

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu terkait inspeksi visual Printed Circuit Board (PCB) telah banyak dikembangkan dengan memanfaatkan teknologi pengolahan citra digital (*image processing*). Pada penelitian yang ditulis oleh Singh, (2012) membahas tentang sistem inspeksi otomatis berbasis *image processing* untuk mendeteksi cacat pada PCB dengan tujuan mendeteksi komponen yang hilang, mengidentifikasi orientasi komponen yang salah dan mendeteksi kerusakan jalur seperti putus atau *short*. Penelitian ini bekerja dengan membandingkan PCB standar (*reference*) dengan PCB yang diuji (*test board*) sehingga hasil yang didapatkan yaitu status *Good and Not Good*.

Penelitian lain ditulis oleh (Punith Kumar et al., 2020) mengembangkan sistem inspeksi kualitas PCB secara otomatis menggunakan teknik *image processing* untuk menggantikan inspeksi manual yang lambat dan rentan kesalahan. Sistem ini menggunakan kamera digital untuk pengambilan citra PCB. Perancangan sistem dilakukan dengan tahapan akuisisi citra, pre-processing, ekstraksi fitur, serta penerapan algoritma deteksi cacat berbasis *thresholding*, *image subtraction* dan *color mapping*. Pengujian dilakukan pada 48 papan PCB nyata di industri, dengan waktu pemrosesan kurang dari 2 detik untuk setiap papan. Hasil pengujian menunjukkan sistem memiliki akurasi sekitar 85,7% dengan sensitivity 84% dan specificity 87% serta mampu menampilkan lokasi cacat melalui overlay visual sehingga memudahkan teknisi melakukan perbaikan. Pengembangan yang dapat dilakukan antara lain meningkatkan akurasi deteksi dengan penggunaan kamera yang memiliki resolusi lebih tinggi dengan integrasi sistem real-time.

Penelitian dari (Gunawan et al., 2016) merancang prototipe sistem deteksi cacat pada PCB berbasis pengolahan citra digital (*image processing*) untuk membantu proses inspeksi kualitas papan PCB. Teknik yang digunakan yaitu dengan memanfaatkan kamera sebagai alat akuisisi citra PCB, yang kemudian diolah dengan teknik pengolahan citra yaitu konversi grayscale, *filtering*, *thresholding*, serta membandingkan citra terhadap PCB referensi. Metode *template*

matching dan *image differencing* dengan pengujian yang dilakukan pada beberapa sampel PCB dengan kondisi normal dan cacat untuk melihat kemampuan sistem dalam mendeteksi perbedaan. Hasil pengujian menunjukkan bahwa sistem mampu mengidentifikasi lokasi cacat dan menandainya pada citra hasil proses, sehingga mempermudah proses evaluasi kualitas PCB. Akan tetapi penelitian tersebut hanya berfokus pada proses deteksi cacat PCB tanpa adanya penyimpanan data hasil inspeksi.

Selanjutnya Mira Safitri, (2024) dalam penelitiannya, pendekatan *image processing* hanya untuk mendeteksi dan menghitung jumlah lubang pada PCB secara otomatis dengan menggunakan metode *transformasi hough* sebagai penentuan pola tertentu dengan mendeteksi titik-titik dalam ruang parameter dan memanfaatkan garis, lingkaran dan elips dari suatu gambar. Perancangan pada penelitian ini meliputi *preprocessing grayscale*, *gaussian filter* dan *hough circle transform*. Hasil pengujian dengan evaluasi ketepatan jarak dengan pengambilan data memberikan persentase 95%. Penelitian tersebut memberikan masukan agar pengembangan selanjutnya difokuskan penambahan variasi citra PCB yang digunakan dalam pendeteksian.

Penelitian yang ditulis oleh (Julianto et al., 2022) relevan dilakukan pada pengembangan alat inspeksi kecacatan PCB berbasis *image processing* dengan tujuan menggantikan inspeksi visual manusia guna meningkatkan efisiensi. Sistem yang dikembangkan menggunakan kamera Webcam sebagai media akuisisi citra dan Raspberry Pi sebagai pengolahan utama yang menjalankan program berbasis *Python*. Metode yang digunakan pada penelitian tersebut adalah *deep learning*. Hasil pengujian menunjukkan bahwa algoritma hanya mampu mendeteksi lima jenis cacat pada PCB. Akan tetapi, sistem masih memiliki keterbatasan antara lain hanya mendeteksi cacat pada jenis PCB polos serta bergantung pada model *deep learning* yang memerlukan training dan dataset yang cukup besar. Selain itu, sistem belum dengan mekanisme inspeksi otomatis kamera control dan berulang secara konsisten dalam posisi pengambilan citra.

Berdasarkan berbagai penelitian tersebut, sistem inspeksi visual PCB berbasis *image processing* telah menunjukkan kemampuan yang baik dalam mendeteksi cacat PCB menggunakan berbagai metode seperti, *thresholding*, *image*

differencing, template matching, transformasi hough, hingga pendekatan *deep learning*. Sebagian besar penelitian berfokus pada peningkatan akurasi algoritma dan performa deteksi.

Namun, integrasi antara sistem deteksi dengan kendali mekanik dan antarmuka pengguna (GUI), serta penyimpanan hasil inspeksi dalam bentuk data pengujian dan citra dokumentasi masih terbatas.

Oleh karena itu, penelitian ini dikembangkan sistem inspeksi berbasis *image processing* dengan metode *thresholding* dan *operasi morfologi* dengan sistem kendali mekanik berbasis Arduino, menampilkan hasil inspeksi secara *real-time* melalui GUI, serta menyimpan hasil inspeksi data pengujian dan citra hasil Pass/Fail sebagai dokumentasi pengujian ke database.

Pada tabel 2.1 dapat dilihat untuk memperjelas penelitian-penelitian terdahulu yang telah dijelaskan diatas, sebagai berikut.

Tabel 2. 1 Penelitian Terdahulu

No	Penelitiann Terdahulu		Hasil
1	Judul	Image Processing Based Automatic Visual Inspection System for PCB	Sistem mampu mendeteksi kerusakan jalur seperti putus atau <i>short</i> dengan membandingkan PCB standar (<i>reference</i>) dengan PCB yang diuji (<i>test board</i>) sehingga hasil yang didapatkan yaitu status <i>Good</i> and <i>Not Good</i> .
	Penulis/Tahun	(Sanveer Singh & Manu Bharti, 2019)	
	Metodologi	Algoritma: <i>Template Matching</i>	
		Deteksi: Komponen hilang dan orientasi komponen yang salah dan jalur short.	
		Pengujian: CCD Camera	
2	Judul	Automated Quality Inspection of PCB Assembly Using Image Processing	Sistem memiliki akurasi sekitar 85,7% dengan sensitivity 84% dan specificity 87%, serta mampu menampilkan lokasi cacat melalui overlay visual dalam waktu pemrosesan
	Penulis/Tahun	(Punith Kumar et al., 2020).	
	Metodologi	Algoritma: <i>Thresholding & Image Subtraction</i>	

		Deteksi: <i>Missing component, short circuit, open circuit, misalignment solder defect.</i>	kurang dari 2 detik untuk setiap papan.
		Pengujian: Camera digital	
3	Judul	Pembuatan Prototipe Sistem Pendeteksi Cacat Jalur pada Printed Circuit Board	Sistem mampu mengidentifikasi lokasi cacat dan menandainya pada citra hasil proses dengan pengujian yang dilakukan pada beberapa sampel PCB.
	Penulis/Tahun	(Gunawan et al., 2016).	
	Metodologi	Algoritma: <i>Image Differencing & Template Matching</i>	
		Deteksi: Jalur tembaga atau short.	
		Pengujian: Webcam	
4	Judul	Penentuan Lokasi Titik Lubang Pada Printed Circuit Board (PCB) menggunakan metode Transformasi Hough	Sistem mampu mendeteksi titik-titik dalam ruang parameter dan memanfaatkan garis, lingkaran dan elips dari suatu gambar dengan akurasi ketepatan 95%.
	Penulis/Tahun	(Mira Safitri, 2024)	
	Metodologi	Algoritma: Transformasi <i>Hough</i>	
		Deteksi: Menghitung jumlah lubang pada PCB	
		Pengujian: camera beresolusi 6000x8000	
5	Judul	Pembuatan Alat Inspeksi Visual Jalur PCB Menggunakan Pengolahan Citra Untuk Kegiatan Praktikum Pengawatan Dan Teknologi PCB	Mampu mendeteksi lima jenis cacat pada PCB polos dengan training dan dataset yang cukup besar.
	Penulis/Tahun	(Julianto et al., 2022)	
	Metodologi	Algoritma: Pendekatan <i>Deep Learning</i>	

		Deteksi: <i>Short, Open Circuit, Missing Hole, Mouse Bite, dan Spur.</i>	
		Pengujian: Webcam + Raspberry Pi	
6	Judul	PCB plug-in solder joint defect detection method based on coordinated attention-guided information fusion.	Mampu mendeteksi cacat solder PCB menggunakan peningkatan pada arsitektur YOLOv3. Model yang diusulkan memperoleh akurasi deteksi sebesar 96,43%, sehingga mampu meningkatkan ketelitian deteksi cacat solder.
	Penulis/Tahun	(Chen et al., 2024)	
	Metodelogi	Algoritma: <i>Deep Learning</i>	

Berdasarkan tabel 2.1, dapat dilihat bahwa sebagian besar penelitian inspeksi visual PCB berfokus pada peningkatan algoritma deteksi, seperti *template matching*, perbandingan citra referensi, maupun metode berbasis pembelajaran mesin.

2.2 Landasan Teori

2.2.1 Inspeksi Visual Otomatis

Inspeksi visual adalah teknik untuk mendeteksi cacat dengan memanfaatkan *human vision* atau penglihatan manusia untuk memastikan bahwa produk berfungsi dengan baik dan sesuai dengan spesifikasi. Namun, metode ini memiliki kekurangan yang bergantung pada ketelitian manusia serta berpotensi dapat menimbulkan ketidakkonsistenan hasil inspeksi.

Fonseca et al., (2024) mengemukakan bahwa inspeksi visual otomatis dikembangkan untuk menggantikan metode manual melalui *machine vision* yang mampu mendeteksi cacat secara objektif dan konsisten. Sistem ini mengintegrasikan beberapa tahapan utama, yaitu akuisisi citra, pemrosesan citra, analisis fitur dan pengambilan keputusan. Secara umum, sistem inspeksi visual otomatis terdiri dari komponen pencahayaan, perangkat akuisisi citra yaitu kamera, unit pemrosesan data, serta algoritma deteksi yang bekerja secara terintegrasi.

Berdasarkan pendekatan yang digunakan, sistem inspeksi visual otomatis dapat dibagi menjadi dua kategori utama, yaitu pendekatan berbasis aturan (*rule-based*) dan pendekatan berbasis pembelajaran mesin (*machine learning*) (Motulo et al., 2025). Pendekatan *rule-based* memanfaatkan parameter dan ambang batas tertentu yang telah ditentukan sebelumnya, sedangkan *machine learning* memerlukan proses pelatihan menggunakan dataset dalam jumlah besar.

2.2.2 Pengolahan Citra Digital (Image Processing)

Pemrosesan atau Pengolahan Citra (*image processing*) merupakan kumpulan teknik komputasi untuk menganalisis, meningkatkan, mengompresi dan merekonstruksi gambar. *Image processing* juga memiliki pengertian lain yaitu, merupakan suatu bentuk pengolahan atau pemrosesan sinyal dengan input berupa gambar dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu.

Image Processing agar dapat diproses oleh komputer, citra atau gambar direpresentasikan dalam bentuk citra digital, yaitu susunan data numerik yang terdiri dari elemen-elemen kecil yang disebut *pixel*. Pada setiap *pixel* ada bentuk

data image yang berformat RGB dan *grayscale*. Image RGB adalah image yang memiliki warna utama yaitu merah (red), hijau (green), dan biru (blue) disetiap *pixel*. Image *grayscale* adalah image yang menampilkan warna abu-abu hasil dari rata-rata image RGB (Singh et al., 2021).

Menurut Punith Kumar et al. (2020) tinjauan literatur tahapan umum pengolahan citra digital meliputi beberapa tahapan pemrosesan dasar yaitu:

1. *Image Acquisition*: Proses pengambilan atau perolehan citra digital menggunakan sensor yaitu berupa kamera digital.
2. *Preprocessing*: Langkah awal yang bertujuan meningkatkan kualitas citra dan mempersiapkannya untuk proses selanjutnya. Beberapa tindakan umum dalam pra-pemrosesan meliputi pengurangan *noise* dengan menggunakan metode penyaringan seperti *filter Gaussian* atau *filter median*, serta perbaikan geometris dan penyesuaian intensitas.
3. *Segmentation*: Membagi citra menjadi wilayah-wilayah bermakna untuk memisahkan objek PCB dari latar belakang. Segmentasi dilakukan menggunakan metode *thresholding*, kemudian hasil segmentasi diperbaiki menggunakan *operasi morfologi* seperti erosi dan dilasi untuk menghilangkan noise serta memperjelas bentuk objek.
4. Analisis Berbasis Parameter Geometris: Analisis citra dilakukan dengan menghitung parameter geometris objek hasil *segmentasi* seperti luas area, panjang jalur, jumlah lubang dan kontinuitas jalur. Analisis ini dilakukan menggunakan perhitungan dan logika berbasis aturan tanpa proses pembelajaran mesin.
5. Pengambilan Keputusan: Pengambilan keputusan dilakukan dengan membandingkan hasil analisis citra terhadap nilai referensi PCB dengan PCB uji.
6. *Interpretation and Results Representation*: Tahap akhir di mana hasil pengolahan citra diinterpretasikan dan direpresentasikan dalam format yang mudah dipahami.

2.2.3 Thresholding

Metode deteksi cacat PCB berbasis pengolahan citra digital dilakukan dengan menganalisis perbedaan karakteristik visual PCB normal dan PCB cacat melalui proses *thresholding*. *Thresholding* adalah salah satu metode segmentasi citra yang memisahkan antara objek dengan background dalam suatu citra berdasarkan pada perbedaan tingkat kecerahan atau gelap terang.

Metode *thresholding* bekerja dengan menentukan nilai ambang (*threshold*) sehingga piksel dengan intensitas di atas ambang diklasifikasikan sebagai objek. Sedangkan, piksel di bawah nilai ambang diklasifikasikan sebagai latar belakang. Secara matematis, proses *thresholding* biner dapat dinyatakan sebagai:

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases}$$

dimana $f(x, y)$ merupakan nilai intensitas piksel pada citra asli, sedangkan T adalah nilai ambang, dan $g(x, y)$ merupakan citra hasil segmentasi dalam bentuk biner (Syaharuddin et al., 2025).



Gambar 2. 1 Proses segmentasi menggunakan *thresholding* (diadaptasi dari (Syaharuddin et al., 2025)).¹

2.2.4 Morphological Operation

Operasi morfologi suatu teknik pengolahan citra yang digunakan untuk memodifikasi bentuk objek dalam citra biner. Operasi dasar morfologi adalah erosi dan dilasi yang berguna untuk membersihkan noise dan menguatkan fitur objek. Metode ini didasarkan pada teori himpunan dan memanfaatkan elemen struktur sebagai kernel untuk mengevaluasi hubungan spasial antar piksel. Menurut (Gunawan et al., 2016), operasi dasar ini dapat didefinisikan sebagai:

¹ "What is Thresholding in Image Processing? A Guide," diakses 27 Februari 2026, <https://blog.roboflow.com/image-thresholding/>

Dilasi:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

Erosi:

$$A \ominus B = \{z \mid B_z \subseteq A\}$$

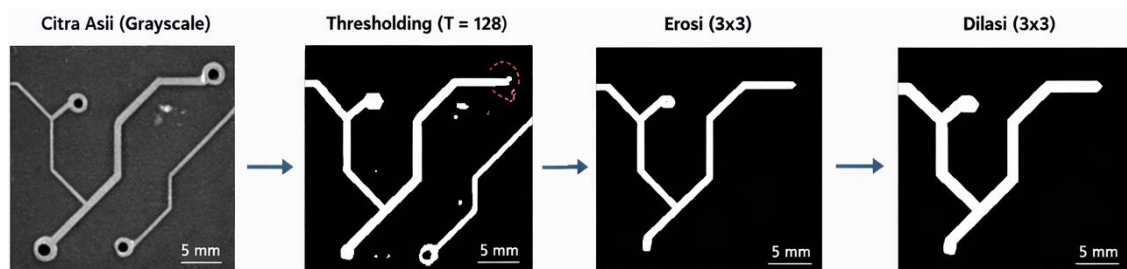
Keterangan:

A : Himpunan piksel Objek

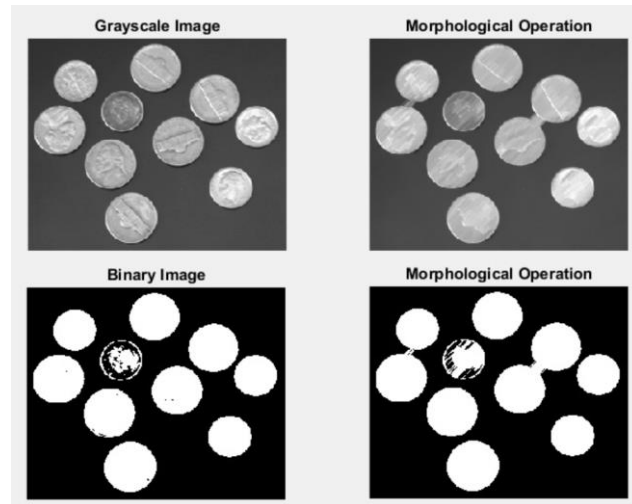
B : Structuring element

Sehingga operasi morfologi memiliki fungsi dan cara kerja sebagai berikut:

1. *Erosion*: Operasi morfologi fundamental yang mengurangi ukuran objek dalam gambar biner. Cara kerjanya adalah dengan menghilangkan piksel dari batas objek. Tujuannya yaitu, untuk menghilangkan noise kecil, memisahkan objek yang terhubung, dan mengikis batas.
2. *Dilation*: Operasi morfologi yang berguna untuk memperbesar ukuran objek dalam gambar. Tujuannya yaitu untuk menggabungkan objek yang berdekatan, mengisi lubang kecil, dan meningkatkan fitur dengan cara kerjanya yaitu, elemen penstruktur bergeser di atas gambar, dan untuk setiap posisi, jika ada piksel di bawah elemen penstruktur yang sama dengan latar depan, piksel dalam gambar keluaran akan diatur menjadi latar depan.



Gambar 2. 2 Ilustrasi Tahapan Operasi Morfologi



Gambar 2. 3 Tahapan Morfologi pada Lubang PCB
(diadaptasi dari (Malge PS & Nadaf RS, 2015)).²

2.2.5 Filter Gaussian

Filter Gaussian adalah jenis filter linear yang memberikan nilai bobot pada setiap elemen berdasarkan bentuk fungsi gaussian. Filter ini digunakan untuk proses penghalusan citra karena memiliki karakteristik central pada kernel, sehingga hasil penyaringan lebih halus dan lebih fokus. Filter Gaussian merupakan metode yang sangat efektif dalam mengurangi interferensi atau noise yang muncul pada citra digital, terutama yang diperoleh melalui penggunaan kamera. Noise ini biasanya muncul secara alami seperti sifat fisik refleksi cahaya dan respon sensor cahaya dari kamera. Secara matematis, fungsi Gaussian dua dimensi dinyatakan sebagai:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

dengan:

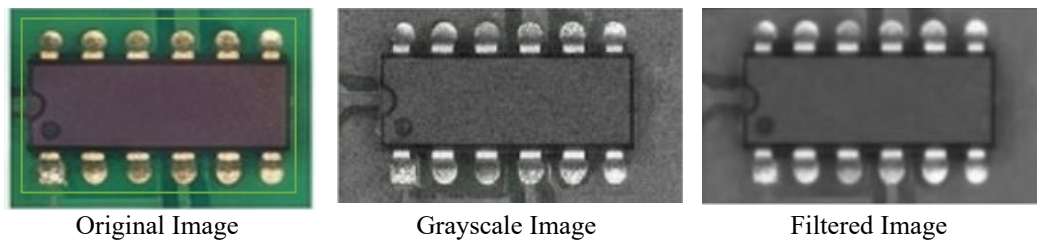
- A. x dan y adalah posisi piksel,
- B. σ adalah standar deviasi yang menentukan tingkat penghalusan citra.

Semakin besar nilai σ , maka efek penghalusan citra akan semakin kuat, tetapi detail citra juga semakin banyak yang hilang. Oleh karena itu, pemilihan

² "Morphological Operation," diakses 27 February 2026, [Morphological Operation – GUI Matlab | Pemrograman Matlab](#)

ukuran kernel dan nilai σ harus disesuaikan dengan kondisi citra PCB yang digunakan (Mazayah Tsaqofah et al., 2025).

Gaussian filter digunakan pada tahap *preprocessing* untuk mengurangi noise dan menghaluskan citra sebelum dilakukan segmentasi atau deteksi komponen. Proses ini dilakukan dengan melakukan konvolusi antara citra dan kernel gaussian sehingga perubahan intensitas yang terlalu tajam dapat dikurangi. Penggunaan gaussian filter mampu membuat posisi pin dan badan komponen DIP menjadi lebih jelas dan stabil terhadap gangguan pencahayaan.



Gambar 2. 4 Implementasi Filter Gaussian untuk Menghilangkan Derau pada Citra PCB (diadaptasi dari (Mira Safitri, 2024)).³

2.2.6 *You Only Look Once (YOLO)*

You Only Look Once (YOLO) merupakan salah satu algoritma object detection berbasis *deep learning* yang dirancang untuk mendeteksi dan mengklasifikasikan objek dalam sebuah citra. Berbeda dengan metode *image processing* berbasis *rule-based* yang bergantung pada parameter seperti *threshold*, ROI, dan *template matching*, YOLO mampu mempelajari karakteristik objek secara langsung dari data pelatihan sehingga proses deteksi dan klasifikasi dilakukan dapat memprediksi lokasi objek dalam bentuk *bounding box*, tingkat kepercayaan (*confidence score*), serta kelas objek yang terdeteksi secara langsung.

Algoritma YOLO banyak digunakan pada berbagai aplikasi *computer vision* karena memiliki keseimbangan yang baik antara kecepatan pemrosesan dan akurasi deteksi. Berbagai versi YOLO telah berhasil diterapkan pada sistem inspeksi PCB untuk mendeteksi jenis cacat pada solderan. Namun demikian, beberapa penelitian sebelumnya masih menemukan kendala dalam mendeteksi objek berukuran kecil maupun objek dengan bentuk yang tidak beraturan. Kemampuan YOLO dalam melakukan deteksi objek secara cepat menjadikannya

³ “PCB dataset for defect detection,” diakses 30 Maret 2026, <https://www.kaggle.com/datasets/mauriziocalabrese/soldef-ai-pcb-dataset-for-defect-detection>

sangat sesuai untuk aplikasi inspeksi otomatis pada lingkungan industri. (Sani et al., 2025).

Pada penelitian ini, algoritma YOLO digunakan pada Position 3 untuk mendeteksi cacat solder secara otomatis. Pemilihan metode ini didasarkan pada kemampuannya dalam mempelajari karakteristik objek dari data pelatihan sehingga mampu mengenali berbagai variasi bentuk cacat dengan lebih adaptif dibandingkan metode *image processing* berbasis *rule-base*. Berbeda dengan pemeriksaan dimensi PCB maupun keberadaan komponen yang masih dapat dilakukan menggunakan *image processing* dan *template matching*, proses deteksi cacat solder memiliki karakteristik visual yang lebih kompleks karena bentuk, ukuran, dan posisinya dapat bervariasi.

2.2.7 Development Environment

Pada proyek akhir ini, lingkungan pengembangan sistem yang digunakan sebagai *platform* untuk menulis program yaitu *Visual Studio Code*. *Visual Studio Code* adalah aplikasi *code editor* buatan Microsoft yang dapat dijalankan di semua perangkat desktop. *Visual Studio Code* dirancang untuk pengembangan editor yang ringan, cepat, tetapi tetap memiliki kemampuan yang setara dengan *Integrated Development Environment* (IDE) sehingga banyak mendukung bahasa pemrograman di antaranya pemrograman *JavaScript*, *TypeScript* hingga *Node.js* tidak hanya itu, *Visual Studio Code* juga mendukung bahasa pemrograman lain seperti *Python*, PHP, .Net dan Java.

Visual Studio Code menyediakan fitur-fitur penting seperti *syntax highlighting*, *code completion*, *debugging*, serta pengelolaan folder proyek yang terstruktur, sehingga memudahkan pengembang dalam menulis, menguji, dan memelihara kode program. Dengan dukungan ekstensi *Python*, *Visual Studio Code* dapat digunakan untuk menjalankan dan menguji program pengolahan citra digital berbasis pustaka *OpenCV* secara langsung. *Python* merupakan bahasa pemrograman berorientasi objek, interpretative dan interaktif. Bahasa

yang menyediakan struktur data tingkat tinggi seperti, *such as list, tuples, set, array asosiatif*, modul, memory management dan *otomatic*.



Gambar 2. 5 *Visual Studio Code*⁴ & *Python*⁵.

Dalam konteks proyek akhir ini, *Visual Studio Code* digunakan sebagai lingkungan pengembangan utama (*development environment*) untuk menulis, menjalankan, dan menguji kode program yang dibuat menggunakan bahasa pemrograman *Python*. *Visual Studi Code* berfungsi sebagai media pengembangan perangkat lunak yang memudahkan proses penulisan kode, pengelolaan file program, serta proses debugging sistem inspeksi visual PCB. Sedangkan *Python* berperan sebagai bahasa pemrograman utama untuk mengimplementasikan sistem inspeksi visual PCB otomatis berbasis pengolahan citra digital. *Python* berperan dalam proses akuisisi citra, pra-pemrosesan, segmentasi, analisis berbasis parameter geometris, pengambilan keputusan PASS/FAIL, serta penyajian hasil inspeksi dalam bentuk antarmuka pengguna grafis (*GUI*) yang disimpan dalam bentuk data pengujian serta citra hasil deteksi ke dalam folder database sebagai dokumentasi pengujian.

2.2.8 *Library OpenCV*

OpenCV (OpenSource Computer Vision) merupakan sebuah library yang dirancang untuk menyelesaikan berbagai tugas yang berkaitan dengan design computer visison. *OpenCV* mendukung penggunaan berbagai bahasa object termasuk *Python, C++ dan Java*, dan tersedia di berbagai *platform* termasuk *iOS, Android, OS X, Linux, dan Windows* (Singh et al., 2021).

⁴ “Visual Studio Code Adalah: Pengertian Hingga Manfaatnya”, diakses 2 February 2026 <https://itbox.id/blog/visual-studio-code-adalah-pengertian-hingga-manfaatnya>

⁵ “Python is a programming language that lets you work quicklyand integrate systems more effectively”, diakses 2 February 2026 <https://www.python.org/>



Gambar 2. 6 Library *OpenCV*.⁶

OpenCV dirancang untuk bekerja dengan gambar dan video, menyediakan alat-alat yang diperlukan untuk pemrosesan, analisis dan pemahaman visual untuk mengambil gambar dari kamera, memuatnya dari disk, atau menerima aliran video secara real-time dan algoritma yang diterapkan pada gambar akan di analisis. Dalam upaya memaksimalkan kinerja tugas visi yang intensif komputasi, *OpenCV* menyertakan dukungan untuk hal-hal berikut yaitu, multithreading pada komputer multi-core menggunakan *Threading Building Blocks* (TBB), Subset dari *Integrated Performance Primitives* (IPP) pada prosesor Intel untuk meningkatkan kinerja dan antarmuka untuk pemrosesan.

Aplikasi *OpenCV* mencakup berbagai bidang seperti segmentasi dan pengenalan, perangkat fitur 2D dan 3D, identifikasi objek, pengenalan wajah, pelacakan gerakan, pengenalan isyarat, penyambungan gambar, pencitraan rentang dinamis tinggi (HDR), realitas tertambah, dan sebagainya García et al. (2015). Dalam proyek akhir ini, *OpenCV* digunakan sebagai library utama untuk mengimplementasikan seluruh tahapan pengolahan citra digital pada sistem inspeksi visual PCB otomatis. *OpenCV* dimanfaatkan untuk melakukan akuisisi citra PCB menggunakan kamera, melakukan preprocessing citra seperti konversi ke grayscale dan pengurangan noise, serta melakukan segmentasi citra menggunakan metode thresholding.

Pada proyek akhir ini, *OpenCV* menjadi library pengolah citra berbasis *rule-based* (*thresholding & morfologi*) yang digunakan untuk mendeteksi parameter PCB secara *real-time*. Selain itu, *OpenCV* digunakan untuk melakukan operasi morfologi seperti erosi dan dilasi guna memperbaiki hasil segmentasi, serta mendeteksi kontur objek untuk menganalisis parameter geometris PCB. Parameter

⁶ "OpenCV-Python Tutorials", diakses 27 February 2026 https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

yang dianalisis meliputi luas area jalur, panjang jalur konduktor, serta jumlah lubang (via) pada PCB. Hasil analisis tersebut kemudian dibandingkan dengan nilai referensi PCB normal untuk menentukan kondisi PCB dalam kategori PASS atau FAIL.

2.2.9 Tkinter / PyQt (GUI)

Graphical User Interface (GUI) merupakan antarmuka berbasis grafis yang memungkinkan pengguna berinteraksi dengan sistem melalui elemen visual seperti tombol (button), teks, gambar, dan jendela tampilan. GUI bertujuan untuk mempermudah pengguna dalam mengoperasikan sistem tanpa harus berinteraksi langsung dengan kode program.



Gambar 2. 7 Tkinter.⁷

Sebuah library untuk membangun GUI ini yaitu Tkinter library standar *Python* yang digunakan untuk membangun *Graphical User Interface* (GUI). Tkinter menyediakan berbagai komponen antarmuka seperti window, label, button, canvas, dan entry yang dapat digunakan untuk membangun aplikasi berbasis desktop secara sederhana dan efisien. Karena merupakan library bawaan *Python*, Tkinter tidak memerlukan instalasi tambahan dan mudah diintegrasikan dengan program *Python* lainnya.

Dalam sistem inspeksi visual otomatis, GUI akan berperan sebagai media interaksi antara pengguna dengan sistem pengolahan citra, sehingga pengguna dapat mengontrol proses inspeksi, melihat hasil pengolahan citra, serta memantau hasil analisis secara visual dengan itu Tkinter akan menjadi media untuk menampilkan citra PCB hasil akuisisi kamera, menampilkan hasil deteksi cacat

⁷ "Improving Tkinter designs with ttkbootstrap", diakses 2 February 2026 <https://medium.com/@jdgb.projects/improving-tkinter-designs-with-ttkbootstrap-7c88de67d570>

PCB, menampilkan status inspeksi berupa PASS atau FAIL, serta menyediakan tombol kontrol untuk menjalankan proses inspeksi dan menyimpan hasil pengujian ke dalam database sistem. Tkinter tidak melakukan proses pengolahan citra, melainkan hanya digunakan sebagai antarmuka visual.

2.2.10 Camera Webcam Logitech C922

Kamera webcam *Logitech* merupakan sensor utama dalam sistem inspeksi yang berfungsi untuk mengambil citra PCB dengan kemampuannya untuk melakukan kontrol parameter internal secara programatik melalui *OpenCV*, seperti penguncian fokus (*focus locking*) dan pengaturan pencahayaan manual (*manual exposure*). Hal ini krusial untuk menjaga konsistensi data visual saat kamera berpindah posisi.



Gambar 2. 8 WebCam *Logitech C299*.⁸

⁸ “C922 PRO HD STREAM WEBCAM”, diakses 20 February 2026 <https://www.logitech.com/en-us/shop/p/c922-pro-stream-webcam>