungan *error* atau persentase selisih antara data aktual dan data prediksi. Jika error biasa merupakan rasio dari nilai prediksi dan aktual maka MSE dihitung sebagai rata-rata dari kuadrat selisih antara nilai koordinat prediksi *bounding box* dengan nilai koordinat target *bounding box* yang sesuai, dan akan diperhitungkan untuk setiap *bounding box* pada setiap gambar dalam dataset. Semakin kecil nilai MSE, semakin baik performa model dalam melakukan prediksi *bounding box* dan label yang tepat untuk objek pada gambar.

$$\text{error} = \frac{(y - y_i)}{y} \quad (6)$$

Dimana :

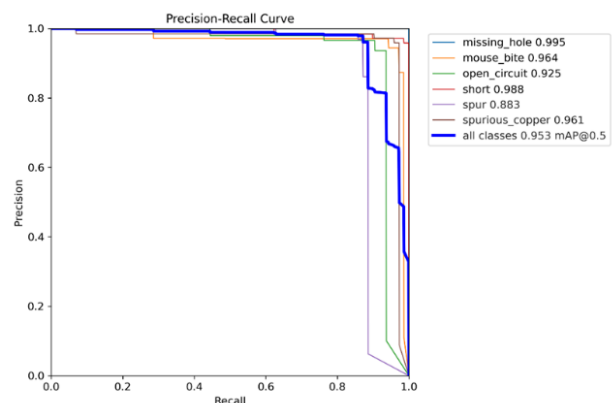
$n$  = Jumlah data dalam dataset

$y$  = *Output* yang sebenarnya

$y_i$  = nilai prediksi dari model untuk data ke- $i$

## III. HASIL DAN PEMBAHASAN

Setelah proses pelatihan model objek deteksi selesai dilakukan. Maka model akan melalui proses validasi. Model dievaluasi menggunakan matriks evaluasi yaitu *mean Average Precision* (mAP). Adapun hasil validasi dari model objek deteksi yang telah dibuat, ditunjukkan pada Gambar 3.



Gambar 3. Grafik *Precision-Recall Curve*

Pada Gambar 3, terlihat grafik kurva *precision-recall* yang memberikan gambaran performa model dalam hal presisi dan *recall* dengan berbagai ambang batas *threshold*. Dengan menggunakan grafik tersebut, nilai *Average Precision* (AP) pada setiap kelas dapat dihitung. Secara keseluruhan, rata-rata AP atau mAP mencapai 0.953.

Proses pengujian model *machine learning* untuk mendeteksi kecacatan pada PCB melibatkan sampel data sebanyak 120 citra yang telah dikumpulkan sebelumnya. Model *machine learning* akan melakukan deteksi kecacatan pada setiap citra PCB yang berfungsi sebagai data uji. Setelah proses selesai, hasil deteksi secara otomatis akan disimpan dalam folder "runs/detect". Ilustrasi dari proses

pengujian deteksi kecacatan pada citra PCB dapat dilihat pada Gambar 4.

```
Fusing layers...
Model summary: 212 layers, 20873139 parameters, 0 gradients, 47.9 GFLOPs
image 1/120 C:\test\data uji ke-1.jpg: 352x640 3 missing_holes, 703.8ms
image 2/120 C:\test\data uji ke-10.jpg: 352x640 3 open_circuits, 707.6ms
image 3/120 C:\test\data uji ke-100.jpg: 512x640 5 spurs, 1345.9ms
image 4/120 C:\test\data uji ke-101.jpg: 576x640 5 missing_holes, 1285.8ms
image 5/120 C:\test\data uji ke-102.jpg: 576x640 6 mouse_bites, 1292.2ms
image 6/120 C:\test\data uji ke-103.jpg: 576x640 5 open_circuits, 1302.2ms
image 7/120 C:\test\data uji ke-104.jpg: 576x640 4 shorts, 1264.3ms
image 8/120 C:\test\data uji ke-105.jpg: 576x640 5 spurious_coppers, 1375.5ms
image 9/120 C:\test\data uji ke-106.jpg: 640x640 5 mouse_bites, 1383.0ms
image 10/120 C:\test\data uji ke-107.jpg: 640x640 5 mouse_bites, 1450.9ms
```

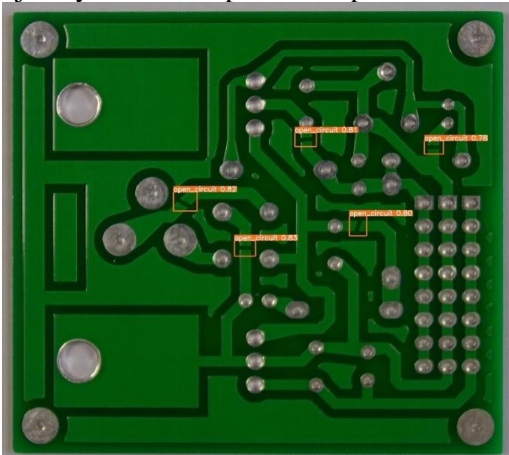
Gambar 4. Proses Pengujian Model *Machine Learning*

Pada pengujian yang terlihat dalam Gambar 4, menggunakan laptop dengan CPU Ryzen 3 menunjukkan bahwa pada pengujian data ke-1 hingga ke-20, sebagian besar data yang diuji memerlukan waktu lebih dari 1000ms atau 1 detik. Waktu pengujian ini tentu bergantung pada spesifikasi perangkat keras yang digunakan, di mana semakin tinggi spesifikasinya, semakin cepat proses deteksi dapat dilakukan. Hal ini terbukti ketika pengujian dilakukan menggunakan komputer dengan spesifikasi lebih tinggi, yakni menggunakan CPU Intel i3 dan GPU Nvidia GTX 1650, seperti yang diperlihatkan dalam Tabel 2.

Tabel 2. Perbandingan Waktu Pengujian

| Spesifikasi                  | Rata-rata waktu pengujian |
|------------------------------|---------------------------|
| Ryzen 3                      | 1225 ms                   |
| Intel i3 dan Nvidia GTX 1650 | 55,65 ms                  |

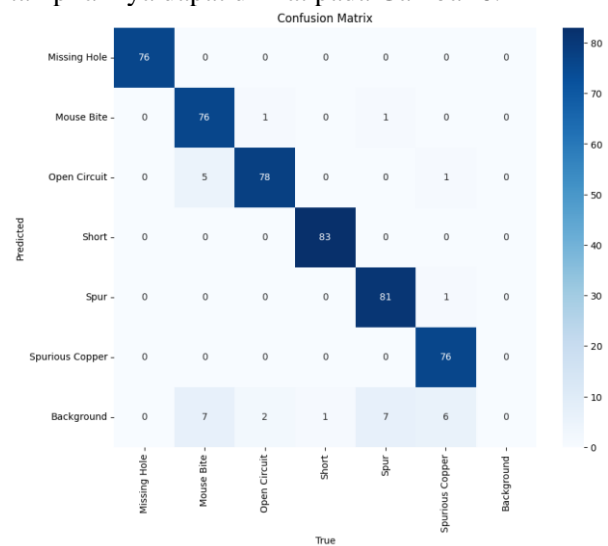
Pada Tabel 2 dapat dilihat bahwa dengan spesifikasi yang tinggi mampu untuk mempercepat proses pengujian. Pengujian hanya dengan menggunakan cpu ryzen 3 menunjukkan waktu pengujian rata-rata selama 1224 ms sedangkan dengan cpu intel i3 dan gpu nvidia GTX 1650 yaitu 55,65 ms.. Selain itu dengan meningkatkan perangkat keras maka proses inspeksi kecepatan PCB dapat dipercepat. Untuk contoh hasil pengujianya sendiri dapat dilihat pada Gambar 5.



Gambar 5. Contoh Hasil Deteksi Kecacatan PCB pada Data Uji

Dalam contoh hasil deteksi data uji pada Gambar 5, setiap objek kecacatan PCB yang terdeteksi akan ditampilkan dengan kotak pembatas (*bounding box*) dan *confidence score*-nya. *Confidence score* ini mencerminkan tingkat kepercayaan model terhadap objek yang terdeteksi. Sebagai contoh, model mendeteksi objek kecacatan berupa *open circuit* dengan *confidence score* sebesar 0.83, yang berarti model yakin sebanyak 83% bahwa objek yang terdeteksi merupakan *open\_circuit*.

Setelah menyelesaikan pengujian dengan 120 data, langkah selanjutnya adalah melakukan analisis terhadap hasil pengujian tersebut. Analisis data pengujian dilaksanakan dengan memeriksa hasil pengujian pada sistem *machine learning* yang telah dikembangkan. Semua data uji yang telah diuji pada model *machine learning* akan dianalisis secara individual. Objek yang terdeteksi dalam data uji akan dievaluasi untuk memastikan bahwa hasil prediksi sesuai dengan objek yang sebenarnya. Semua hasil prediksi objek yang terdeteksi pada setiap gambar akan dicatat dalam sebuah tabel yang nantinya akan digunakan untuk menyusun *confusion matrix*. *Confusion matrix* ini akan diwujudkan dalam bentuk heatmap untuk mempermudah pembacaan, dan tampilannya dapat dilihat pada Gambar 6.



Gambar 6. *Confusion Matrix*

Pada Gambar 6, disajikan *confusion matrix* hasil pengujian yang berfungsi sebagai perbandingan antara objek cacat yang sebenarnya dengan prediksi yang diperoleh selama pengujian. Ketika hasil prediksi sesuai dengan objek yang sebenarnya, hal tersebut dinyatakan sebagai *True Positive* (TP). Sebagai contoh, jika objek *missing hole* diprediksi sebagai *missing hole*. Sebaliknya, jika hasil prediksi menunjukkan objek yang sebenarnya tidak sesuai, maka itu dikategorikan sebagai *False Positive* (FP). Sebagai contoh, objek *mouse bite* diprediksi sebagai *open circuit*. Selain itu, ketika suatu objek seperti

*open circuit* tidak terdeteksi dan diprediksi sebagai background, itu dikategorikan sebagai *False Negative* (FP). Dalam konteks deteksi objek, tidak ada kategori *True Negative* (FN) karena *background* sendiri tidak dianggap sebagai objek.

Berdasarkan hasil pengujian objek kecacatan *mouse bite* paling banyak mengalami salah deteksi yaitu sebanyak 5 kali. Dalam kasus tersebut objek *mouse bite* terdeteksi sebagai *open circuit*, hal tersebut dikarenakan memang ada kemiripan antara objek *mouse bite* dengan *open circuit*. Selain itu untuk objek yang paling banyak mengalami kegagalan deteksi oleh sistem yaitu kecacatan *mouse bite* dan *spur* sebanyak 7 kali. Dari semua data yang telah diuji pada *confusion matrix*, maka selanjutnya dapat dilihat ringkasan dari *confusion matrix* sebelumnya seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Ringkasan *Confusion Matrix*

| No | Objek                  | TP | FP | FN |
|----|------------------------|----|----|----|
| 1  | <i>Missing Hole</i>    | 76 | 0  | 0  |
| 2  | <i>Mouse Bite</i>      | 76 | 5  | 7  |
| 3  | <i>Open circuit</i>    | 78 | 1  | 2  |
| 4  | <i>Short</i>           | 83 | 0  | 1  |
| 5  | <i>Spur</i>            | 81 | 1  | 7  |
| 6  | <i>Spurious Copper</i> | 76 | 2  | 6  |

Dari tabel dapat dihitung nilai matriks evaluasinya, seperti yang terlihat pada Tabel 4.

Tabel 4. Hasil Perhitungan Matriks Evaluasi

| No                | Objek                  | Akurasi %<br>(akurasi x 100%) | Presisi %<br>(presisi x 100%) | Recall %<br>(recall x 100%) | Error %<br>(error x 100%) |
|-------------------|------------------------|-------------------------------|-------------------------------|-----------------------------|---------------------------|
| 1                 | <i>Missing Hole</i>    | 100                           | 100                           | 100                         | 0                         |
| 2                 | <i>Mouse Bite</i>      | 86,36                         | 93,83                         | 91,57                       | 13,64                     |
| 3                 | <i>Open circuit</i>    | 96,30                         | 98,73                         | 97,50                       | 3,70                      |
| 4                 | <i>Short</i>           | 98,81                         | 100                           | 98,81                       | 1,19                      |
| 5                 | <i>Spur</i>            | 91,01                         | 98,78                         | 92,04                       | 8,98                      |
| 6                 | <i>Spurious Copper</i> | 90,48                         | 97,43                         | 92,68                       | 9,52                      |
| <b>Rata -Rata</b> |                        | 93,83                         | 98,13                         | 95,43                       | 6,17                      |

Berdasarkan hasil pengujian objek kecacatan *mouse bite* paling banyak mengalami salah deteksi yaitu sebanyak 5 kali. Dalam kasus tersebut objek *mouse bite* terdeteksi sebagai *open circuit*, hal tersebut dikarenakan memang ada kemiripan antara objek *mouse bite* dengan *open circuit*. Selain itu untuk objek yang paling banyak mengalami kegagalan deteksi oleh sistem yaitu kecacatan *spur* dan *mouse bite* juga sebanyak 7 kali.

Dari hasil perhitungan data uji yang telah dilakukan, dapat dilihat bahwa persentase rata-rata

akurasi dari sistem adalah sebesar 93,83% dengan objek *missing hole* memiliki akurasi tertinggi yaitu 100% dan objek *mouse bite* terendah dengan 86,36%. Dengan akurasi tersebut artinya sekitar 93,83% objek kecacatan PCB pada data uji mampu dideteksi oleh model *machine learning* telah dibuat. Selanjutnya nilai rata-rata presisi dari sistem yaitu 98,13% yang artinya dari semua prediksi terhadap kecacatan PCB, sekitar 98,33% hasil prediksinya benar sesuai dengan kelas objeknya. Selain itu *recall* sendiri memiliki nilai rata-rata 95,43% yang mana berarti model telah berhasil mendeteksi 95,43% dari semua objek positif yang sebenarnya ada dalam. Terakhir untuk *error*-nya sendiri yaitu 6,17 % yang mana menunjukkan tingkat kegagalan dari keseluruhan objek yang terdeteksi.

#### IV. KESIMPULAN

Secara keseluruhan, penerapan algoritma YOLOv5 dalam inspeksi kecacatan PCB menunjukkan peningkatan kinerja seiring dengan spesifikasi perangkat keras yang lebih tinggi. Model *machine learning* yang dihasilkan berhasil menghasilkan mAP sebesar 95,3% dan menunjukkan kemampuannya dengan akurasi 93,83%, presisi 98,13%, dan *recall* 95,43% dalam pengujian menggunakan 120 data uji. Meskipun model menunjukkan performa optimal dalam mendeteksi objek "*missing hole*" dengan hasil evaluasi sempurna, tantangan muncul pada deteksi objek "*mouse bite*" yang menunjukkan nilai akurasi 86,36%, presisi 93,83%, dan *recall* 91,57% serta *error* yang cukup besar yaitu 13,64%. Penemuan ini memberikan gambaran komprehensif tentang efektivitas algoritma YOLOv5 dalam mengidentifikasi kecacatan PCB dengan menggambarkan keunggulan dan tantangan yang ditemui.

#### DAFTAR PUSTAKA

- [1] S. Tang, F. He, X. Huang, and J. Yang, "Online PCB Defect Detector On A New PCB Defect Dataset," Feb. 2019, [Online]. Available: <http://arxiv.org/abs/1902.06197>
- [2] I. C. Chen, R. C. Hwang, and H. C. Huang, "PCB Defect Detection Based on Deep Learning Algorithm," *Processes*, vol. 11, no. 3, Mar. 2023, doi: 10.3390/pr11030775.
- [3] Y. Ningsih, D. Wahiddin, S. Arum, and P. Lestari, "Implementasi Metode Canny Edge Detection Untuk Identifikasi Defect Solder," vol. II, no. 1, 2021.
- [4] B. Farnham, S. Tokyo, B. Boston, F. Sebastopol, and T. Beijing, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow

- Concepts, Tools, and Techniques to Build Intelligent Systems SECOND EDITION.”
- [5] U. Sri Rahmadhani and N. Lysbetti Marpaung, “Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN,” vol. 8, no. 2, 2023.
- [6] F. Paraijun *et al.*, “Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah,” vol. 11, no. 1, 2022, doi: 10.33322/kilat.v11i1.1458.
- [7] R. Ding, L. Dai, G. Li, and H. Liu, “TDD-Net: A tiny defect detection network for printed circuit boards,” *CAAI Trans Intell Technol*, vol. 4, no. 2, pp. 110–116, Jun. 2019, doi: 10.1049/trit.2019.0019.
- [8] T. Mahendrakar *et al.*, *Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets*.
- [9] V. R. Joseph, “Optimal ratio for data splitting,” *Stat Anal Data Min*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
- [10] P. Kanah Arieska, N. Herdiani, P. Studi Ilmu Kesehatan Masyarakat, F. Kesehatan, and U. Nahdlatul Ulama Surabaya Alamat, “PEMILIHAN TEKNIK SAMPLING BERDASARKAN PERHITUNGAN EFISIENSI RELATIF,” 2018. [Online]. Available: <http://jurnal.unimus.ac.id>