

Algoritma Heuristik untuk Pengolahan Citra: Implementasi Algoritma A* dalam Segmentasi Citra

Derwin Rustanly - 13522115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): drustanly@gmail.com

Segmentasi citra adalah langkah penting dalam pengolahan citra yang mempengaruhi berbagai aplikasi, mulai dari pengenalan objek hingga analisis medis. Makalah ini mengeksplorasi penggunaan algoritma A* berbasis heuristik untuk meningkatkan hasil segmentasi citra. Algoritma A* dengan heuristik Euclidean distance diimplementasikan untuk memperhalus segmentasi awal yang diperoleh melalui *K-means clustering*. Hasil eksperimen menunjukkan bahwa penggunaan A* dapat secara signifikan memperbaiki akurasi dan ketepatan batas-batas segmen dalam citra. Algoritma A* menunjukkan kemampuan untuk mengatasi noise dan ketidakakuratan dalam segmentasi awal, menghasilkan citra yang lebih tersegmentasi secara konsisten dan akurat. Makalah ini memberikan wawasan tentang efektivitas algoritma A* dalam pengolahan citra dan potensinya untuk diterapkan dalam berbagai aplikasi pengolahan citra yang membutuhkan segmentasi presisi tinggi.

Keywords—segmentasi citra, A*, Klaster, K-means

I. PENDAHULUAN

Segmentasi citra merupakan proses esensial dalam pengolahan citra yang bertujuan untuk membagi citra menjadi beberapa bagian atau segmen yang lebih mudah dianalisis. Segmentasi yang akurat memungkinkan identifikasi objek atau fitur tertentu dalam citra, yang sangat penting dalam berbagai aplikasi seperti pengenalan objek, pengawasan, analisis medis, dan pemrosesan video. Kualitas segmentasi citra sangat mempengaruhi hasil akhir dari aplikasi-aplikasi tersebut, sehingga penelitian dalam bidang ini terus berkembang untuk menemukan metode yang lebih efektif dan efisien.

Algoritma A* merupakan salah satu algoritma pencarian jalur yang populer dalam komputasi dan sering digunakan dalam berbagai aplikasi, termasuk permainan, robotika, dan pengolahan citra. A* menggunakan heuristik untuk mengevaluasi jalur terpendek dari titik awal ke titik tujuan. Dalam konteks segmentasi citra, A* dapat digunakan untuk memperhalus hasil segmentasi awal dengan mengoptimalkan batas-batas segmen. Penggunaan heuristik Euclidean distance dalam algoritma A* membantu dalam menentukan jalur yang paling efisien berdasarkan jarak geometris antara piksel-piksel dalam citra.

Implementasi algoritma A* dalam segmentasi citra dengan heuristik Euclidean distance memiliki potensi untuk memperbaiki hasil segmentasi yang diperoleh melalui metode clustering seperti K-means. K-means clustering adalah metode

yang sering digunakan untuk segmentasi awal karena kesederhanaannya dan kemampuannya dalam mengelompokkan piksel berdasarkan warna atau intensitas. Namun, hasil segmentasi K-means seringkali tidak sempurna, dengan batas-batas segmen yang kasar dan kurang akurat. Algoritma A* dapat digunakan untuk memperhalus batas-batas ini, menghasilkan segmen yang lebih konsisten dan akurat.

Penelitian ini bertujuan untuk mengeksplorasi efektivitas algoritma A* dalam memperbaiki hasil segmentasi citra. Dengan menerapkan A* pada hasil segmentasi K-means, diharapkan dapat dicapai peningkatan dalam ketepatan dan kualitas segmentasi. Analisis ini penting untuk menunjukkan potensi algoritma A* dalam pengolahan citra, khususnya dalam aplikasi-aplikasi yang membutuhkan segmentasi presisi tinggi. Penelitian ini juga bertujuan untuk memberikan wawasan tentang bagaimana heuristik yang tepat dapat mempengaruhi kinerja algoritma segmentasi citra, serta mengidentifikasi area potensial untuk pengembangan lebih lanjut.

II. LANDASAN TEORI

A. Graf

Graf merupakan suatu representasi dari kumpulan objek diskrit dan relasi yang terdapat di antara objek-objek tersebut. Suatu graf memiliki 2 komponen utama, yakni simpul (*vertex*) dan sisi (*edges*). Suatu graf didefinisikan sebagai suatu tuple berelemen 2 dengan persamaan matematis berikut.

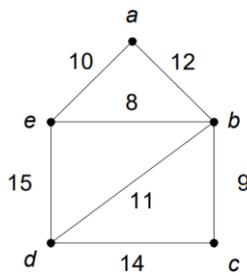
$$G = (V, E)$$

dengan V yang merupakan himpunan simpul yang terdapat pada graf dan E yang merupakan himpunan sisi yang menghubungkan sepasang simpul, atau dapat dituliskan sebagai

$$V = \{v_1, v_1, \dots, v_n\} \quad E = \{e_1, e_1, \dots, e_n\}$$

Graf berbobot adalah struktur data yang terdiri dari himpunan simpul (atau *vertex*) dan himpunan sisi (atau *edge*) di mana setiap sisi diberi suatu nilai yang disebut bobot. Dalam notasi matematis, Pada graf berbobot, diperkenalkan fungsi bobot $w : E \rightarrow \mathbb{R}$, yang memberikan bobot atau nilai kepada

setiap sisi $e \in E$. Nilai ini bisa mewakili jarak, biaya, kapasitas, atau ukuran lain tergantung pada konteks aplikasi graf tersebut.



Gambar 1. Graf Berbobot (Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)

B. Heuristik

Heuristik merupakan teknik yang digunakan untuk menemukan solusi yang cukup baik, cepat, atau praktis dalam waktu yang wajar, meskipun tidak selalu menjamin solusi yang paling optimal. Dalam pengambilan keputusan, pengembangan algoritma, dan pemecahan masalah, pendekatan heuristik sering digunakan untuk mempercepat proses pencarian solusi dalam situasi dimana metode eksak yang mungkin menghabiskan waktu terlalu lama atau tidak praktis untuk diterapkan. Heuristik khususnya efektif dalam menangani masalah yang kompleks dan berukuran besar, seperti dalam pencarian rute, optimasi jadwal, dan segmentasi citra, di mana sebuah solusi yang sempurna adalah sulit atau sangat mahal untuk ditemukan.

Dalam aplikasi nyata, heuristik dapat memanfaatkan aturan praktis atau intuisi yang berfokus pada fitur-fitur penting dari masalah untuk menghasilkan solusi yang memadai dengan lebih cepat. Misalnya, dalam algoritma pencarian, heuristik seperti algoritma A* menggunakan fungsi heuristik untuk memperkirakan jarak terdekat dari keadaan saat ini ke tujuan akhir, memungkinkan algoritma tersebut bergerak secara efisien menuju solusi tanpa harus mengeksplorasi setiap kemungkinan secara detail. Meskipun solusi yang dihasilkan mungkin tidak optimal, efisiensi dan kecepatan yang ditawarkan membuat heuristik menjadi pilihan yang sangat berharga, terutama di bawah keterbatasan waktu atau sumber daya.

C. Algoritma Uniform Cost Search (UCS)

Algoritma Uniform Cost Search (UCS) dikategorikan sebagai metode Uninformed Search, yang tidak mengandalkan informasi heuristik atau data eksternal di luar struktur graf yang diberikan. Karakteristik utama dari UCS adalah operasinya yang berbasis pada biaya kumulatif dari simpul asal ke simpul yang sedang dievaluasi, yang dinotasikan dengan fungsi biaya $g(n)$. UCS melakukan ekspansi terhadap simpul yang memiliki total biaya terendah terakumulasi, menjadikan algoritma ini efektif untuk menjamin pencarian solusi dengan biaya minimum tanpa mempertimbangkan estimasi jarak atau biaya menuju tujuan.

Dalam penerapannya, UCS menggunakan struktur data seperti antrian prioritas untuk mempertahankan pencarian pada

jalur dengan biaya terendah. Setiap simpul yang diekspansi dalam graf akan ditambahkan ke dalam antrian prioritas berdasarkan nilai $g(n)$, yang merefleksikan total biaya dari simpul awal hingga simpul tersebut. Algoritma ini terus mengulangi proses ekspansi simpul hingga mencapai tujuan atau hingga semua kemungkinan jalur telah dijelajahi. Kelebihan UCS terletak pada kemampuannya untuk secara sistematis mengeksplorasi dan menemukan solusi optimal dengan mengabaikan jalur yang memiliki total biaya lebih tinggi, menjadikannya pilihan yang *robust* untuk pencarian jalur dalam graf berbobot tanpa siklus negatif.

D. Algoritma Greedy Best First Search (Greedy Best First Search)

Algoritma Greedy Best First Search (GBFS) adalah metode pencarian jalur yang dikategorikan sebagai Informed Search, yang memanfaatkan estimasi heuristik, ditandai sebagai $h(n)$ untuk memproyeksikan biaya dari simpul ke- n ke simpul tujuan. Berbeda dengan Uniform Cost Search yang berfokus pada biaya kumulatif yang sudah dikeluarkan, GBFS menggunakan nilai heuristik sebagai dasar untuk mengembangkan simpul. Hal ini berarti algoritma GBFS mengutamakan ekspansi simpul berdasarkan estimasi biaya terkecil menuju tujuan, bukan berdasarkan biaya yang telah dikeluarkan. Seperti UCS, GBFS juga memanfaatkan struktur data priority queue untuk mengatur simpul-simpul yang aktif, namun simpul-simpul tersebut diurutkan berdasarkan nilai heuristiknya, bukan biaya total mereka.

Dari perspektif teoretis, GBFS mengadopsi pendekatan yang serakah—seperti yang tersirat dari namanya—dalam mencari solusi. Pendekatan ini berarti bahwa algoritma secara konsisten memilih simpul yang tampak paling menjanjikan pada setiap langkah, berdasarkan heuristik yang tersedia, tanpa mempertimbangkan konsekuensi dari pilihan tersebut terhadap seluruh solusi. Pendekatan ini sering kali mengarah pada penemuan solusi lokal yang optimal namun tidak selalu menghasilkan solusi optimal global. Akibatnya, meskipun GBFS dapat lebih cepat dalam mencapai simpul tujuan daripada algoritma yang tidak menggunakan informasi heuristik, algoritma ini dapat mengabaikan jalur yang pada akhirnya lebih efektif karena terpaku pada opsi yang terlihat menguntungkan secara segera. Ini menggambarkan limitasi utama GBFS: meskipun efektif dalam banyak skenario, tidak dapat dijamin bahwa solusi yang ditemukannya merupakan solusi yang terbaik secara keseluruhan.

E. Algoritma A*

Algoritma A* merupakan salah satu pendekatan dalam Informed Search yang mengintegrasikan estimasi heuristik $h(n)$ dengan pertimbangan biaya sebenarnya $g(n)$ untuk menghasilkan fungsi evaluasi $f(n)=g(n)+h(n)$. Fungsi $g(n)$ mengukur biaya aktual dari simpul awal ke simpul n , sementara $h(n)$ memberikan estimasi biaya dari simpul n ke simpul tujuan. Dalam konteks pencarian yang efisien, keadilan atau 'admissibility' dari heuristik sangat penting. Heuristik dikatakan admissible jika nilai estimasi $h(n)$ tidak pernah melebihi biaya sebenarnya dari simpul n ke tujuan, yaitu $h(n) \leq h^*(n)$. Dengan memastikan bahwa heuristik adalah

admissible, algoritma A* dapat menjamin bahwa solusi yang ditemukan adalah optimal.

Algoritma A* menjamin optimalitas dari pencarian apabila nilai heuristik yang digunakan bersifat admissible. Secara teoritis, algoritma A* dengan estimasi heuristik yang admissible akan cenderung lebih optimal dibandingkan dengan algoritma UCS, karena algoritma A* menghindari ekspansi dari simpul yang terlalu “mahal” berdasarkan penjumlahan dari nilai cost dari simpul akar beserta nilai heuristik menuju simpul tujuan.

F. Segmentasi Citra

Segmentasi citra adalah proses fundamental dalam pengolahan citra yang memisahkan atau membagi citra digital menjadi berbagai segmen atau wilayah yang memiliki karakteristik serupa. Tujuan utamanya adalah untuk memudahkan analisis lebih lanjut dengan mengorganisir informasi visual dalam citra menjadi komponen yang lebih sederhana dan berarti. Dalam aplikasi praktis, segmentasi citra digunakan untuk mengidentifikasi objek dan batasan dalam citra, yang penting dalam berbagai aplikasi seperti pengenalan wajah, pemantauan medis, pengolahan citra satelit, dan navigasi otomatis kendaraan.

Proses segmentasi citra membagi piksel-piksel dalam citra menjadi kelompok-kelompok yang homogen berdasarkan kriteria seperti intensitas warna, tekstur, atau kedekatan spasial. Misalnya, dalam pengolahan citra medis, segmentasi dapat digunakan untuk membedakan antara jaringan sehat dan jaringan yang terkena penyakit, atau dalam penginderaan jauh untuk memisahkan berbagai jenis penggunaan lahan. Pendekatan yang digunakan untuk segmentasi citra dapat bervariasi dari metode sederhana berbasis ambang batas (thresholding), yang membagi piksel berdasarkan nilai intensitasnya, hingga teknik lebih kompleks seperti clustering, edge detection, dan region growing.

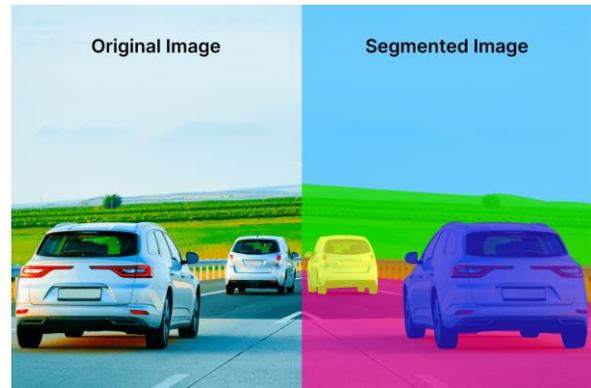
Teknik thresholding adalah salah satu metode segmentasi yang paling sederhana dan efektif, terutama bila objek dan latar belakang memiliki kontras yang tinggi. Metode ini menetapkan batasan nilai intensitas, di mana piksel yang melebihi batasan tersebut dianggap sebagai bagian dari objek, sementara yang lain merupakan latar belakang. Meskipun mudah diimplementasikan dan cepat, pendekatan ini seringkali tidak efektif dalam citra dengan tingkat kontras yang rendah atau dengan variasi intensitas yang kompleks.

Pendekatan yang lebih canggih, seperti clustering, melibatkan pengelompokan piksel berdasarkan kesamaan fitur menggunakan algoritma seperti K-means atau Fuzzy C-means. Piksel-piksel dengan karakteristik serupa dikelompokkan bersama untuk membentuk segmen. Teknik ini lebih fleksibel dan dapat digunakan pada citra yang lebih kompleks, tetapi memerlukan pemrosesan yang lebih intensif dan mungkin membutuhkan penyesuaian parameter yang cermat untuk mendapatkan hasil yang optimal.

Selain itu, metode berbasis edge detection mengidentifikasi batas atau tepi antara wilayah dengan menggunakan teknik seperti operator Sobel, Canny, atau detektor tepi Laplacian. Metode ini efektif untuk citra yang memiliki definisi batas

objek yang jelas. Alternatif lain, region growing, adalah teknik yang memulai dengan seed point dan menambahkan piksel yang bersebelahan yang memiliki properti serupa ke dalam region yang sama. Teknik ini sangat berguna untuk segmentasi berbasis area di mana keseragaman wilayah lebih diutamakan.

Secara keseluruhan, pemilihan metode segmentasi citra sangat bergantung pada jenis citra yang ditangani dan kebutuhan aplikasi spesifik. Setiap teknik memiliki kekuatan dan keterbatasannya, dan dalam banyak kasus, kombinasi dari beberapa pendekatan dapat memberikan hasil yang lebih baik. Efektivitas segmentasi secara langsung mempengaruhi keberhasilan aplikasi pengolahan citra lebih lanjut, membuatnya menjadi salah satu langkah kritis dalam banyak sistem visi komputer modern.



Gambar 2. Ilustrasi Segmentasi Citra (<https://deeplobe.ai/image-segmentation-the-most-interesting-applications/>)

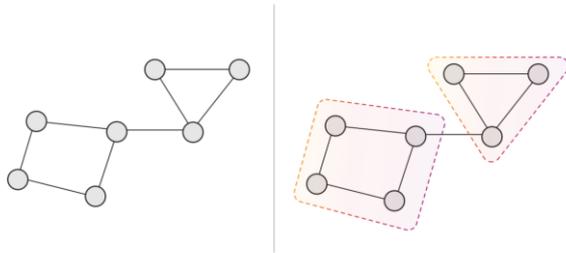
G. Klasterisasi Graf

Klasterisasi graf merupakan metode yang digunakan untuk mengelompokkan simpul-simpul dalam graf berdasarkan pola keterhubungan di antara simpul-simpul tersebut. Tujuan utama dari klasterisasi graf adalah untuk mengidentifikasi kelompok-kelompok simpul yang memiliki hubungan lebih erat satu sama lain dibandingkan dengan simpul-simpul di luar kelompok tersebut. Kelompok simpul tersebut merepresentasikan struktur komunitas atau modul dalam graf, di mana simpul-simpul dalam satu klaster memiliki banyak tepi yang menghubungkan mereka, menciptakan subgraf yang lebih padat.

Prinsip dasar klasterisasi graf melibatkan analisis struktur graf untuk menemukan pola keterhubungan yang signifikan. Dalam graf, simpul-simpul yang berada dalam satu klaster cenderung memiliki lebih banyak hubungan satu sama lain daripada dengan simpul-simpul di klaster lain. Dengan menemukan dan menganalisis subgraf-subgraf yang padat ini, dapat diungkap struktur internal dari data yang diwakili oleh graf tersebut. Misalnya, dalam jaringan sosial, klasterisasi graf dapat mengidentifikasi kelompok-kelompok individu yang memiliki interaksi lebih sering di antara mereka.

Proses klasterisasi graf dapat melibatkan beberapa langkah, seperti membangun graf dari data, memilih ukuran kesamaan atau keterhubungan, dan kemudian menerapkan algoritma klasterisasi untuk mengidentifikasi klaster. Ukuran kesamaan atau keterhubungan ini bisa berdasarkan jarak, kekuatan hubungan, atau metrik lainnya yang relevan dengan konteks

graf. Hasil akhir dari klusterisasi graf adalah pembagian simpul-simpul graf menjadi kluster-kluster yang menggambarkan struktur hubungan atau komunitas yang tersembunyi dalam data.



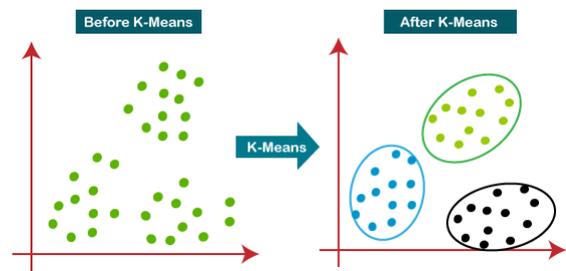
Gambar 3. Ilustrasi Klusterisasi Graf (Sumber: <https://memgraph.com/>)

H. Algoritma K-means

Algoritma K-means adalah metode klusterisasi yang terkenal dan banyak digunakan dalam pengolahan data, termasuk segmentasi citra. Algoritma ini bertujuan untuk membagi data ke dalam sejumlah kluster k yang telah ditentukan sebelumnya, di mana setiap kluster didefinisikan oleh pusat atau centroid yang merupakan rata-rata dari semua titik data dalam kluster tersebut. Proses klusterisasi dimulai dengan memilih k pusat kluster secara acak. Setiap titik data kemudian dialokasikan ke kluster terdekat berdasarkan jarak Euclidean dari pusat kluster.

Setelah alokasi awal, pusat kluster diperbarui dengan menghitung rata-rata posisi semua titik data dalam masing-masing kluster. Proses ini berulang dalam iterasi, dengan setiap iterasi mengalokasikan ulang titik data ke pusat kluster yang baru dan memperbarui pusat kluster hingga tidak ada perubahan signifikan dalam alokasi titik data atau posisi pusat kluster. Iterasi ini berlanjut hingga algoritma mencapai konvergensi, yaitu ketika alokasi titik data ke kluster tidak lagi berubah atau perubahan yang terjadi sangat minimal. Konvergensi dicapai ketika perbedaan antara pusat kluster pada iterasi sebelumnya dan iterasi saat ini berada di bawah ambang batas yang telah ditentukan.

K-means dikenal karena kesederhanaan dan efisiensinya dalam mengelompokkan data besar. Namun, algoritma ini memiliki beberapa keterbatasan. Salah satu keterbatasan utama adalah ketergantungan pada nilai k yang telah ditentukan sebelumnya, yang mungkin tidak selalu diketahui atau mudah ditentukan. Selain itu, hasil akhir dari K-means dapat dipengaruhi oleh inisialisasi pusat kluster awal, yang berarti bahwa hasil klusterisasi dapat berbeda pada setiap eksekusi jika pusat kluster awal dipilih secara acak. Untuk mengatasi masalah ini, metode inisialisasi yang lebih canggih seperti K-means++ dapat digunakan untuk memilih pusat kluster awal dengan lebih baik. Meskipun demikian, K-means tetap menjadi salah satu algoritma klusterisasi yang paling populer dan banyak digunakan karena kesederhanaan, kecepatan, dan kemampuannya untuk menangani dataset besar dengan efisien.



Gambar 4. Algoritma K-Means (Sumber: <https://www.javatpoint.com/>)

III. IMPLEMENTASI

Secara keseluruhan, penerapan algoritma A* dalam proses segmentasi citra akan diimplementasikan dalam beberapa tahapan, yaitu sebagai berikut.

A. Pemrosesan Awal Citra

Pemrosesan awal citra merupakan langkah fundamental yang mempersiapkan citra untuk tahap segmentasi berikutnya. Langkah pertama dalam proses ini adalah pemuatan citra, yaitu mengimpor citra dari sumber data yang telah ditentukan. Sumber data ini dapat berupa file citra dalam berbagai format seperti JPEG, PNG, atau BMP. Proses pemuatan ini penting untuk memastikan bahwa citra tersedia dalam format yang kompatibel dengan algoritma yang akan digunakan selanjutnya.

Setelah citra berhasil dimuat, langkah berikutnya adalah konversi warna dari ruang warna BGR (Blue, Green, Red) ke ruang warna RGB (Red, Green, Blue) menggunakan pustaka OpenCV. Konversi ini penting karena banyak algoritma pengolahan citra yang bekerja lebih optimal dalam ruang warna RGB. Ruang warna RGB lebih sesuai dengan persepsi manusia terhadap warna dan sering digunakan dalam berbagai aplikasi pengolahan citra.

Tahap selanjutnya adalah normalisasi piksel, yang bertujuan untuk mengubah nilai piksel dari rentang 0-255 menjadi rentang 0.0-1.0. Proses ini penting untuk memastikan bahwa data piksel berada dalam skala yang sesuai untuk analisis selanjutnya oleh algoritma segmentasi. Normalisasi juga membantu mengurangi perbedaan intensitas yang ekstrem antara piksel, sehingga algoritma dapat lebih mudah mengidentifikasi pola dan struktur dalam citra. Dengan demikian, pemrosesan awal citra memastikan bahwa citra yang digunakan berada dalam kondisi optimal untuk segmentasi yang akurat dan efisien.

B. Segmentasi Awal dengan Algoritma K-means

Segmentasi awal dengan K-means merupakan tahap kritis dalam proses segmentasi citra. Algoritma K-means adalah metode klusterisasi yang mengelompokkan piksel citra berdasarkan kesamaan warna atau intensitas. Proses segmentasi dimulai dengan inisialisasi jumlah kluster (k) dan penentuan pusat kluster secara acak. Inisialisasi yang tepat sangat penting karena pusat kluster awal yang dipilih secara acak dapat

mempengaruhi konvergensi dan hasil akhir dari algoritma. Selama iterasi klusterisasi, setiap piksel dialokasikan ke klaster terdekat berdasarkan jarak Euclidean dari pusat klaster.

Setelah alokasi awal, pusat klaster diperbarui dengan menghitung rata-rata posisi semua piksel dalam setiap klaster. Proses ini berulang secara iteratif, dengan setiap iterasi mengalokasikan ulang piksel ke pusat klaster terdekat yang baru dan memperbarui pusat klaster hingga tidak ada perubahan signifikan dalam alokasi piksel atau posisi pusat klaster. Hasil akhir dari segmentasi K-means adalah citra yang tersegmentasi di mana setiap piksel diberi label sesuai dengan klaster yang menjadi anggotanya.

Meskipun K-means efektif dalam mengelompokkan piksel berdasarkan kesamaan warna atau intensitas, hasil segmentasinya sering kali memiliki batas-batas yang kasar dan tidak akurat. Oleh karena itu, diperlukan langkah pemurnian lebih lanjut untuk meningkatkan akurasi dan kualitas segmentasi. Proses pemurnian ini dapat dilakukan menggunakan algoritma lain yang lebih canggih atau dengan metode yang mengoptimalkan hasil segmentasi awal.

C. Perbaikan Segmentasi dengan Algoritma A*

Perbaikan segmentasi menggunakan algoritma A* bertujuan untuk memperhalus batas-batas segmen yang dihasilkan oleh K-means. Algoritma A* adalah algoritma pencarian jalur berbasis heuristik yang menggunakan Euclidean distance untuk menentukan jalur optimal antara dua titik. Dalam konteks segmentasi citra, A* digunakan untuk mengoptimalkan batas-batas segmen dengan mengevaluasi jalur antara piksel-piksel yang berada pada batas segmen. Proses ini dimulai dengan mendefinisikan fungsi heuristik Euclidean distance, yang mengukur jarak geometris antara dua piksel.

Setelah heuristik ditentukan, algoritma A* diinisialisasi dengan menyiapkan struktur data yang diperlukan, seperti open set dan closed set, serta menginisialisasi nilai g dan f untuk setiap simpul. Nilai g merepresentasikan biaya dari simpul awal ke simpul saat ini, sementara nilai h adalah estimasi biaya dari simpul saat ini ke tujuan berdasarkan heuristik. Nilai f adalah jumlah dari g dan h, yang digunakan untuk menentukan prioritas simpul dalam pencarian. Algoritma kemudian melakukan pencarian jalur optimal dengan mengevaluasi simpul-simpul tetangga dan memperbarui nilai g, h, dan f hingga mencapai tujuan.

Setelah jalur optimal ditemukan, label piksel dalam citra diperbarui berdasarkan hasil pencarian, menghasilkan batas segmen yang lebih halus dan akurat. Proses perbaikan ini memungkinkan peningkatan kualitas segmentasi dengan mengatasi ketidakakuratan dan ketidakkonsistenan yang ada pada hasil segmentasi awal. Dengan demikian, penerapan algoritma A* dalam perbaikan segmentasi citra dapat memberikan hasil yang lebih optimal dan presisi tinggi.

D. Evaluasi Hasil Segmentasi

Evaluasi hasil segmentasi merupakan tahap krusial dalam proses segmentasi citra untuk menilai kualitas dan akurasi dari hasil yang telah diperoleh. Tahap ini melibatkan penggunaan

metrik evaluasi yang tepat untuk membandingkan hasil segmentasi awal dengan hasil yang telah diperbaiki oleh algoritma A*. Dua metrik evaluasi yang umum digunakan adalah *Dice Coefficient* dan *Jaccard Index*. *Dice Coefficient* mengukur kesamaan antara dua himpunan data dengan menghitung dua kali jumlah elemen yang beririsan dibagi dengan jumlah elemen pada kedua himpunan tersebut. Di sisi lain, *Jaccard Index* mengukur kesamaan dengan membagi jumlah elemen yang beririsan dengan jumlah total elemen unik dalam kedua himpunan.

Setelah metrik evaluasi ditentukan, langkah berikutnya adalah melakukan perhitungan terhadap hasil segmentasi. *Dice Coefficient* dan *Jaccard Index* digunakan untuk mengukur kesesuaian antara hasil segmentasi awal dan hasil yang telah diperbaiki. Perhitungan ini memberikan gambaran kuantitatif tentang efektivitas algoritma A* dalam memperbaiki segmentasi. Selain itu, visualisasi hasil segmentasi juga dilakukan untuk memberikan pemahaman yang lebih intuitif tentang perubahan yang terjadi setelah perbaikan. Visualisasi ini melibatkan perbandingan citra hasil segmentasi awal dengan citra hasil perbaikan, sehingga perbedaan dan peningkatan kualitas dapat diamati secara langsung.

Analisis kinerja juga dilakukan untuk menilai efektivitas dan efisiensi algoritma A* dalam perbaikan segmentasi citra. Analisis ini melibatkan pengukuran waktu komputasi yang dibutuhkan oleh algoritma serta penilaian terhadap peningkatan akurasi yang dicapai. Hasil analisis ini penting untuk menentukan apakah algoritma A* memberikan peningkatan yang signifikan dalam kualitas segmentasi tanpa menambah beban komputasi yang berlebihan. Dengan demikian, evaluasi hasil segmentasi tidak hanya memberikan penilaian kuantitatif dan visual terhadap kualitas segmentasi, tetapi juga memberikan wawasan tentang kinerja algoritma yang digunakan dalam proses perbaikan.

E. Implementasi dalam Kode

Pada makalah ini, proses segmentasi citra dengan algoritma A* diimplementasikan dalam bahasa pemrograman Python, dengan memanfaatkan kakas pemrosesan citra berupa OpenCV. OpenCV digunakan untuk melakukan berbagai operasi pemrosesan awal citra, seperti konversi ruang warna dan normalisasi piksel, yang sangat penting untuk memastikan citra berada dalam kondisi optimal untuk segmentasi. Algoritma K-means digunakan sebagai langkah awal untuk mengelompokkan piksel citra berdasarkan kesamaan warna atau intensitas, menghasilkan citra tersegmentasi yang dapat diperbaiki lebih lanjut dengan algoritma A*.

Algoritma A* diterapkan untuk memperhalus batas-batas segmen yang dihasilkan oleh K-means. Dalam implementasi ini, struktur data antrian prioritas (*Priority queue*) digunakan untuk memproses setiap simpul selama pencarian jalur optimal. Antrian prioritas membantu mengatur simpul-simpul yang dievaluasi berdasarkan prioritas, yang ditentukan oleh kombinasi biaya aktual dari simpul awal ke simpul saat ini dan estimasi biaya dari simpul saat ini ke tujuan. Heuristik *Euclidean distance* digunakan sebagai fungsi estimasi biaya untuk membantu algoritma A* menemukan jalur yang paling efisien.

Selain itu, algoritma A* dibandingkan dengan dua metrik evaluasi utama, yaitu *Dice Coefficient* dan *Jaccard Index*, untuk menilai kualitas dan akurasi hasil segmentasi yang telah diperbaiki. Metrik-metrik ini memberikan gambaran kuantitatif tentang efektivitas algoritma A* dalam memperbaiki segmentasi citra. Visualisasi hasil segmentasi sebelum dan sesudah perbaikan juga dilakukan untuk memberikan pemahaman intuitif tentang peningkatan yang dicapai. Dengan demikian, implementasi dan evaluasi ini menunjukkan bahwa algoritma A* dapat secara signifikan meningkatkan kualitas segmentasi citra melalui pemanfaatan heuristik dan struktur data yang efisien.

IV. PENGUJIAN DAN ANALISIS

A. Hasil Pengujian



Gambar 5. Hasil Pengujian 1 untuk citra grayscale berukuran 256x256 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 6. Hasil Pengujian 2 untuk citra grayscale berukuran 512x512 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 7. Hasil Pengujian 3 untuk citra grayscale berukuran 256x256 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 8. Hasil Pengujian 3 untuk citra grayscale berukuran 256x256 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 9. Hasil Pengujian 4 untuk citra berwarna berukuran 512x512 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 10. Hasil Pengujian 5 untuk citra berwarna berukuran 512x512 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)



Gambar 11. Hasil Pengujian 6 untuk citra berwarna berukuran 512x512 Sumber: informatika.stei.itb.ac.id/~rinaldi.munir/)

B. Analisis

Berdasarkan hasil pengujian, dapat diamati bahwa penggunaan algoritma A* untuk memperbaiki batas-batas kluster hasil segmentasi citra grayscale maupun berwarna dengan algoritma K-means memberikan hasil yang memuaskan, bahkan dengan nilai k yang relatif kecil ($k = 3$). Penggunaan nilai k yang kecil dalam algoritma K-means berarti citra akan dikelompokkan menjadi sejumlah kluster yang terbatas. Nilai k yang kecil sering digunakan untuk memperoleh segmentasi yang lebih umum atau kasar, yang dapat membantu dalam hal kesederhanaan dan kejelasan visualisasi, reduksi noise, serta peningkatan kinerja komputasi.

Hasil pengujian menunjukkan bahwa penggunaan algoritma A* untuk memperbaiki segmentasi citra memberikan nilai *Dice Coefficient* dan *Jaccard Index* yang mendekati satu. Hal ini menunjukkan bahwa algoritma A* berhasil memperhalus batas-batas kluster yang dihasilkan oleh K-means, menghasilkan segmentasi yang lebih akurat dan konsisten. Nilai metrik yang mendekati satu mengindikasikan bahwa hasil perbaikan sangat mirip dengan hasil segmentasi awal, tetapi dengan peningkatan pada kejelasan batas-batas segmen.

Dengan menggunakan nilai k yang kecil, algoritma K-means dapat menghasilkan segmentasi yang sederhana dan efisien. Namun, hasil segmentasi awal ini sering kali memiliki batas-batas yang kasar. Penggunaan algoritma A* untuk memperhalus batas-batas ini terbukti efektif, sebagaimana ditunjukkan oleh nilai *Dice Coefficient* dan *Jaccard Index* yang mendekati satu. Hal ini menunjukkan bahwa algoritma A* dapat secara signifikan meningkatkan kualitas segmentasi citra, baik grayscale maupun berwarna, dengan penggunaan nilai k yang kecil. Proses ini tidak hanya meningkatkan akurasi segmentasi tetapi juga mempertahankan efisiensi komputasi, menjadikannya cocok untuk berbagai aplikasi pengolahan citra.

V. KESIMPULAN

Pada makalah ini, telah diimplementasikan dan dievaluasi pemanfaatan dari algoritma A* untuk memperbaiki hasil segmentasi citra yang diperoleh melalui algoritma K-means dengan nilai k yang kecil ($k = 3$). Hasil pengujian

menunjukkan bahwa penggunaan algoritma A* secara signifikan meningkatkan kejelasan dan akurasi batas-batas klaster dalam citra grayscale maupun berwarna, sebagaimana dibuktikan oleh nilai Dice Coefficient dan Jaccard Index yang mendekati satu. Algoritma A* mampu mengatasi keterbatasan segmentasi awal yang sering kali memiliki batas-batas yang kasar, menghasilkan segmentasi yang lebih halus dan konsisten.

Dengan demikian, dapat disimpulkan bahwa algoritma A* efektif dalam memperbaiki hasil segmentasi citra dengan nilai k yang kecil, meningkatkan kualitas dan presisi segmentasi tanpa menambah beban komputasi yang signifikan. Implementasi ini menunjukkan potensi besar untuk diterapkan dalam berbagai aplikasi pengolahan citra yang membutuhkan segmentasi presisi tinggi. Algoritma A* tidak hanya mempertahankan efisiensi komputasi tetapi juga memberikan peningkatan substansial dalam akurasi segmentasi, menjadikannya alat yang berharga dalam bidang pengolahan citra.

PENUTUP

Akhir kata, penulis memanjatkan puji syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, proses pembuatan makalah Strategi Algoritma dapat terselesaikan dengan baik. Selain itu, penulis juga ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen kelas 01 Strategi Algoritma yang telah memaparkan ilmu dan konsep Strategi Algoritma yang dapat dipahami secara mendalam oleh penulis, serta telah mewedahi dan memfasilitasi berbagai sarana pembelajaran mata kuliah Strategi Algoritma dalam bentuk sumber belajar utama maupun referensi penunjang lainnya. Penulis berharap makalah ini dapat mendatangkan manfaat khususnya dalam eksplorasi terkait analisis pengolahan citra dan dapat menjadi bahan rujukan untuk penelitian mendatang.

LAMPIRAN

Tautan repositori:

<https://github.com/DerwinRustanly/ImplementasiAStardalamSegmentasiCitra>

DAFTAR PUSTAKA

- [1] Felzenszwalb, Pedro F., dan Huttenlocher, Daniel P. 2004. Efficient Graph-Based Image Segmentation. *International journal of computer vision*, 59, 167-181.
- [2] Ulfa Maulidevi, Nur. 2024. Penentuan Rute (Route Planning) bagian 1, Informatika. Dilansir dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/>. Diakses pada tanggal 10 Juni 2024.
- [3] Ulfa Maulidevi, Nur. 2024. Penentuan Rute (Route Planning) bagian 2, Informatika. Dilansir dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/>. Diakses pada tanggal 10 Juni 2024.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024



Derwin Rustanly 13522115