

PERBAIKAN LOKAL KANDIDAT CACAT TANPA PEMBELAJARAN UNTUK INSPEKSI PCB BERBASIS TEMPLATE DENGAN KLASIFIKASI ROI SELEKTIF

Bernadectus Yudi Dwiandiyanta^{1*}, Silvester Tena²

¹⁾ Departement Informatika, Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia

²⁾ Fakultas Sains dan Teknik, Universitas Nusa Cendana, Kupang, Indonesia

Email: yudi.dwiandiyanta@uajy.ac.id, siltena@staf.undana.ac.id

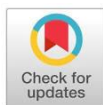
Info Artikel

Histori Artikel:

Diterima Agu 29, 2026

Direvisi Sep 24, 2026

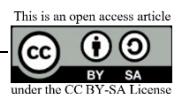
Disetujui Sep 28, 2026



ABSTRACT

Printed circuit board (PCB) defect inspection is difficult because many defects are small and visually subtle. This study proposes a template-based inspection method combining image differencing, learning-free local candidate refinement, and selective ROI classification. Initial candidates are obtained from aligned test and template images, then locally refined using grayscale residual information without adding new candidates. Refined ROIs are represented by test, template, and residual patches and classified using MobileNetV3-Small. On DeepPCB, candidate recall at IoU 0.20 increases from 0.3389 to 0.9586 and at IoU 0.30 from 0.1153 to 0.6344. ROI classification achieves 0.9874 accuracy and 0.9855 macro F1. With a rejection threshold of 0.90, coverage remains 0.9762 with macro F1 of 0.9927. The method improves coarse defect localization, although strict bounding-box localization remains limited.

Keywords: PCB defect inspection; template-based inspection; candidate refinement; image differencing; MobileNetV3-Small; selective prediction



ABSTRAK

Inspeksi cacat pada printed circuit board (PCB) sulit dilakukan karena banyak cacat berukuran kecil dan memiliki perbedaan visual yang tipis. Penelitian ini mengusulkan metode inspeksi berbasis template yang menggabungkan image differencing, perbaikan lokal kandidat tanpa pembelajaran, dan klasifikasi ROI secara selektif. Kandidat awal diperoleh dari citra uji dan template yang telah disejajarkan, kemudian diperbaiki secara lokal menggunakan residual grayscale tanpa menambah kandidat baru. ROI hasil perbaikan direpresentasikan oleh patch citra uji, template, dan residual, lalu diklasifikasikan menggunakan MobileNetV3-Small. Pada DeepPCB, recall kandidat pada IoU 0,20 meningkat dari 0,3389 menjadi 0,9586 dan pada IoU 0,30 dari 0,1153 menjadi 0,6344. Klasifikasi ROI menghasilkan accuracy 0,9874 dan Macro F1 0,9855. Pada threshold 0,90, coverage tetap 0,9762 dengan Macro F1 0,9927. Metode ini meningkatkan lokalisasi kasar kandidat cacat, meskipun lokalisasi bounding box yang ketat masih terbatas.

Kata Kunci: inspeksi cacat PCB; inspeksi berbasis template; perbaikan kandidat; image differencing; MobileNetV3-Small; selective prediction

Penulis Korespondensi:

Bernadectus Yudi Dwiandiyanta,
Program Studi Informatika, Fakultas Teknologi Industri,
Universitas Atma Jaya Yogyakarta,
Jl. Babarsari 43 Yogyakarta.
Email: yudi.dwiandiyanta@uajy.ac.id



1. PENDAHULUAN

Printed circuit board (PCB) merupakan bagian penting pada hampir seluruh perangkat elektronik modern karena menyediakan jalur interkoneksi listrik sekaligus dukungan mekanis bagi berbagai komponen. Meningkatnya kepadatan jalur, ukuran komponen yang semakin kecil, dan tuntutan reliabilitas perangkat membuat kualitas PCB menjadi bagian penting dalam proses manufaktur. Cacat yang terlihat kecil pada jalur konduktor atau permukaan PCB dapat menyebabkan gangguan fungsi setelah perakitan, sehingga inspeksi tidak hanya berkaitan dengan kualitas visual tetapi juga dengan reliabilitas produk elektronik secara keseluruhan [1].

Pemeriksaan manual masih mungkin diterapkan pada produksi skala kecil, tetapi semakin sulit dipertahankan ketika jumlah PCB meningkat dan ukuran cacat semakin kecil. Variasi intensitas, tekstur jalur konduktor, serta kemiripan pola antara area normal dan cacat membuat beberapa kerusakan sulit dibedakan secara konsisten oleh operator maupun sistem otomatis sederhana [2]. Kinerja sistem inspeksi juga dipengaruhi oleh karakteristik *dataset* dan konfigurasi proses pembelajaran, sehingga model dengan arsitektur serupa belum tentu menghasilkan performa yang sama ketika kondisi pelatihan dan data berubah [3].

Sebelum penggunaan *deep learning* meluas, inspeksi PCB banyak memanfaatkan karakter referensial yang diperoleh dari proses produksi. Citra PCB yang diperiksa dibandingkan dengan pola PCB normal melalui registrasi, segmentasi, *image subtraction*, operasi morfologi, dan analisis komponen untuk menentukan lokasi yang mungkin mengalami kerusakan. Pendekatan ini menarik karena lokasi yang berbeda dapat ditelusuri secara langsung dan tidak seluruh proses lokalisasinya harus dipelajari dari data [4].

Keterbatasan metode konvensional terutama muncul ketika perubahan pencahayaan, ketidaksempurnaan registrasi, atau *noise* menghasilkan perbedaan antara citra referensi dan citra uji yang sebenarnya bukan merupakan cacat. Penelitian berbasis pengolahan citra masih membutuhkan tahap segmentasi dan penapisan yang cukup hati-hati agar area berbeda yang dihasilkan tidak langsung dianggap sebagai cacat [5].

Perkembangan *deep learning* kemudian menggeser arah penelitian menuju sistem yang mempelajari karakteristik cacat secara langsung dari data. Namun, pendekatan ini membawa masalah lain

berupa kebutuhan terhadap data beranotasi yang cukup banyak, sementara proses anotasi citra inspeksi industri tidak selalu murah. Penggunaan *semi-supervised learning* telah diteliti untuk memanfaatkan kombinasi data PCB berlabel dan tidak berlabel agar ketergantungan terhadap anotasi manual dapat dikurangi [6].

Sebagian besar pengembangan penelitian diarahkan pada kemampuan menangkap cacat berukuran kecil. DDTR menggabungkan CNN dengan *Swin Transformer* agar informasi lokal tetap dipertahankan sementara hubungan spasial yang lebih luas dapat dipelajari oleh model [7]. DCR-YOLO menggunakan struktur yang lebih ringan serta mekanisme fusi fitur untuk meningkatkan sensitivitas terhadap target kecil tanpa mengabaikan kebutuhan pemrosesan yang cepat [8]. YOLO-MBBi mengambil arah serupa dengan memodifikasi YOLOv5 melalui MBConv, *attention mechanism*, serta fusi fitur multiskala untuk memperbaiki deteksi cacat permukaan PCB [9].

Modifikasi arsitektur YOLO terus berkembang karena terdapat kebutuhan untuk menyeimbangkan akurasi, kecepatan, dan kompleksitas jaringan. CDI-YOLO mengembangkan YOLOv7-tiny dengan *coordinate attention* dan perubahan struktur jaringan untuk meningkatkan performa pada cacat PCB yang relatif kecil [10]. YOLO-RRL juga dirancang untuk mempertahankan kemampuan deteksi objek kecil dengan memperbaiki *feature pyramid*, menggunakan *dynamic upsampling*, serta menyederhanakan *detection head* [11].

Masalah hilangnya informasi spasial selama proses *downsampling* juga menjadi perhatian. YOLOv5_ES menggunakan SPD-Conv dan mekanisme perhatian multiskala agar informasi cacat yang berukuran kecil tidak terlalu cepat hilang ketika fitur diproses pada lapisan yang semakin dalam [12]. Sementara itu, pendekatan *multistage* menunjukkan bahwa penggabungan informasi semantik dan konteks pada beberapa tahap juga dapat meningkatkan kemampuan jaringan untuk menemukan cacat PCB yang secara visual sulit dibedakan dari latar belakang [13].

Tidak semua penelitian memilih memperbesar jaringan. LGCL-CenterNet justru menggabungkan informasi lokal dan global dalam rancangan yang relatif ringan untuk mempertahankan kemampuan deteksi sekaligus menekan kebutuhan komputasi [14]. MDD-DETR menggunakan pendekatan berbasis *Transformer* dan memberikan perhatian khusus pada cacat kecil seperti *mousebite* dan *spur*, yang menunjukkan bahwa lokalisasi cacat minor

masih menjadi persoalan bahkan ketika arsitektur detektor yang lebih modern digunakan [15].

Upaya memperoleh model yang lebih efisien semakin terlihat pada penelitian berikutnya. SEPDNet menunjukkan bahwa struktur jaringan yang disesuaikan dengan distribusi ukuran cacat dapat menggunakan parameter yang lebih sedikit dibandingkan model YOLO pembandingan tetapi masih memberikan peningkatan pada F1 dan mAP [16]. Di sisi lain, pengembangan YOLOv11-PCB tetap menggunakan strategi modifikasi arsitektur dengan menggabungkan *efficient multi-scale attention*, CARAFE, serta EIoU untuk meningkatkan representasi fitur dan regresi *bounding box* [17].

Pendekatan berbasis DETR juga terus dikembangkan. HSA-RTDETR menggunakan *attention mechanism* berbasis skala untuk meningkatkan kemampuan model dalam menangani variasi ukuran cacat PCB [18]. SRN_Net menekankan kombinasi informasi lokal dan global untuk mengurangi kegagalan deteksi pada area cacat yang kecil dan berada pada latar belakang PCB yang kompleks [19].

Perbaikan pada representasi fitur multiskala juga masih menjadi arah penelitian yang aktif. MS-YOLO memperkuat informasi tepi serta mengubah mekanisme *downsampling* agar karakter spasial dari cacat kecil dapat dipertahankan lebih baik [20]. PD-YOLOv8 menggunakan *attention mechanism* dan modifikasi jaringan untuk meningkatkan sensitivitas terhadap objek kecil, kembali menunjukkan bahwa sebagian besar peningkatan performa masih diperoleh dengan melakukan perubahan langsung pada arsitektur detektor [21].

Pengembangan tersebut belum berhenti. Arsitektur hibrid CNN-Transformer dengan mekanisme fusi fitur adaptif telah digunakan untuk memperkuat representasi cacat kecil dan berkontras rendah [22]. Modifikasi terhadap YOLOv10 juga menunjukkan bahwa peningkatan ekstraksi fitur dan fusi informasi masih menjadi strategi yang digunakan untuk menghadapi lingkungan inspeksi PCB yang kompleks [23].

Berdasarkan penelitian di atas terlihat bahwa perkembangan deteksi cacat PCB sebagian besar bergerak pada arah yang sama. Peningkatan performa banyak diperoleh melalui perubahan *backbone*, *attention mechanism*, fusi fitur, *feature pyramid*, *detection head*, atau regresi *bounding box*. Strategi ini memang dapat menghasilkan performa deteksi yang tinggi, tetapi sekaligus membuat

lokalisasi semakin bergantung pada model pembelajaran.

Karakteristik inspeksi PCB sebenarnya menyediakan kemungkinan lain. Dalam sistem produksi tertentu tersedia citra referensi bebas cacat yang mempunyai struktur geometris sangat dekat dengan citra yang sedang diperiksa. Informasi tersebut memungkinkan area yang mengalami perubahan diperoleh terlebih dahulu melalui perbandingan citra. Dengan demikian, keseluruhan proses lokalisasi tidak selalu harus diserahkan kepada jaringan saraf.

Permasalahannya hasil *image differencing* dan analisis komponen tidak selalu menghasilkan kotak kandidat yang mengikuti lokasi cacat dengan baik. Kandidat dapat sedikit bergeser, terlalu besar, atau mengikuti residual yang terbentuk akibat perbedaan lokal antara citra uji dan *template*. Salah satu cara sederhana untuk memperbesar peluang menemukan cacat adalah menghasilkan lebih banyak kandidat, tetapi pendekatan tersebut juga meningkatkan jumlah area yang harus diperiksa oleh *classifier*.

Penelitian ini mengambil pendekatan yang berbeda. Citra uji dan citra *template* terlebih dahulu dibandingkan untuk memperoleh kandidat awal lokasi cacat. Setiap kandidat kemudian diperbaiki menggunakan perbaikan lokal kandidat (*learning-free local candidate refinement*) yang memanfaatkan informasi residual pada area di sekitarnya. Tahap ini tidak menggunakan proses pembelajaran dan tidak menghasilkan kandidat tambahan. Jumlah kandidat sebelum dan sesudah *refinement* tetap sama. Perubahan hanya dilakukan pada posisi dan ukuran kotak kandidat.

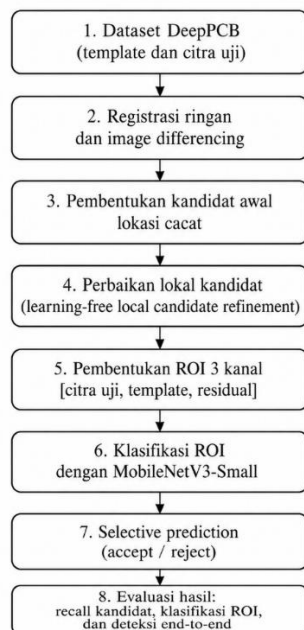
Kandidat hasil perbaikan selanjutnya direpresentasikan sebagai ROI tiga kanal yang terdiri atas patch citra uji, *patch template*, dan residual keduanya. ROI tersebut diklasifikasikan menggunakan MobileNetV3-Small menjadi enam jenis cacat PCB dan satu kelas *background*. Mekanisme *selective prediction* juga digunakan untuk memungkinkan sistem menolak kandidat dengan tingkat keyakinan rendah daripada memaksa masuk ke salah satu kelas.

Kontribusi penelitian ini terletak pada tiga bagian. Pertama, dikembangkan perbaikan lokal kandidat cacat yang bersifat *learning-free* untuk meningkatkan cakupan spasial kandidat berbasis citra referensi tanpa menambah jumlah kandidat. Kedua, digunakan representasi ROI yang secara eksplisit menggabungkan informasi citra uji, *template*, dan residual untuk proses klasifikasi. Ketiga, evaluasi dilakukan secara terpisah pada

tingkat kandidat, klasifikasi ROI, *selective prediction*, dan deteksi *end-to-end*, sehingga peningkatan pada satu tahap tidak langsung diinterpretasikan sebagai peningkatan keseluruhan sistem.

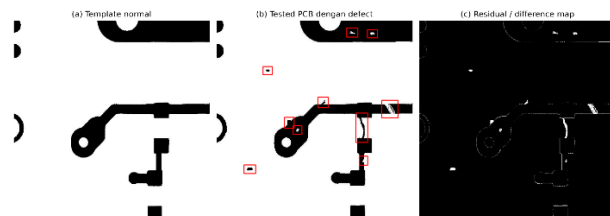
2. METODE PENELITIAN

2.1 Dataset dan Alur Penelitian



Gambar 1. Langkah penelitian

Penelitian menggunakan *dataset* DeepPCB yang terdiri atas 1.500 pasangan citra PCB berukuran 640×640 piksel. Langkah penelitian secara umum dapat dilihat pada Gambar 1. Setiap pasangan terdiri atas citra referensi bebas cacat (*template image*) dan citra uji (*tested image*) yang telah disejajarkan, serta anotasi enam jenis cacat, yaitu *open*, *short*, *mousebite*, *spur*, *spurious copper*, dan *pin-hole*. *Dataset* menyediakan 1.000 pasangan citra untuk pengembangan dan 500 pasangan untuk pengujian.



Gambar 2. Contoh pasangan citra DeepPCB: (a) citra *template* bebas cacat, (b) citra uji yang mengandung cacat, dan (c) citra residual

Contoh pasangan citra *template*, citra uji yang mengandung cacat, dan citra residual ditunjukkan pada Gambar 2. Secara visual, perbedaan antara keduanya biasanya hanya muncul pada area kecil. Karena posisi pola PCB pada kedua citra hampir

sama, citra *template* dapat digunakan sebagai referensi untuk mencari perubahan lokal pada citra uji. *Dataset* DeepPCB sendiri memang dirancang dengan pasangan semacam ini dan menggunakan registrasi serta *thresholding* sebagai bagian penting dari proses inspeksi.

Alur penelitian terdiri atas empat tahap utama: pembentukan kandidat awal dari perbedaan antara citra uji dan *template*, perbaikan posisi kandidat secara lokal, pembentukan ROI tiga kanal, dan klasifikasi cacat menggunakan MobileNetV3-Small.

2.2 Pembentukan dan Refinement Kandidat

Citra uji dan *template* terlebih dahulu diselaraskan menggunakan registrasi ringan berbasis *Enhanced Correlation Coefficient* (ECC). Kedua citra kemudian dibinarisasi menggunakan metode Otsu, dan peta perbedaan dihitung sebagai:

$$D_b = |B_t - B_r|$$

dengan B_t dan B_r masing-masing merupakan citra biner uji dan *template*. Operasi morfologi dan *connected component analysis* digunakan untuk memperoleh kandidat awal.

Kontribusi utama penelitian berada pada tahap perbaikan lokal kandidat cacat (*learning-free local candidate refinement*), yaitu memperbaiki posisi dan ukuran kotak kandidat tanpa menambah jumlah kandidat. Untuk setiap kandidat awal, dibuat jendela lokal di sekitarnya dan residual *grayscale* dihitung dengan

$$D_g(x, y) = |I_t(x, y) - I_r(x, y)|$$

Residual lokal diproses menggunakan Gaussian *filtering*, *threshold* Otsu, *threshold* berbasis persentil, dan operasi morfologi. Komponen lokal yang paling sesuai kemudian digunakan untuk memperbaiki posisi dan ukuran *bounding box*. Jika hasil *refinement* terlalu kecil atau tidak stabil, kandidat awal tetap dipertahankan. Dengan strategi ini, *refinement* hanya memperbaiki koordinat kandidat yang sudah tersedia dan tidak menambah jumlah kandidat. Parameter utama yang digunakan pada tahap perbaikan lokal kandidat dirangkum pada Tabel 1.

Tabel 1. Parameter utama perbaikan lokal kandidat

Parameter	Nilai
Minimum <i>component area</i>	12 piksel
Local margin	10 piksel
Gaussian <i>kernel</i>	(5 x 5)
<i>Percentile threshold</i>	85
<i>Refinement padding</i>	2 piksel
Minimum <i>local component</i>	8 piksel

2.3 Pembentukan ROI dan Klasifikasi

Setiap kandidat diubah menjadi ROI berukuran 96x96 piksel. *Input classifier* terdiri atas tiga kanal:

$$X = [P_t, P_r, P_d]$$

dengan P_t adalah *patch* citra uji, P_r *patch template*, dan P_d residual antara keduanya. Representasi ini memungkinkan *classifier* melihat pola aktual, pola referensi, dan perubahan lokal pada posisi yang sama. ROI positif diperoleh dari anotasi dan kandidat yang beririsan dengan *ground truth*, sedangkan ROI negatif dipilih dari kandidat non-cacat dengan respons residual relatif tinggi sebagai *hard-negative samples*.

Classifier menggunakan MobileNetV3-Small dengan bobot awal ImageNet dan tujuh kelas keluaran, yaitu enam kelas cacat dan satu kelas *background*. *Optimizer* yang digunakan adalah AdamW dengan *learning rate* 10^{-3} , *weight decay* 10^{-4} , *batch size* 64, dan maksimum delapan *epoch*.

Selain klasifikasi biasa, digunakan *selective prediction*. Prediksi hanya diterima apabila *confidence* tertinggi memenuhi

$$f(x) = \begin{cases} \hat{y}, & \text{jika } \max_c p(y = c | x) \geq \tau \\ \text{reject}, & \text{jika } \max_c p(y = c | x) < \tau \end{cases}$$

Prediksi dengan *confidence* di bawah *threshold* akan ditolak dan diarahkan untuk pemeriksaan lebih lanjut.

2.4 Evaluasi

Evaluasi dilakukan pada tiga level. Pertama, kualitas kandidat diukur menggunakan *recall* kandidat pada IoU 0,20, 0,30, dan 0,50. Kedua, kemampuan *classifier* ROI dievaluasi menggunakan *Accuracy*, *Balanced Accuracy*, *Precision*, *Recall*, dan *Macro F1*. Ketiga, seluruh *pipeline* diuji secara *end-to-end* menggunakan *Precision*, *Recall*, *F1*, *Average Precision*, dan *mAP* pada IoU 0,33 dan 0,50. Untuk setiap *ground-truth box*, dicari kandidat dengan nilai IoU tertinggi, dan cacat dinyatakan tercakup apabila nilai IoU tersebut memenuhi ambang yang ditentukan. IoU 0,33 digunakan sebagai kriteria lokalisasi yang lebih toleran untuk mengevaluasi kemampuan lokalisasi kasar kandidat, sedangkan IoU 0,50 digunakan sebagai kriteria lokalisasi yang lebih ketat.

Implementasi dilakukan dengan menggunakan Python dengan library PyTorch, dan OpenCV. Pelatihan *classifier* dilakukan menggunakan GPU NVIDIA L40S, sedangkan registrasi, *image differencing*, operasi morfologi, dan perbaikan lokal kandidat dijalankan pada CPU.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

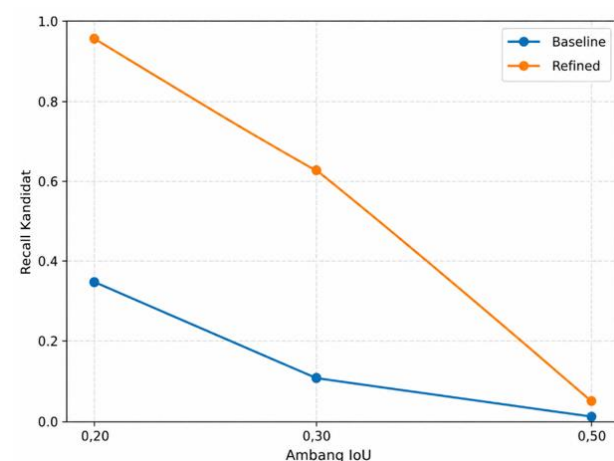
3.1.1 Pengaruh perbaikan lokal kandidat terhadap Kualitas Kandidat Area Cacat

Eksperimen pertama mengevaluasi efektivitas perbaikan lokal kandidat dalam memperbaiki lokasi kandidat cacat yang dihasilkan oleh tahap *template differencing*. Evaluasi dilakukan pada tiga ambang *Intersection over Union* (IoU), yaitu 0,20, 0,30, dan 0,50. Kandidat area cacat sebelum *refinement* digunakan sebagai *baseline*, sedangkan kandidat area cacat setelah perbaikan lokal disebut sebagai kandidat hasil *refinement*. Hasil perbandingan *recall* kandidat sebelum dan sesudah *refinement* disajikan pada [Tabel 2](#) dan divisualisasikan pada [Gambar 3](#).

Tabel 2. Perbandingan *recall* kandidat sebelum dan sesudah *refinement*

IoU	Baseline	Refined	Peningkatan absolut
0,20	0,3389	0,9586	+0,6197
0,30	0,1153	0,6344	+0,5191
0,50	0,0070	0,0500	+0,0430

Peningkatan paling nyata terjadi pada IoU rendah hingga menengah. Pada IoU 0,20, *recall* kandidat meningkat dari 0,3389 menjadi 0,9586. Artinya, setelah *refinement* sekitar 95,86% cacat mempunyai sedikitnya satu kandidat yang cukup beririsan dengan *ground-truth box*. Pada IoU 0,30, peningkatan masih cukup besar, dari 0,1153 menjadi 0,6344.

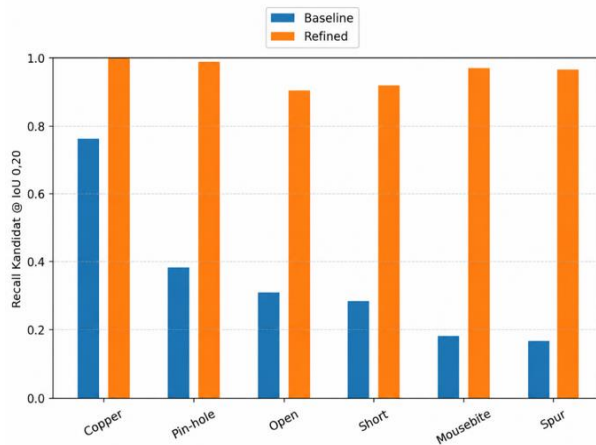


Gambar 3. *Recall* kandidat sebelum dan sesudah *local refinement* pada beberapa ambang IoU

Baseline dan *refined* menghasilkan rata-rata 14,254 kandidat per citra, dengan median delapan kandidat dan maksimum 173 kandidat pada satu citra. Perilaku *refinement* kemudian dianalisis pada masing-masing jenis cacat. Hasilnya disajikan pada [Tabel 3](#) dan [Gambar 4](#).

Tabel 3. Recall kandidat per kelas pada IoU 0,20

Kelas cacat	Baseline	Refined	Peningkatan
Copper	0,7651	1,0000	+0,2349
Pin-hole	0,3809	0,9915	+0,6106
Open	0,3096	0,9059	+0,5964
Short	0,2866	0,9226	+0,6360
Mousebite	0,1826	0,9761	+0,7935
Spur	0,1698	0,9731	+0,8033

**Gambar 4.** Perbandingan recall kandidat baseline dan refined pada masing-masing kelas cacat

3.1.2 Kinerja Klasifikasi ROI

Setelah kandidat diperoleh, ROI diklasifikasikan menjadi enam kelas cacat dan satu kelas *background*. Evaluasi dilakukan terhadap 4.587 ROI. Kinerja keseluruhan *classifier* ROI ditunjukkan pada [Tabel 4](#).

Tabel 4. Kinerja keseluruhan *classifier* ROI

Metrik	Nilai
Accuracy	0,9874
Balanced Accuracy	0,9859
Macro F1	0,9855
Jumlah ROI evaluasi	4.587

Classifier menghasilkan *Accuracy* sebesar 0,9874, *Balanced Accuracy* sebesar 0,9859, dan *Macro F1* sebesar 0,9855. Hasil klasifikasi untuk masing-masing kelas ditampilkan pada [Tabel 5](#).

Tabel 5. Hasil klasifikasi pada masing-masing kelas

Kelas	Precision	Recall	F1-score	Support
Background	0,9434	0,9687	0,9559	447
Open	0,9964	0,9927	0,9945	826
Short	0,9877	0,9860	0,9868	570
Mousebite	0,9862	0,9944	0,9903	717
Spur	0,9911	0,9823	0,9867	565
Copper	0,9926	0,9926	0,9926	816
Pin-hole	0,9984	0,9845	0,9914	646

Performa keenam kelas cacat relatif seragam. F1-score tertinggi diperoleh kelas open sebesar 0,9945, disusul copper sebesar 0,9926 dan pin-hole sebesar 0,9914. Bahkan spur, yang relatif sulit pada tahap kandidat awal, masih menghasilkan F1-score 0,9867 ketika ROI yang relevan telah tersedia.

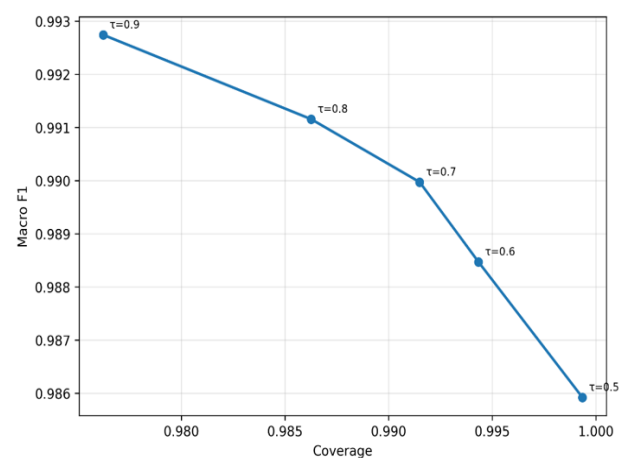
3.1.3 Pengaruh Reject Threshold terhadap Coverage dan Kinerja Klasifikasi

Sistem selanjutnya dievaluasi menggunakan mekanisme *selective prediction*. Pada skenario ini, prediksi dengan *confidence* di bawah ambang tertentu dapat ditolak sehingga *classifier* tidak harus memberikan keputusan pada setiap ROI. Pengaruh perubahan *reject threshold* terhadap *coverage* dan performa klasifikasi ditunjukkan pada [Tabel 6](#) dan [Gambar 5](#).

[Gambar 5](#) menunjukkan pola yang konsisten: ketika *threshold* dinaikkan, *coverage* berkurang secara perlahan, tetapi kualitas prediksi pada sampel yang tetap diterima meningkat. Pada *threshold* 0,90, *coverage* masih mencapai 0,9762, sementara *Balanced Accuracy* meningkat menjadi 0,9930 dan *Macro F1* menjadi 0,9927.

Tabel 6. Pengaruh *reject threshold* terhadap *coverage* dan performa klasifikasi

Threshold	Coverage	Balanced Accuracy	Macro F1
0,50	0,9993	0,9865	0,9859
0,60	0,9943	0,9890	0,9885
0,70	0,9915	0,9903	0,9900
0,80	0,9863	0,9915	0,9912
0,90	0,9762	0,9930	0,9927

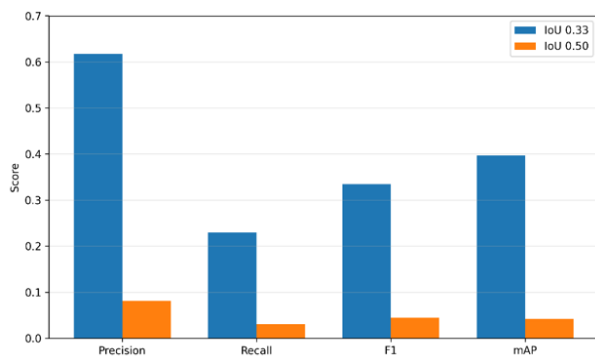
**Gambar 5.** Hubungan *coverage* dan *Macro F1* pada beberapa *reject threshold*

3.1.4 Evaluasi End-to-End

Evaluasi berikutnya dilakukan secara *end-to-end*. Pada tahap ini, lokasi ROI tidak diambil dari *ground truth*, tetapi sepenuhnya berasal dari kandidat refined yang kemudian diklasifikasikan. Hasil evaluasi *end-to-end* pada IoU 0,33 dan 0,50 disajikan pada [Tabel 7](#) dan [Gambar 6](#).

Tabel 7. Hasil evaluasi *end-to-end*

IoU	Precision	Recall	F1	mAP
0,33	0,6170	0,2293	0,3343	0,3970
0,50	0,0814	0,0303	0,0441	0,0417



Gambar 6. Precision, recall, F1, dan mAP pada evaluasi end-to-end

Hasil pada IoU 0,33 menunjukkan precision sebesar 0,6170 dan recall 0,2293. Kondisi tersebut menyebabkan F1 end-to-end hanya sebesar 0,3343 dan mAP sebesar 0,3970.

Pada IoU 0,50, F1 menurun menjadi 0,0441 dan mAP menjadi 0,0417. [Gambar 6](#) memperlihatkan perbedaan ini dengan cukup jelas. Penurunan tersebut sejalan dengan evaluasi kandidat pada [Gambar 3](#), dengan recall kandidat pada IoU 0,50 juga masih rendah.

3.1.5 Pengaruh Confidence Threshold, NMS, dan Score Fusion

Untuk memastikan bahwa rendahnya performa end-to-end bukan hanya diakibatkan konfigurasi keputusan akhir, dilakukan parameter sweep terhadap confidence threshold, NMS, dan cara pembentukan skor deteksi. Pengujian menggunakan confidence classifier maupun fused score antara confidence classifier dan skor kandidat.

Konfigurasi terbaik pada IoU 0,33 menggunakan fused score, NMS 0,30, dan confidence threshold pada rentang 0,10–0,40. Konfigurasi tersebut menghasilkan precision 0,6106, recall 0,2347, F1 0,3391, dan mAP 0,4077. Perbandingan antara konfigurasi awal dan konfigurasi terbaik hasil parameter sweep dirangkum pada .

Tabel 8. Perbandingan performa sebelum dan setelah parameter sweep

IoU	Kondisi	Precision	Recall	F1	mAP
0,33	Konfigurasi awal	0,6170	0,2293	0,3343	0,3970
0,33	Konfigurasi terbaik	0,6106	0,2347	0,3391	0,4077
0,50	Konfigurasi awal	0,0814	0,0303	0,0441	0,0417

Peningkatan yang diperoleh setelah sweep relatif kecil. Pada IoU 0,33, F1 meningkat dari 0,3343 menjadi 0,3391 dan mAP dari 0,3970 menjadi 0,4077. Pada IoU 0,50, perubahan threshold dan NMS juga tidak menghasilkan kenaikan F1 yang berarti. Beberapa konfigurasi fused score mampu

menghasilkan mAP sekitar 0,0533, tetapi F1 tetap berada pada kisaran 0,044.

3.2 Pembahasan Hasil Penelitian

Hasil penelitian menunjukkan bahwa perbaikan lokal kandidat efektif meningkatkan cakupan spasial kandidat cacat, terutama pada IoU rendah hingga menengah. Recall pada IoU 0,20 meningkat dari 0,3389 menjadi 0,9586 dan pada IoU 0,30 dari 0,1153 menjadi 0,6344. Peningkatan tersebut terjadi tanpa perubahan jumlah kandidat, sehingga kenaikan recall berasal dari perbaikan posisi dan ukuran kandidat yang telah tersedia, bukan dari penambahan jumlah kandidat.

Peningkatan paling besar terjadi pada kelas spur dan mousebite. Kedua kelas tersebut memiliki recall baseline paling rendah, tetapi setelah refinement mencapai masing-masing 0,9731 dan 0,9761. Hasil ini menunjukkan bahwa pemrosesan residual secara lokal membantu ketika perubahan visual cacat berukuran kecil belum membentuk komponen global yang sesuai dengan anotasi. Hasil ini sesuai dengan penelitian HSA-RTDETR yang menekankan bahwa lokalisasi cacat kecil pada PCB sangat sensitif terhadap pergeseran posisi bounding box, sehingga mekanisme regresi khusus diperlukan untuk mempertahankan akurasi pada objek berukuran kecil [18].

Meskipun demikian, recall pada IoU 0,50 masih rendah, yaitu 0,0500 setelah refinement. Hal ini menunjukkan bahwa metode learning-free yang diusulkan lebih efektif untuk lokalisasi kasar daripada lokalisasi presisi tinggi. Kondisi tersebut dapat dipahami karena pendekatan ini tidak menggunakan mekanisme bounding-box regression berbasis pembelajaran seperti yang diterapkan pada detektor end-to-end modern.

Pada tahap klasifikasi, representasi ROI tiga kanal yang menggabungkan citra uji, template, dan residual menghasilkan Macro F1 sebesar 0,9855. Hasil ini menunjukkan bahwa setelah kandidat yang relevan tersedia, informasi referensial antara kondisi aktual dan kondisi normal cukup kuat untuk membedakan enam kelas cacat. Hasil ini sejalan dengan kecenderungan penelitian sebelumnya yang menyesuaikan representasi fitur dengan karakteristik cacat kecil, seperti pada SEPDNet yang menunjukkan bahwa rancangan yang disesuaikan dengan distribusi ukuran cacat dapat meningkatkan performa tanpa harus memperbesar kompleksitas jaringan secara berlebihan [16].

Nilai F1 kelas background yang lebih rendah dibandingkan kelas cacat dapat dikaitkan dengan penggunaan hard-negative sampling. ROI negatif

berasal dari kandidat dengan respons residual relatif tinggi, sehingga secara visual lebih menyerupai cacat dibandingkan *background* acak. Sementara itu, mekanisme *selective prediction* menunjukkan bahwa sebagian kecil ROI dengan *confidence* rendah dapat ditahan tanpa mengurangi *coverage* secara besar. Pada *threshold* 0,90, *coverage* masih sebesar 0,9762 dengan *Macro F1* 0,9927.

Pemisahan evaluasi pada tingkat kandidat, klasifikasi ROI, dan *end-to-end* memperlihatkan bahwa performa tinggi pada satu tahap tidak dapat langsung dianggap sebagai performa keseluruhan sistem. *Macro F1* klasifikasi ROI mencapai 0,9855, sedangkan *F1 end-to-end* pada IoU 0,33 hanya sebesar 0,3343. Hasil ini menunjukkan bahwa *bottleneck* utama sistem berada pada ketepatan lokalisasi, bukan pada kemampuan membedakan kelas setelah ROI tersedia.

Posisi metode ini terlihat lebih jelas ketika dibandingkan dengan detektor *end-to-end* pada DeepPCB. YOLOv11-PCB dilaporkan mencapai *mAP@0.5* sebesar 98,9% dan MS-YOLO mencapai *mAP@0.5* sebesar 97,7% [17], [20]. Nilai tersebut jauh lebih tinggi dibandingkan *mAP* penelitian ini pada IoU 0,50. Namun, kedua metode tersebut mempelajari ekstraksi fitur dan regresi *bounding box* secara langsung, sedangkan penelitian ini menggunakan lokalisasi berbasis *template* yang tidak memerlukan proses pembelajaran pada tahap pembentukan dan perbaikan kandidat.

Dengan demikian, hasil penelitian ini tidak menunjukkan keunggulan sebagai detektor *end-to-end*, tetapi memperlihatkan adanya *trade-off* antara kesederhanaan lokalisasi berbasis referensi dan presisi *bounding box*. Parameter *sweep* yang hanya meningkatkan *mAP* dari 0,3970 menjadi 0,4077 pada IoU 0,33 juga memperkuat bahwa keterbatasan utama bukan berada pada *threshold* atau NMS, melainkan pada kualitas spasial kandidat. Secara keseluruhan, metode yang dikembangkan dapat diposisikan sebagai *template-guided coarse-to-fine candidate localization and verification framework* yang menonjolkan perbaikan kandidat tanpa menambah jumlah kandidat, pemanfaatan informasi referensial pada klasifikasi ROI, dan evaluasi bertingkat untuk menunjukkan kemampuan serta keterbatasan tiap tahap.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa perbaikan lokal kandidat cacat yang bersifat *learning-free* mampu meningkatkan kualitas lokalisasi kasar tanpa menambah jumlah kandidat, yang ditunjukkan oleh

kenaikan *recall* kandidat dari 0,3389 menjadi 0,9586 pada IoU 0,20 dan dari 0,1153 menjadi 0,6344 pada IoU 0,30. Penggunaan representasi ROI yang menggabungkan citra uji, *template*, dan residual terbukti efektif untuk membedakan jenis cacat setelah kandidat tersedia, dengan *Macro F1* sebesar 0,9855, sedangkan mekanisme *selective prediction* dapat meningkatkan keandalan keputusan dengan tetap mempertahankan *coverage* yang tinggi. Evaluasi yang dilakukan secara terpisah pada tingkat kandidat, klasifikasi ROI, dan *end-to-end* menunjukkan bahwa kekuatan utama metode berada pada perbaikan cakupan kandidat dan verifikasi kelas, sementara ketepatan lokalisasi pada IoU yang lebih ketat masih menjadi faktor pembatas. Dengan demikian, kontribusi penelitian ini adalah menghasilkan mekanisme inspeksi berbasis *template* yang mampu memperbaiki kandidat cacat secara ringan dan tanpa pembelajaran pada tahap lokalisasi, memanfaatkan informasi referensial secara eksplisit dalam klasifikasi ROI, serta memberikan evaluasi bertingkat yang memperlihatkan secara jelas kemampuan dan batas setiap tahap sistem.

DAFTAR PUSTAKA

- [1] H. Guo, H. Zhao, Y. Zhao, and W. Liu, "PCB defect detection algorithm based on deep learning," *Optik (Stuttg)*, vol. 315, p. 1-13, 2024, doi:<http://dx.doi.org/10.1016/j.jleo.2024.172036>.
- [2] P. Chen and F. Xie, "A Machine Learning Approach for Automated Detection of Critical PCB Flaws in Optical Sensing Systems," *Photonics*, vol. 10, no. 9, 2023, doi:<http://dx.doi.org/10.3390/photonics10090984>.
- [3] J. H. Park, Y. S. Kim, H. Seo, and Y. J. Cho, "Analysis of Training Deep Learning Models for PCB Defect Detection," *Sensors*, vol. 23, no. 5, p. 2766, 2023/3, doi:<http://dx.doi.org/10.3390/s23052766>.
- [4] J. Onshaunjit and J. Srinonchat, "Algorithmic scheme for concurrent detection and classification of printed circuit board defects," *Comput. Mater. Contin.*, vol. 71, no. 1, p. 355-367, 2022, doi:<http://dx.doi.org/10.32604/cmc.2022.017698>.
- [5] X. Chen, Y. Wu, X. He, and W. Ming, "A Comprehensive Review of Deep Learning-Based PCB Defect Detection," *IEEE Access*, vol. 11, no. December, p. 139017-139038, 2023, doi:<http://dx.doi.org/10.1109/ACCESS.2023.3339561>.
- [6] T. T. A. Pham, D. K. T. Thoi, H. Choi, and S. Park, "Defect Detection in Printed Circuit Boards Using Semi-Supervised Learning," *Sensors*, vol. 23, no. 6, 2023, doi:<http://dx.doi.org/10.3390/s23063246>.

- [7] B. Feng and J. Cai, "PCB Defect Detection via Local Detail and Global Dependency Information," *Sensors*, vol. 23, no. 18, p. 1-18, 2023 2023, doi:<http://dx.doi.org/10.3390/s23187755>.
- [8] Y. Jiang, M. Cai, and D. Zhang, "Lightweight Network DCR-YOLO for Surface Defect Detection on Printed Circuit Boards," *Sensors*, vol. 23, no. 17, 2023 2023, doi:<http://dx.doi.org/10.3390/s23177310>.
- [9] B. Du, F. Wan, G. Lei, L. Xu, C. Xu, and Y. Xiong, "YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5," *Electron*, vol. 12, no. 13, p. 1-24, 2023 2023, doi:<http://dx.doi.org/10.3390/electronics12132821>.
- [10] G. Xiao, S. Hou, and H. Zhou, "PCB defect detection algorithm based on CDI-YOLO," *Sci. Rep.*, vol. 14, no. 1, p. 1-12, 2024 2024, doi:<http://dx.doi.org/10.1038/s41598-024-57491-3>.
- [11] T. Zhang, J. Zhang, P. Pan, and X. Zhang, "YOLO-RRL: A Lightweight Algorithm for PCB Surface Defect Detection," *Appl. Sci.*, vol. 14, no. 17, 2024 2024, doi:<http://dx.doi.org/10.3390/app14177460>.
- [12] Y. Gao, Z. Li, Y. Wang, and S. Zhu, "A Novel YOLOv5_ES based on lightweight small object detection head for PCB surface defect detection," *Sci. Rep.*, vol. 14, no. 1, p. 1-12, 2024 2024, doi:<http://dx.doi.org/10.1038/s41598-024-74368-7>.
- [13] T. Kiobya, J. Zhou, B. Maiseli, and M. Khan, "Attentive context and semantic enhancement mechanism for printed circuit board defect detection with two-stage and multi-stage object detectors," *Sci. Rep.*, vol. 14, no. 1, p. 1-18, 2024 2024, doi:<http://dx.doi.org/10.1038/s41598-024-69207-8>.
- [14] W. Chen, S. Meng, and X. Wang, "Local and Global Context-Enhanced Lightweight CenterNet for PCB Surface Defect Detection," *Sensors*, vol. 24, no. 14, p. 4729, 2024/7 2024, doi:<http://dx.doi.org/10.3390/s24144729>.
- [15] J. Peng, W. Fan, S. Lan, and D. Wang, "MDD-DETR: Lightweight Detection Algorithm for Printed Circuit Board Minor Defects," *Electron*, vol. 13, no. 22, 2024 2024, doi:<http://dx.doi.org/10.3390/electronics13224453>.
- [16] D. Lang and Z. Lv, "SEPDNet: simple and effective PCB surface defect detection method," *Sci. Rep.*, vol. 15, no. 1, p. 1-15, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-024-84859-2>.
- [17] Z. F. Elsharkawy, "Enhanced YOLOv11 framework for high precision defect detection in printed circuit boards," *Sci. Rep.*, vol. 15, no. 1, p. 1-17, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-025-27415-w>.
- [18] Y. Wang, "Enhanced PCB defect detection via HSA-RTDETR on RT-DETR," *Sci. Rep.*, vol. 15, no. 1, p. 1-19, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-025-11394-z>.
- [19] L. Zhu and R. Zhao, "A novel PCB surface defect detection method based on separated global context attention to guide residual context aggregation," *Sci. Rep.*, vol. 15, no. 1, p. 1-18, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-024-84961-5>.
- [20] W. Chen, X. Ke, and S. Meng, "Small defect detection in printed circuit boards based on the multiscale edge strengthening and an improved YOLOv10," *Sci. Rep.*, vol. 15, no. 1, p. 1-15, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-025-20387-x>.
- [21] L. Bai and W. H. Xu, "Improved printed circuit board defect detection scheme," *Sci. Rep.*, vol. 15, no. 1, p. 1-13, 2025 2025, doi:<http://dx.doi.org/10.1038/s41598-025-85245-2>.
- [22] Y. Yan, G. Chen, Y. Gu, and X. Peng, "Hybrid CNN-transformer with adaptive multi-scale fusion for tiny PCB defect identification," *Complex Intell. Syst.*, 2026 2026, doi:<http://dx.doi.org/10.1007/s40747-026-02326-z>.
- [23] Z. Shen, Y. Yao, L. Liu, Y. Cao, and L. Lu, "Defect detection method of printed circuit boards based on EDF-YOLOv10," *PLoS One*, vol. 21, no. 3, p. 1-19, doi:<http://dx.doi.org/10.1371/journal.pone.0343130>.