

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

Object segmentation merupakan proses memisahkan atau mengekstraksi bagian tertentu dalam citra sehingga area objek dapat dibedakan dari latar belakang. Proses ini bertujuan menghasilkan representasi citra yang lebih sederhana dan terstruktur sehingga objek dapat dianalisis atau diproses lebih lanjut, misalnya untuk pengukuran bentuk, pengenalan objek, pemindaian dokumen, segmentasi objek, dan aplikasi robotika. Dalam pengolahan citra klasik, segmentasi sering dilakukan tanpa anotasi data pelatihan, sehingga pemisahan objek dilakukan berdasarkan karakteristik visual seperti intensitas, warna, tekstur, atau batas tepi objek. Pendekatan seperti ini sering disebut sebagai *unsupervised segmentation*, yaitu proses ekstraksi objek secara langsung dari citra untuk memperoleh *Region of Interest (ROI)* tanpa memerlukan proses pelabelan piksel terlebih dahulu. Segmentasi jenis ini masih banyak digunakan karena relatif sederhana serta dapat diterapkan pada berbagai jenis citra tanpa proses pelatihan model yang kompleks (Jasim & Mohammed, 2021; Ramadan dkk., 2020).

Secara historis, metode segmentasi tradisional mencakup teknik *thresholding*, *region growing*, deteksi tepi yang diikuti dengan pengisian area objek, serta metode berbasis kontur dan *graph*. Metode-metode tersebut memiliki kelebihan berupa struktur algoritma yang sederhana, kebutuhan komputasi yang relatif kecil, serta tidak bergantung pada dataset pelatihan. Oleh karena itu, metode

klasik masih banyak digunakan pada sistem dengan sumber daya terbatas maupun pada tahap awal analisis citra. Namun demikian, metode klasik memiliki keterbatasan berupa sensitivitas terhadap *noise*, perubahan pencahayaan, serta variasi tekstur latar belakang yang dapat menyebabkan hasil segmentasi menjadi tidak stabil. Untuk mengatasi permasalahan tersebut, banyak penelitian mengombinasikan metode segmentasi dengan proses pra-pemrosesan seperti *filtering* maupun operasi morfologi agar hasil segmentasi menjadi lebih konsisten (Jasim & Mohammed, 2021; Plaksyvyi dkk., 2023).

Perkembangan terbaru dalam bidang segmentasi citra menunjukkan pergeseran menuju pendekatan berbasis *deep learning* yang memanfaatkan model *convolutional neural networks* maupun *transformer*. Pendekatan ini umumnya melakukan segmentasi melalui proses pelabelan piksel sehingga setiap piksel pada citra diberi label kelas tertentu, misalnya melalui tugas *semantic segmentation*, *instance segmentation*, maupun *panoptic segmentation*. Model berbasis *deep learning* terbukti mampu menghasilkan akurasi yang tinggi karena dapat mempelajari fitur visual yang kompleks dari data pelatihan berskala besar. Namun demikian, metode ini memerlukan dataset anotasi yang besar, proses pelatihan yang lama, serta perangkat komputasi yang relatif tinggi sehingga tidak selalu sesuai untuk semua kebutuhan pengolahan citra (Minace dkk., 2020; Zhu, 2023).

Kemajuan terbaru dalam bidang segmentasi objek juga ditandai dengan hadirnya model *foundation* yang dirancang memiliki kemampuan generalisasi luas. Salah satu contoh utamanya adalah *Segment Anything Model* yang mampu menghasilkan mask objek secara *zero-shot* hanya melalui *prompt* dari pengguna.

Model ini dilatih menggunakan dataset berskala sangat besar sehingga dapat digunakan pada berbagai jenis citra tanpa pelatihan ulang secara penuh. Walaupun demikian, ukuran model yang besar, kebutuhan komputasi yang tinggi, serta kompleksitas implementasi masih menjadi kendala dalam penerapan model tersebut pada sistem dengan sumber daya terbatas (Kirillov dkk., 2023; C. Zhang dkk., 2023).

Meskipun metode *deep learning* menawarkan tingkat akurasi yang tinggi, beberapa permasalahan masih menjadi perhatian dalam penelitian segmentasi citra. Kebutuhan anotasi piksel yang besar menyebabkan proses pembangunan dataset menjadi mahal dan memerlukan waktu yang lama. Selain itu, kemampuan generalisasi model terhadap jenis citra baru sering memerlukan proses penyesuaian ulang, terutama ketika karakteristik citra berbeda dari data pelatihan. Segmentasi yang presisi pada batas objek juga masih menjadi tantangan, terutama pada objek dengan kontras rendah atau tepi yang tidak jelas. Oleh karena itu, pendekatan segmentasi berbasis teknik klasik masih tetap relevan digunakan, terutama pada kasus yang memerlukan proses segmentasi cepat tanpa ketergantungan pada dataset pelatihan (Krikid dkk., 2024; Plaksyvyi dkk., 2023).

Secara umum dapat disimpulkan bahwa metode segmentasi klasik dan metode berbasis *deep learning* memiliki kelebihan masing-masing. Metode klasik cenderung lebih sederhana dan efisien secara komputasi serta dapat digunakan untuk mengekstraksi objek secara langsung sebagai *Region of Interest*, sedangkan metode *deep learning* mampu memberikan hasil segmentasi yang lebih akurat pada permasalahan yang kompleks. Oleh karena itu, pemilihan metode segmentasi perlu

disesuaikan dengan kebutuhan aplikasi, terutama pada kondisi yang memerlukan proses segmentasi tanpa pelatihan model atau anotasi data.

2.1.1 Image Acquisition

Image Acquisition adalah tahap paling awal dalam proses segmentasi objek yang berfungsi untuk memperoleh citra digital yang akan diproses lebih lanjut. Tahap ini melibatkan pengambilan gambar melalui perangkat optik (misalnya kamera) sehingga data visual dalam bentuk analog dapat diubah menjadi citra digital yang dapat diolah oleh sistem komputer *vision* untuk tugas segmentasi atau pengenalan objek. Proses akuisisi citra sangat penting karena kualitas citra awal secara langsung memengaruhi hasil pada tahap pemrosesan berikutnya seperti deteksi tepi dan segmentasi objek. Akuisisi citra yang baik akan menghasilkan data yang akurat terhadap objek penelitian dan meminimalkan gangguan sejak awal pemrosesan data citra digital (Chen dkk., 2021; Mas'ud dkk., 2023).

2.1.2 Image Preprocessing

Image preprocessing merupakan tahap awal yang penting dalam pengolahan citra digital. Pada tahap ini, citra mentah (*raw image*) yang diperoleh dari berbagai sumber (misalnya kamera digital, *scanner*, atau sensor lainnya) diproses untuk meningkatkan kualitas dan menyesuaikan karakteristik citra sebelum dilakukan analisis lebih lanjut. Langkah ini bertujuan untuk mengurangi gangguan (*noise*), memperbaiki kontras, saturasi, menormalkan intensitas, serta mempersiapkan citra agar algoritma pengolahan atau model *deep learning* dapat

bekerja secara optimal (Ngugi dkk., 2021). Berikut ini adalah penjelasan dari beberapa teknik *image preprocessing*:

1. *Resize* atau *Downscale*

Proses *resize* atau *downscale* dilakukan untuk mengurangi ukuran citra dengan tujuan menurunkan beban komputasi pada operasi pengolahan yang dilakukan pada setiap piksel, seperti proses *denoising*, perhitungan gradien Sobel, dan operasi morfologi. Semakin besar ukuran citra, semakin banyak piksel yang harus diproses, sehingga waktu eksekusi menjadi lebih lama. Dengan mengurangi resolusi citra, waktu pemrosesan dapat ditekan sehingga sistem lebih efisien dan memungkinkan untuk diterapkan pada perangkat dengan keterbatasan sumber daya, seperti perangkat *mobile* (Prayogi dkk., 2025).

Selain memengaruhi kecepatan pemrosesan, *resize* juga berdampak pada parameter lain yang bergantung pada ukuran citra, misalnya ambang luas area pada proses *connected components*. Oleh karena itu, nilai ambang area dalam penelitian ini dihitung berdasarkan ukuran citra setelah proses *resize*, bukan ukuran citra asli. Secara umum, proses *resize* dilakukan dengan menentukan faktor skala berdasarkan sisi terpanjang citra. Misalkan W dan H menyatakan lebar dan tinggi citra asli atau $maxSide = \max(W, H)$. Jika nilai $maxSide$ melebihi batas maksimum yang ditentukan, yaitu $maxOutSide$, maka faktor skala(s) dihitung menggunakan Persamaan (2.1):

$$s = \frac{maxOutSide}{maxSide} \quad (2.1)$$

Ukuran citra yang baru kemudian diperoleh dengan mengalikan lebar dan tinggi citra asli dengan faktor skala tersebut, sebagaimana ditunjukkan pada Persamaan (2.2):

$$W_{new} = \lfloor W \times s \rfloor, H_{new} = \lfloor H \times s \rfloor. \quad (2.2)$$

Kompromi utama dari strategi *down scaling* adalah potensi hilangnya detail halus ketika citra diperkecil terlalu berlebihan. Oleh karena itu, pemilihan nilai *maxOutSide* perlu dilakukan secara hati-hati agar beban komputasi dapat dikurangi tanpa menghilangkan informasi tepi penting yang dibutuhkan pada tahap deteksi berikutnya.

2. *Denoising*

Tahap *filtering* dan *denoising* bertujuan untuk mengurangi *noise* serta artefak yang tidak diinginkan pada citra sebelum dilakukan analisis lanjutan. *Noise* yang tidak ditangani sejak awal dapat mengganggu perhitungan gradien dan menyebabkan hasil deteksi tepi menjadi tidak stabil (H. Li & Xu, 2025). Penelitian ini menggunakan metode *Mean Shift Filtering* sebagai teknik utama dalam tahap *denoising*. *Mean Shift* merupakan metode berbasis analisis *feature space* yang bekerja dengan mengelompokkan piksel berdasarkan kesamaan warna dan kedekatan spasial. Metode ini tidak hanya mengurangi *noise*, tetapi juga melakukan segmentasi awal secara implisit dengan mempertahankan batas antar objek yang memiliki perbedaan warna signifikan.

Secara matematis, proses *Mean Shift* dapat dinyatakan sebagai pergeseran vektor menuju densitas maksimum dalam ruang fitur, dapat dinyatakan melalui persamaan (2.3):

$$m(x) = \frac{\sum_{i=1}^n K(x_i - x)x_i}{\sum_{i=1}^n K(x_i - x)} - x \quad (2.3)$$

Dalam implementasinya, *Mean shift* diambil dari fungsi `pyrMeanShiftFiltering` pada *library* OpenCV untuk digunakan dalam menyederhanakan distribusi warna citra dengan mempertahankan struktur tepi objek. Pendekatan ini terbukti efektif dalam meningkatkan kualitas segmentasi karena mampu mereduksi *noise* sekaligus menjaga diskontinuitas intensitas pada batas objek. Pendekatan *Mean Shift filtering* dipilih karena mampu untuk mereduksi *noise*, sekaligus mempertahankan batas antar objek berdasarkan distribusi warna. Hasil *filtering* menghasilkan distribusi warna yang lebih homogen sehingga proses ekstraksi *L-channel* pada ruang LAB menjadi lebih stabil. Selanjutnya *L-channel* digunakan sebagai *input* pada proses deteksi tepi menggunakan metode *Canny Edge Detection* dan *Adaptive Thresholding*.

Tahap *preprocessing* bertujuan mengurangi *noise* seawal mungkin agar perhitungan gradien warna dan proses *thresholding* menjadi lebih stabil. Pendekatan modern seperti *non-local means* dan penggunaan model *deep learning* juga telah diaplikasikan untuk mencapai *proses denoising* yang lebih adaptif dan akurat (Salvi dkk., 2021a).

3. *L-Channel*

Pada tahap ini, citra dikonversi ke dalam ruang warna CIE-LAB (LAB) dan hanya komponen *L-channel (luminance)* yang digunakan sebagai input utama untuk proses deteksi tepi menggunakan metode Canny serta proses *Adaptive Thresholding*. Berbeda dengan konversi *grayscale* konvensional, *L-channel* pada

ruang warna LAB merepresentasikan tingkat kecerahan secara lebih terpisah dari informasi warna (*chrominance*), yaitu komponen a dan b. Transformasi dari ruang warna RGB ke LAB dilakukan untuk memisahkan informasi *luminance* dan warna, sehingga analisis intensitas menjadi lebih stabil terhadap variasi warna dan pencahayaan. Secara umum, ruang warna LAB dirancang agar lebih mendekati persepsi visual manusia, di mana perubahan nilai L merepresentasikan perubahan kecerahan secara linear terhadap persepsi. Penggunaan L-channel memberikan beberapa keuntungan dalam konteks segmentasi berbasis tepi. Pertama, proses deteksi tepi menjadi lebih robust karena hanya bergantung pada perubahan intensitas, bukan variasi warna. Kedua, metode Adaptive Thresholding dapat bekerja lebih stabil dalam kondisi pencahayaan yang tidak merata, karena perhitungan ambang dilakukan berdasarkan distribusi lokal intensitas. Pendekatan ini banyak digunakan dalam berbagai penelitian modern karena kemampuannya dalam meningkatkan kualitas segmentasi, terutama pada citra dengan variasi warna kompleks dan pencahayaan yang tidak seragam (H. Li dkk., 2023).

Secara keseluruhan, tahap *image preprocessing* merupakan komponen yang sangat penting dalam rangkaian pengolahan citra digital. Teknik-teknik yang diterapkan pada tahap ini tidak hanya berfungsi untuk meningkatkan kualitas visual citra, tetapi juga membangun fondasi yang kuat bagi penerapan algoritma pengolahan citra pada tahap selanjutnya. Proses *preprocessing* yang tepat dapat membantu mengurangi *noise*, menstabilkan pencahayaan, serta memperjelas struktur objek, sehingga algoritma yang lebih kompleks dapat bekerja secara lebih optimal dan konsisten (Chen dkk., 2021). Oleh karena itu, pemahaman yang

mendalam mengenai berbagai metode *preprocessing* serta penyesuaiannya terhadap karakteristik domain aplikasi menjadi faktor kunci dalam pengembangan sistem pengolahan citra yang akurat dan efisien.

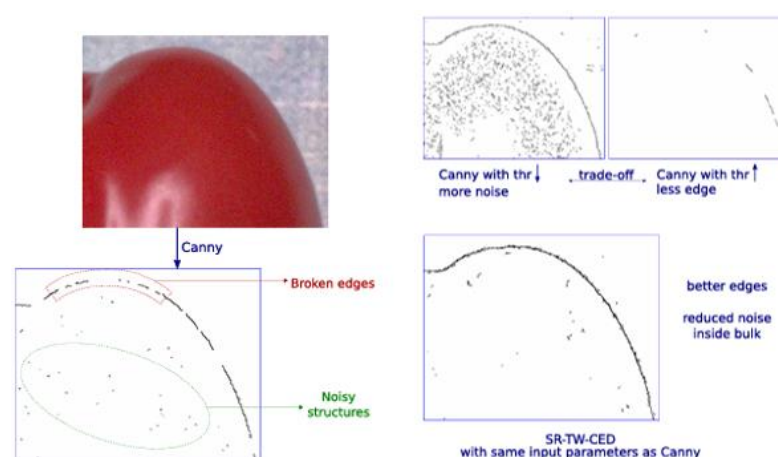
2.1.3 Canny Edge Detection

Algoritma *Canny Edge Detection* merupakan salah satu metode klasik yang paling banyak digunakan dalam pendeteksian tepi pada pengolahan citra digital. Metode ini diperkenalkan oleh Canny pada tahun 1986 dan dirancang untuk memenuhi tiga kriteria utama, yaitu rasio sinyal terhadap *noise* yang tinggi, ketepatan lokasi tepi yang baik, serta respon tepi tunggal agar tidak terjadi deteksi ganda pada satu struktur tepi yang sama (Dhillon & Chouhan, 2022). Dengan karakteristik tersebut, algoritma Canny dikenal memiliki keseimbangan yang baik antara sensitivitas terhadap tepi dan ketahanan terhadap *noise*.

Tahap awal pada algoritma Canny adalah pengurangan *noise* melalui proses *filtering*. Umumnya, proses ini dilakukan menggunakan *Gaussian filter* untuk menghaluskan citra sebelum perhitungan gradien dilakukan (Cao dkk., 2020). Meskipun efektif dalam mereduksi *noise*, penggunaan *Gaussian filter* memerlukan penentuan parameter secara manual dan dapat menyebabkan hilangnya detail tepi akibat efek pengaburan (*blur*) yang berlebihan (K. Liu dkk., 2017). Oleh karena itu, beberapa penelitian mengimplementasikan penggunaan *adaptive median filtering* sebagai alternatif karena dinilai lebih mampu menjaga ketajaman tepi sambil tetap menekan *noise* (Dhillon & Chouhan, 2022).

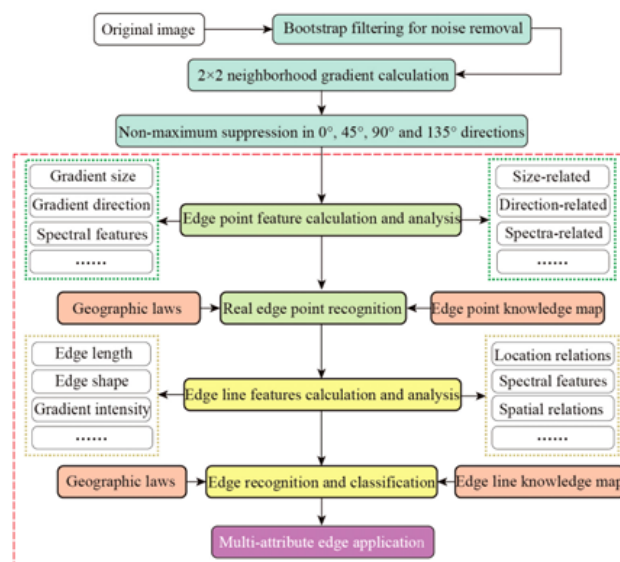
Secara umum, algoritma Canny terdiri dari beberapa tahap utama yang saling berurutan. Setelah proses perataan citra untuk mereduksi *noise*, dilakukan perhitungan gradien intensitas untuk mendeteksi perubahan nilai piksel yang signifikan, yang mengindikasikan keberadaan tepi (Guo dkk., 2023).

Tahap berikutnya adalah *double thresholding*, yang menggunakan dua nilai ambang, yaitu ambang tinggi dan ambang rendah. Piksel dengan nilai magnitudo gradien di atas ambang tinggi diklasifikasikan sebagai tepi kuat, sedangkan piksel dengan nilai gradien di antara ambang tinggi dan ambang rendah dikategorikan sebagai tepi lemah. Setelah itu, dilakukan proses *edge tracking by hysteresis*, di mana piksel tepi lemah akan dipertahankan sebagai bagian dari tepi akhir jika memiliki keterhubungan dengan piksel tepi kuat melalui piksel-piksel bertetangga. Piksel tepi lemah yang tidak terhubung dengan tepi kuat akan dibuang. Mekanisme ini berperan penting dalam menjaga kontinuitas tepi sekaligus menekan kemunculan tepi palsu akibat *noise* (Agrawal & Desai, 2024).



Gambar 2.1 Permasalahan *Canny Edge Detection* (Dhillon & Chouhan, 2022)

Beberapa penelitian selanjutnya mengembangkan algoritma Canny dengan memperkenalkan penentuan ambang secara adaptif, misalnya dengan memanfaatkan metode Otsu atau pendekatan berbasis algoritma genetika, guna meningkatkan akurasi deteksi tepi pada berbagai kondisi citra (R. Liu & Mao, 2018). Selain itu, pendekatan lain memanfaatkan *geo-information mapping* untuk memperbaiki tepi yang terputus (*broken Edge*) dan meningkatkan ketepatan deteksi, khususnya pada citra penginderaan jauh (Lijun dkk., 2023).



Gambar 2.2 *Flow chart improved canny method* (Lijun dkk., 2023)

Sebagai ilustrasi, Gambar 2.1 menunjukkan permasalahan umum yang sering muncul pada hasil deteksi tepi menggunakan algoritma Canny, seperti tepi yang terputus atau terdeteksi ganda. Sementara itu, Gambar 2.2 menampilkan alur metode Canny yang telah ditingkatkan untuk mengatasi permasalahan tersebut. Meskipun algoritma Canny telah terbukti efektif dan banyak diaplikasikan di berbagai bidang, penentuan parameter *filtering* dan nilai *threshold* yang tepat masih menjadi tantangan utama. Oleh karena itu, berbagai upaya pengembangan terus

dilakukan untuk menghasilkan peta tepi yang lebih akurat dan lebih tahan terhadap gangguan *noise* (Dhillon & Chouhan, 2022).

2.1.4 Adaptive Thresholding

Adaptive Thresholding adalah teknik binarisasi citra yang menentukan nilai ambang secara lokal sehingga setiap area kecil pada citra dapat memiliki ambang yang berbeda sesuai karakteristik intensitasnya. Teknik ini dikembangkan untuk mengatasi keterbatasan *thresholding* global yang menggunakan satu nilai ambang untuk seluruh citra dan sering gagal ketika pencahayaan tidak merata atau muncul gradien iluminasi. Pada praktik modern, *Adaptive Thresholding* tetap menjadi langkah penting pada alur pemrosesan dan segmentasi objek karena kemampuannya memisahkan *foreground* dari *background* pada kondisi pencahayaan yang bervariasi (Guo dkk., 2023). Secara matematis, untuk setiap posisi (x, y) ambang lokal dihitung sebagai rata-rata intensitas pada jendela $W(x, y)$ dikurangi suatu konstanta C :

$$T(x, y) = \frac{1}{|W|} \sum_{(i, j) \in W(x, y)} I(i, j) - C \quad (2.4)$$

Dalam perkembangan terakhir, *Adaptive Thresholding* tidak lagi dilihat hanya sebagai algoritma tunggal, tetapi sering dirancang sebagai modul multi-tahap. Beberapa penelitian menyarankan strategi dua tahap atau multi-skala di mana tahap pertama melakukan peningkatan kontras dan reduksi *noise*, sedangkan tahap kedua melakukan perhitungan ambang lokal pada jendela dinamis yang disesuaikan terhadap skala objek. Pendekatan dua tahap terbukti lebih tahan terhadap kondisi *noise* dan kontras rendah, misalnya dalam citra bawah air atau citra dokumen yang

terdegradasi, karena tahap pra-pemrosesan meminimalkan artefak yang dapat memengaruhi perhitungan ambang lokal (Song dkk., 2022).

Perbaikan lain yang banyak diteliti adalah penggabungan *Adaptive Thresholding* dengan teknik denoising yang menjaga tepi, misalnya *bilateral filtering* atau metode *denoising* berbasis *deep learning* yang bersifat *edge-preserving*. Integrasi ini bertujuan agar statistik lokal yang digunakan untuk menentukan ambang tidak tercemar oleh *noise*, sehingga hasil binarisasi mempertahankan detail objek dan mengurangi *false positive* pada tekstur latar. Penelitian terbaru juga mengkaji penggunaan model pembelajaran untuk melakukan *prefiltering* sehingga *thresholding* lokal bekerja pada citra yang lebih bersih tanpa menghilangkan tepian penting (Maus dkk., 2024). Selain itu, beberapa studi mengembangkan teknik *windowing* yang adaptif, yakni ukuran jendela untuk perhitungan ambang tidak statis melainkan ditentukan berdasarkan karakteristik lokal seperti kontras, densitas tepian, atau ukuran komponen yang terdeteksi. Pendekatan ini mengurangi *trade-off* antara deteksi detail halus dan stabilitas terhadap *noise*. Metode *window* dinamis dan strategi penggabungan hasil dari berbagai skala telah dilaporkan meningkatkan kualitas binarisasi pada dokumen yang memiliki area sangat terang atau sangat gelap, serta pada citra yang mengalami gradasi *luminance* yang kuat.

Hasil evaluasi terhadap berbagai metode binarisasi adaptif menunjukkan bahwa tidak ada satu metode pun yang selalu unggul, sehingga pemilihannya harus disesuaikan dengan karakteristik citra dan kebutuhan aplikasi. Studi perbandingan terkini juga menegaskan pentingnya metrik evaluasi yang

mempertimbangkan noise, ketajaman tepi atau teks, serta efisiensi proses. Hal ini membantu menentukan kombinasi pra-pemrosesan, *thresholding* lokal, dan morfologi pasca-proses yang paling efektif untuk kasus tertentu (Nugroho dkk., 2019). *Adaptive Thresholding* optimal bila dikombinasikan dengan pra-pemrosesan, strategi lokal yang tepat, dan pasca-pemrosesan. Pada tahap pra-pemrosesan lakukan koreksi *iluminance* dan *denoising* yang mempertahankan tepian sehingga statistik lokal tidak terdistorsi. Pada tahap *thresholding* gunakan ukuran jendela atau pendekatan multi-skala yang sesuai dengan skala objek untuk menghindari *oversegmentation* atau hilangnya detail kecil. Akhiri dengan pasca-pemrosesan morfologi untuk menghapus artefak kecil dan menyambung komponen yang terputus sehingga *mask* akhir lebih stabil.

2.1.5 Distance transform

Distance transform merupakan salah satu teknik penting dalam pengolahan citra biner yang digunakan untuk menghitung jarak setiap piksel terhadap piksel terdekat pada objek atau latar belakang. Secara umum, distance transform mengubah citra biner menjadi sebuah peta jarak (*distance map*), di mana nilai setiap piksel merepresentasikan jarak minimum terhadap batas objek (*boundary*). Teknik ini banyak digunakan dalam berbagai aplikasi seperti segmentasi citra, analisis bentuk, serta deteksi struktur objek (Silva dkk., 2025). Dalam implementasinya, *distance transform* biasanya diterapkan setelah proses *thresholding*, di mana citra telah dipisahkan menjadi *foreground* (objek) dan *background*. Nilai jarak yang dihasilkan dapat dihitung menggunakan beberapa metrik, seperti *Euclidean distance*, *Manhattan distance*, maupun *Chessboard*

distance. Di antara ketiganya, *Euclidean distance transform* (EDT) merupakan metode yang paling umum digunakan karena memberikan hasil jarak yang lebih akurat secara geometris terhadap kontur objek (Öfverstedt dkk., 2020).

Penggunaan *distance transform* dalam segmentasi citra memiliki peran penting dalam memperkuat representasi area objek. Dengan memanfaatkan informasi jarak, metode ini mampu menyoroti bagian tengah objek (*region interior*) dan mengurangi pengaruh *noise* di sekitar batas objek. Hal ini menjadikan *distance transform* efektif dalam membantu proses pemisahan objek dari latar belakang, terutama pada kondisi di mana batas objek tidak jelas atau terputus. Selain itu, *distance transform* juga sering digunakan sebagai langkah tambahan untuk meningkatkan kualitas segmentasi, baik dalam metode klasik maupun modern. Dalam beberapa penelitian, *distance transform* digunakan sebagai peta bobot (*weight map*) yang membantu meningkatkan fokus pada area batas objek sehingga menghasilkan segmentasi yang lebih akurat dan stabil (X. Huang dkk., 2022).

Dalam konteks penelitian ini, *distance transform* digunakan sebagai bagian dari pipeline segmentasi untuk memperkuat struktur area objek setelah proses deteksi tepi dan thresholding. Dengan demikian, metode ini membantu menghasilkan segmentasi yang lebih utuh sebelum dilakukan proses lanjutan seperti operasi morfologi dan pembentukan *convex hull*.

2.1.6 PostProcessing

Postprocessing merupakan tahapan lanjutan dalam proses segmentasi citra yang bertujuan untuk menyempurnakan hasil deteksi awal, seperti *edge map*

yang dihasilkan oleh algoritma deteksi tepi (misalnya *Canny*) dan citra biner hasil *Adaptive Thresholding*, agar diperoleh masker objek yang lebih stabil dan representatif. Secara umum, *postprocessing* digunakan untuk mengurangi komponen kecil yang tidak relevan (*speckle noise* dan pulau-pulau kecil), menutup celah yang menyebabkan terputusnya kontinuitas tepi, memilih komponen utama berdasarkan ukuran atau karakteristik geometris, serta menghaluskan batas objek sehingga menghasilkan *alpha mask* yang lebih alami saat diaplikasikan pada citra asli (Baldissera dkk., 2021; Salvi dkk., 2021b).

Secara konseptual, proses *postprocessing* bekerja pada domain biner dan peta tepi, sehingga berbeda dengan tahap *preprocessing* yang beroperasi langsung pada citra berwarna (*RGB/BGR*). Pada tahap *preprocessing*, teknik *denoising* diterapkan untuk menurunkan derau pada citra input guna menjaga kualitas sinyal sebelum perhitungan gradien dan deteksi tepi dilakukan. Sebaliknya, *postprocessing* tidak berfokus pada perbaikan kualitas citra mentah, melainkan pada perbaikan struktur biner dan struktur tepi hasil deteksi agar segmentasi yang dihasilkan lebih bersih, lebih konsisten secara spasial, dan lebih mudah diolah oleh tahapan selanjutnya (Xu dkk., 2024).

Selain berfungsi sebagai tahap penyempurnaan hasil segmentasi, *postprocessing* juga sering mencakup serangkaian langkah tambahan untuk membentuk masker terhadap *region of interest (ROI)* yang didukung oleh keberadaan tepi, sehingga area-area yang tidak memiliki keterkaitan spasial dengan objek utama dapat dieliminasi secara efektif (Baldissera dkk., 2021).

1. *Resize* atau *Upscale*

Karena proses segmentasi dilakukan pada citra hasil *downscale*, koordinat dan ukuran area objek yang diperoleh masih mengacu pada dimensi citra hasil *resize*. Oleh karena itu, diperlukan proses *upscaling* untuk mengembalikan hasil segmentasi ke ukuran citra asli sehingga posisi dan ukuran objek tetap sesuai dengan kondisi sebenarnya pada citra *original*. Proses ini memanfaatkan faktor skala yang sama dengan faktor yang digunakan pada tahap *downscale* sebagaimana ditunjukkan pada persamaan (2.1). Dengan mengetahui faktor skala tersebut, koordinat maupun dimensi *Region of Interest* (ROI) yang diperoleh dari citra hasil *resize* dapat dipetakan kembali ke sistem koordinat citra asli. Melalui proses ini, hasil segmentasi yang diperoleh pada citra berukuran lebih kecil tetap dapat digunakan secara akurat pada citra asli tanpa menyebabkan perubahan proporsi maupun pergeseran posisi objek.

2. Operasi Morfologi

Operasi morfologi merupakan salah satu teknik dalam pengolahan citra digital yang digunakan untuk memperbaiki hasil segmentasi objek. Teknik ini bekerja dengan memproses bentuk dan struktur objek pada citra biner menggunakan elemen struktur (*structuring element*), yaitu suatu pola atau matriks yang digunakan sebagai acuan untuk memeriksa dan memodifikasi piksel pada citra.

Melalui proses tersebut, bagian-bagian kecil yang tidak diinginkan dapat dihilangkan, sementara bentuk objek dapat dibuat lebih utuh dan jelas. Operasi morfologi terdiri dari beberapa jenis, yaitu *erosion*, *dilation*, *opening*, dan *closing*. Setiap operasi memiliki fungsi yang berbeda dalam memperbaiki kualitas hasil segmentasi, seperti mengurangi noise, memperbaiki kontur objek, dan mengisi

celah-celah kecil pada objek (Hartono dkk., 2019). Oleh karena itu, operasi morfologi sering digunakan sebagai tahap penyempurnaan hasil segmentasi sebelum dilakukan proses lebih lanjut.

3. Convex Hull

Convex hull didefinisikan sebagai poligon cembung terkecil yang menutupi semua titik kontur. Analoginya adalah membentangkan karet gelang mengelilingi titik-titik sehingga terbentuk batas luar yang cembung. Luas kontur dan luas hull ini digunakan untuk menghitung metrik *solidity*, yang secara matematis dinyatakan sebagai :

$$Solidity = \frac{Contour Area}{Convex hull Area} \quad (2.5)$$

Karena area *convex hull* selalu sama atau lebih besar daripada area kontur, nilai *solidity* berada pada rentang [0-1]. Nilai yang mendekati 1 menunjukkan kontur relatif cembung (sedikit lekukan), sehingga pengisian mengikuti kontur asli dianjurkan untuk mempertahankan detail bentuk. Sebaliknya, nilai *solidity* yang kecil menunjukkan kontur sangat berlekuk atau terfragmentasi, dalam kasus ini pengisian berdasarkan *convex hull* (mengisi *hull*, bukan kontur asli) sering kali menghasilkan masker yang lebih stabil dan lebih kebal terhadap noise pada tepi.

4. Smoothing (Median Blur)

Tahap *smoothing* dilakukan sebagai proses akhir pada *postprocessing* untuk mengurangi *noise* kecil seperti *salt-and-pepper noise* serta menghaluskan batas hasil segmentasi agar terlihat lebih natural. Pada penelitian ini digunakan

metode median blur, karena metode ini mampu mereduksi *noise* tanpa menyebabkan pengaburan tepi objek secara berlebihan.

Median blur bekerja dengan mengganti nilai suatu piksel berdasarkan nilai tengah (*median*) dari kumpulan intensitas piksel di sekitarnya. Pendekatan ini efektif untuk menghilangkan *noise* impulsif dan partikel kecil pada hasil segmentasi biner, sekaligus mempertahankan struktur tepi objek agar tetap jelas (Shah dkk., 2022). Oleh karena itu, *median blur* banyak digunakan pada tahap akhir pengolahan citra untuk menghasilkan mask segmentasi yang lebih halus, stabil, dan tidak terlalu kaku.

2.1.7 Intersection over Union (IoU)

Intersection over Union atau *IoU* merupakan metrik yang banyak digunakan untuk mengevaluasi hasil segmentasi objek pada citra digital. *IoU* mengukur tingkat kesesuaian antara hasil segmentasi yang dihasilkan sistem dengan anotasi referensi (*ground truth*) berdasarkan perbandingan area yang saling tumpang tindih terhadap total area gabungan keduanya. Metrik ini banyak digunakan dalam penelitian segmentasi modern karena memberikan ukuran langsung mengenai kesamaan bentuk objek yang terdeteksi dengan objek sebenarnya (Ijaz dkk., 2024; Rezaatofghi dkk., 2019)

Pada tugas segmentasi citra, fokus evaluasi terletak pada kemampuan sistem dalam memisahkan area objek dari latar belakang. Oleh karena itu, *IoU* sering digunakan karena pengukuran dilakukan berdasarkan area objek, bukan berdasarkan keseluruhan piksel citra. Pendekatan ini membuat *IoU* lebih sesuai

dibandingkan *accuracy*, terutama pada dataset segmentasi yang memiliki jumlah piksel latar belakang jauh lebih besar daripada piksel objek (Rezatofighi dkk., 2019).

Secara matematis, *IoU* didefinisikan sebagai perbandingan antara luas irisan dan luas gabungan dari dua himpunan piksel. Jika P menyatakan himpunan piksel hasil prediksi dan G menyatakan himpunan piksel *ground truth*, maka *IoU* dirumuskan sebagai:

$$IoU = \frac{|P \cap G|}{|P \cup G|} \quad (2.6)$$

Dalam implementasi pada citra biner, perhitungan irisan dan gabungan dilakukan melalui operasi logika pada piksel. Jika dikaitkan dengan *confusion matrix* pada segmentasi biner, *IoU* dapat dinyatakan menggunakan komponen *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) sebagai berikut:

$$IoU = \frac{TP}{TP + FP + FN} \quad (2.7)$$

$$Mean IoU = \frac{1}{N} \sum_{i=1}^N IoU_i \quad (2.8)$$

Persamaan 2.7 menunjukkan bahwa *IoU* tidak memasukkan *True Negative* (TN) ke dalam perhitungan. Hal ini dilakukan karena pada sebagian besar citra segmentasi, jumlah piksel latar belakang sangat dominan. Jika TN ikut diperhitungkan, nilai metrik dapat menjadi tinggi meskipun kualitas segmentasi objek sebenarnya kurang baik. Oleh karena itu, *IoU* dianggap lebih tepat digunakan dalam evaluasi segmentasi dibandingkan *accuracy* yang sangat dipengaruhi oleh

dominasi piksel non-objek (Ijaz dkk., 2024). Sedangkan untuk Nilai Mean IoU dapat dihitung menggunakan Persamaan 2.8.

Dalam praktik pelaporan *IoU*, terdapat beberapa kondisi khusus yang perlu diperhatikan. Apabila mask prediksi dan *mask ground truth* sama-sama tidak mengandung objek, maka luas gabungan bernilai nol sehingga *IoU* tidak dapat dihitung secara langsung. Dalam banyak penelitian, kondisi ini ditangani dengan menetapkan nilai *IoU* sebesar 1.0 karena prediksi dan *ground truth* dianggap sepenuhnya sesuai. Sebaliknya, apabila *ground truth* kosong tetapi prediksi menghasilkan area objek, maka *IoU* umumnya ditetapkan bernilai 0.0 karena sistem menghasilkan kesalahan deteksi (Rezatofighi dkk., 2019).

2.2. State of the art

State of the art memuat ringkasan penelitian terdahulu yang relevan dengan topik penelitian ini. Informasi diperoleh melalui studi literatur dari jurnal, artikel ilmiah, laporan penelitian, dan sumber terpercaya lainnya. Studi ini bertujuan untuk memahami perkembangan metode yang telah ada, mengidentifikasi celah penelitian, serta menyusun dasar teori yang mendukung penelitian. Ringkasan penelitian terkait disajikan pada Tabel 2.1.

Tabel 2.1 *State of the art* Penelitian

No	Peneliti	Aplikasi	Metode	Hasil penelitian
1	(Choi & Ha, 2025)	Pendeteksian tepi adaptif (umum, aplikasi pengawasan/segmentasi)	<i>Adaptive Thresholding & Canny Edge Detection</i>	Menerapkan skema adaptif untuk menyeleksi <i>threshold</i> Canny secara otomatis (menggunakan actor-critic / algoritma RL). Laporan eksperimen menunjukkan metode ini bisa memilih <i>threshold</i> yang sesuai per- <i>image</i> dan meningkatkan kualitas citra tepi dibandingkan pengaturan <i>threshold</i> manual.
2	(Prayogi dkk., 2025)	Segmentasi objek sederhana dalam citra digital	<i>Image processing, Canny Edge Detection, & Otsu Thresholding</i>	Penerapan konversi <i>grayscale</i> , <i>filter bilateral</i> , lalu Canny dan Otsu meningkatkan ketajaman tepi dan kejernihan objek utama pada citra, sehingga segmen objek menjadi lebih jelas dan utuh.

3	(Xu dkk., 2020)	Pemahaman citra dokumen (formulir dan kuitansi)	<i>Image processing & deep learning</i>	Memperkenalkan LayoutLM, sebuah model pre-training yang mengintegrasikan informasi teks dan layout. Pendekatan ini meningkatkan performa secara signifikan, misalnya skor <i>form understanding</i> meningkat dari 70,72 menjadi 79,27, serta menghasilkan perbaikan pada klasifikasi dokumen dan ekstraksi informasi pada kuitansi.
4	(Tropin dkk., 2020)	Deteksi dokumen pada perangkat <i>mobile</i>	<i>Image processing</i>	Menawarkan modifikasi metode kontur klasik dengan memasukkan perhitungan kontras antara area dalam dan luar dokumen. Pendekatan ini mengurangi kesalahan pengurutan hipotesis sebesar 40% dan menurunkan

				kesalahan deteksi keseluruhan sebesar 10%.
5	(Oliveira dkk., 2019)	Segmentasi dokumen historis	CNN, FCN/U-Net, <i>Image processing</i>	Memperkenalkan <i>dhSegment</i> , sebuah arsitektur CNN untuk melakukan segmentasi dokumen historis (Sejarah/kuno) secara <i>pixel-wise</i> . Metode ini efektif dalam memisahkan elemen penting seperti teks dan gambar, meskipun terdapat <i>noise</i> , sehingga mendukung digitalisasi dan analisis dokumen sejarah secara lebih optimal.
6	(Y. Li dkk., 2024)	Inspeksi kualitas permukaan suku cadang	<i>Adaptive selective edge detection</i> (filter campuran + model deteksi tepi adaptif 4-parameter)	Metode deteksi tepi adaptif ini (filter selektif + parameter yang diatur otomatis) mampu menekan noise dan mendeteksi tepi dengan stabil. Penelitian menunjukkan kinerjanya melampaui teknik klasik

				(Sobel/Canny) dan tetap akurat pada kondisi <i>noise</i> beragam.
7	(Nugroho dkk., 2019)	Segmentasi <i>optic cup</i> (diagnosis glaukoma)	<i>Image Preprocessing, Adaptive Thresholding, convex hull, morphology.</i>	Segmentasi dengan <i>adaptive threshold</i> dilanjutkan operasi morfologi dan <i>Convex Hull</i> menghasilkan akurasi sangat tinggi ($\approx 99.50\%$) dan spesifisitas $\approx 99.75\%$ pada deteksi <i>optic cup</i> .
8	(Kurniastuti dkk., 2025)	Segmentasi citra kuku jari (gambar medis)	<i>Adaptive Thresholding</i> dan <i>morphology closing</i> , (K-means)	Metode gabungan <i>Adaptive Thresholding</i> dan <i>closing</i> morfologi mencapai akurasi deteksi 0.97 (vs 0.82 pada Canny) dan F1-score 0.99 (vs 0.88 Canny), menandakan performa lebih baik daripada Sobel/Canny tradisional.
9	(K. Zhao dkk., 2021)	Deteksi garis semantik pada gambar natural	CNN & <i>Hough Transform</i>	Menyajikan kerangka kerja <i>end-to-end</i> yang mengintegrasikan CNN dengan <i>Hough Transform</i> untuk

				mendeteksi garis utama pada gambar secara <i>real-time</i> . Sehingga dapat menghasilkan garis yang lebih bersih dan presisi, serta dapat digunakan dalam aplikasi desain dan visual lainnya.
10	(Q. Huang & Liu, 2021)	Deteksi marka jalur pada jalan	<i>Hough Transform & image processing</i>	Melakukan evaluasi mendalam terhadap keterbatasan algoritma <i>Hough Transform</i> dalam kondisi jalan yang kompleks. Studi ini mengidentifikasi kendala utama seperti variasi pencahayaan dan gangguan dari objek lain, serta mengusulkan solusi berupa pembatasan rentang pencarian dan klasifikasi pencahayaan untuk meningkatkan akurasi deteksi marka jalur.

11	(S. Zhang dkk., 2025)	Segmentasi area kaki (foot pressure)	Hybrid: deteksi tepi Canny dan <i>geometric partitioning</i>	Kombinasi Canny dengan partisi geometris menghasilkan segmentasi area kaki yang akurat dan kontur objek yang baik. Hasil penelitian menunjukkan presisi tinggi dan pembentukan batas objek yang baik dengan pendekatan <i>hybrid</i> ini.
12	(Arai, 2025)	Konversi citra SAR menjadi citra visual untuk deteksi area longsor.	Canny & <i>image processing</i>	Metode yang diusulkan meningkatkan ketajaman dan kejelasan gambar visual hasil konversi dari citra SAR, sehingga detail seperti batas area longsor lebih mudah dikenali.
13	(X. Huang dkk., 2022)	Segmentasi citra medis	<i>Distance Transform and Thresholding</i>	Mengusulkan <i>framework</i> segmentasi dua tahap berbasis <i>distance transform</i> yang mampu meningkatkan pemisahan objek dan

				latar belakang, terutama pada area dengan batas yang tidak jelas.
14	(Maksimovic dkk., 2025)	Deteksi tepi umum / perbaikan <i>threshold</i>	<i>Adaptive edge threshold estimation</i>	Menawarkan pendekatan baru untuk memperkirakan <i>threshold</i> deteksi tepi secara adaptif berdasarkan karakteristik gradien citra. Hasil eksperimen melaporkan peningkatan kontinuitas tepi dan pengurangan tepi terputus pada dataset uji.
15	(Kabir dkk., 2022)	Segmentasi objek asing dalam robotika terdistribusi.	<i>Canny Edge Detection, Otsu thresholding, flood-fill, & bounding box</i>	<i>Pipeline</i> segmentasi tanpa pengawasan yang menggabungkan deteksi tepi Canny dan <i>thresholding</i> Otsu (diikuti <i>flood-fill</i> dan <i>bounding box</i>) berhasil memisahkan objek dari latar secara efektif dengan kompleksitas komputasi rendah.

16	(Song dkk., 2022)	<i>Segmentasi plankton / objek kompleks berkontras rendah</i>	<i>Thresholding Sauvola dua tahap (2-stage adaptive binarization)</i>	Metode dua tahap berbasis Sauvola secara signifikan meningkatkan integritas objek kompleks (misal ubur-ubur) pada citra rendah kontras dibanding <i>threshold</i> konvensional. Hasilnya objek diekstraksi lebih utuh, <i>robust</i> terhadap <i>noise</i> , dan efisiensi komputasi meningkat.
----	-------------------	---	---	---

Berbagai penelitian telah dilakukan untuk menjawab pertanyaan mendasar yang menjadi dasar penelitian ini. Sejumlah penelitian sebelumnya telah berfokus pada teknik mendeteksi menggunakan metode *deep learning*. Penelitian-penelitian tersebut terutama mengevaluasi seberapa akurat model *deep learning* dalam mengenali dan menandai bagian-bagian penting dalam dokumen.

Dalam penelitian ini, beberapa parameter utama yang akan digunakan meliputi tingkat akurasi dalam deteksi tepi, seberapa tepat segmentasi terhadap objek, serta kemampuan pemrosesan gambar (*image processing*) untuk menghilangkan gangguan atau *noise* yang dapat mengurangi kualitas hasil deteksi.