

# Implementasi Algoritma SIFT untuk Pembentukan Panorama Citra Lingkungan Desa Menggunakan Metode RANSAC

Ghulam Mushthofa  
Teknik Informatika  
Universitas Darussalam Gontor  
Ponorogo  
[ghulammushthofa51@student.cs.unida.gontor.ac.id](mailto:ghulammushthofa51@student.cs.unida.gontor.ac.id)

**Abstract—** Penelitian ini bertujuan untuk mengimplementasikan dan menganalisis kinerja algoritma *Scale-Invariant Feature Transform* (SIFT) dalam aplikasi penggabungan citra (*image stitching*) untuk membentuk panorama. Objek yang digunakan adalah citra lingkungan pedesaan dengan tekstur alami yang kompleks, seperti vegetasi dan bangunan. Eksperimen dilakukan melalui tahapan deteksi *keypoints*, ekstraksi deskriptor, pencocokan fitur, dan estimasi homografi menggunakan RANSAC. Hasil pengujian menunjukkan bahwa SIFT mampu mendeteksi ribuan fitur yang invarian terhadap skala dan rotasi, serta berhasil menyatukan dua citra dengan perspektif berbeda menjadi satu panorama yang mulus (*seamless*).

**Kata Kunci:** *Computer Vision, SIFT, Image Stitching, Feature Matching, RANSAC, Panorama.*

## I. PENDAHULUAN

Dalam ranah *Computer Vision*, teknik rekonstruksi citra panorama atau *Image Stitching* merupakan salah satu topik yang esensial. Teknik ini bertujuan untuk menggabungkan beberapa citra parsial yang memiliki area tumpang tindih (*overlap*) menjadi satu kesatuan citra dengan sudut pandang lebar (*wide-angle*). Tantangan fundamental dalam proses ini terletak pada kemampuan algoritma untuk mengidentifikasi titik korespondensi yang akurat antara dua citra yang diambil dari perspektif, jarak (skala), dan kondisi pencahayaan yang berbeda.

Metode deteksi fitur konvensional sering kali memiliki keterbatasan dalam menangani perubahan skala citra. Sebuah fitur yang terlihat sebagai sudut tajam pada citra beresolusi tinggi mungkin akan terdeteksi sebagai tepi datar saat citra diperkecil, menyebabkan kegagalan dalam pencocokan. Untuk mengatasi masalah ini, algoritma *Scale-Invariant Feature Transform* (SIFT) hadir sebagai solusi yang menawarkan fitur-fitur invarian. SIFT mampu mengekstraksi deskriptor unik yang stabil terhadap rotasi, penskalaan, dan perubahan iluminasi, menjadikannya standar emas dalam tugas pencocokan fitur.

Laporan ini memaparkan implementasi algoritma SIFT untuk aplikasi penyambungan citra otomatis pada objek lingkungan pedesaan. Penggunaan dataset eksterior (*outdoor*) dengan tekstur vegetasi dan bangunan dipilih secara spesifik untuk memvalidasi ketangguhan algoritma dalam menangani kompleksitas visual alami dibandingkan dengan objek buatan yang memiliki pola geometris teratur.

## II. LANDASAN TEORI

### A. *Scale-Invariant Feature Transform (SIFT)*

Algoritma SIFT, yang dikembangkan oleh David Lowe, merupakan metode ekstraksi fitur yang dirancang untuk

menghasilkan deskriptor citra yang invarian terhadap penskalaan, rotasi, dan translasi, serta sebagian invarian terhadap perubahan iluminasi. SIFT bekerja dengan mentransformasikan data citra ke dalam koordinat varian skala relatif terhadap fitur lokal. Proses ini terdiri dari empat tahapan fundamental:

1. **Konstruksi Ruang Skala (*Scale-Space Extrema Detection*):** Tahap awal melibatkan pencarian lokasi potensial fitur pada berbagai skala. Hal ini dicapai dengan menggunakan fungsi *Difference of Gaussian* (DoG) yang efisien untuk mengidentifikasi titik-titik minat (*interest points*) yang stabil terhadap perubahan skala citra.
2. **Lokalisasi Keypoint (*Keypoint Localization*):** Pada setiap kandidat lokasi fitur, model mendetail dicocokkan untuk menentukan lokasi dan skala yang presisi. Keypoint dipilih berdasarkan ukuran stabilitasnya; titik dengan kontras rendah atau yang hanya merupakan respon tepi (*edge response*) akan dieliminasi untuk meningkatkan akurasi.
3. **Penetapan Orientasi (*Orientation Assignment*):** Satu atau lebih orientasi ditetapkan ke setiap lokasi keypoint berdasarkan arah gradien piksel lokal. Semua operasi selanjutnya dilakukan relatif terhadap orientasi, skala, dan lokasi yang telah ditetapkan ini, sehingga memberikan invariansi terhadap transformasi geometri tersebut.
4. **Deskriptor Keypoint (*Keypoint Descriptor*):** Gradien lokal citra diukur pada skala yang dipilih di sekitar keypoint. Gradien ini ditransformasikan menjadi representasi vektor 128-dimensi yang memungkinkan distorsi bentuk lokal yang signifikan dan perubahan iluminasi.

### B. *Pencocokan Fitur (Feature-Matching)*

Setelah deskriptor fitur diekstraksi dari dua citra yang berbeda, tahap selanjutnya adalah pencocokan fitur. Proses ini umumnya menggunakan metode *Brute-Force Matcher*

yang menghitung jarak Euclidean antara vektor deskriptor. Untuk meminimalkan kesalahan pencocokan (*false positives*), diterapkan metode *Lowe's Ratio Test*. Metode ini membandingkan jarak fitur terdekat dengan jarak fitur terdekat kedua; jika rasionya berada di bawah ambang batas tertentu (biasanya 0.75), maka pasangan fitur tersebut dianggap valid dan unik.

### C. Estimasi Homografi dan RANSAC

Dalam konteks penyambungan citra (image stitching), diperlukan estimasi transformasi geometri yang memetakan titik-titik pada satu citra ke citra lainnya. Hubungan ini dimodelkan oleh matriks Homografi (\$H\$). Karena data pencocokan fitur sering kali mengandung outliers (data yang tidak sesuai model), algoritma Random Sample Consensus (RANSAC) digunakan. RANSAC secara iteratif memilih subset acak dari pasangan fitur untuk mengestimasi model homografi terbaik yang memiliki jumlah inliers (data yang sesuai) terbanyak, sehingga menghasilkan penyatuan citra yang presisi dan minim distorsi.

## III. METODOLOGI

### A. Perangkat dan Akuisisi Data

Penelitian ini diimplementasikan menggunakan bahasa pemrograman Python 3.x dengan dukungan pustaka *Open Source Computer Vision Library* (OpenCV) untuk pemrosesan citra digital. Data yang digunakan merupakan citra primer yang diakuisisi menggunakan kamera *smartphone* di lingkungan pedesaan dengan kondisi pencahayaan alami (*outdoor*). Objek pengamatan meliputi vegetasi dan struktur bangunan semi-permanen yang memiliki variasi tekstur tinggi. Pengambilan citra dilakukan dengan teknik rotasi pada poros sumbu kamera (*panning*) untuk memastikan terdapat area tumpang tindih (*overlap*) minimal 40% antar *frame* sebagai prasyarat penyambungan citra.

### B. Tahapan Implementasi Sistem

Alur kerja sistem dalam membentuk panorama terdiri dari empat tahapan krusial:

1. **Pra-pemrosesan Citra (Preprocessing):** Sebelum masuk ke tahap deteksi, resolusi citra input direduksi (*downsampling*) sebesar 50% untuk mengoptimalkan efisiensi komputasi. Selanjutnya, dilakukan konversi ruang warna dari RGB ke Grayscale. Langkah ini esensial karena algoritma SIFT beroperasi berdasarkan intensitas luminans piksel, sehingga informasi krominans (warna) dapat diabaikan untuk mempercepat proses.
2. **Ekstraksi Fitur Berbasis SIFT:** Deteksi titik fitur (*interest points*) dilakukan menggunakan algoritma SIFT. Pada tahap ini, citra diproses melalui *Gaussian Scale-space* untuk mendapatkan *keypoints* yang invarian terhadap skala. Setiap *keypoint* kemudian diekstraksi menjadi vektor deskriptor 128-dimensi yang merepresentasikan orientasi gradien lokal, memberikan karakteristik unik pada setiap fitur.
3. **Pencocokan Fitur dan Filtrasi:** Proses pencocokan deskriptor antar dua citra dilakukan menggunakan

metode *Brute-Force Matcher* dengan algoritma *k-Nearest Neighbors* ( $k=2$ ). Untuk memvalidasi korespondensi fitur dan mengeliminasi pasangan yang ambigu, diterapkan *Lowe's Ratio Test* dengan ambang batas (*threshold*) rasio 0.75. Hanya pasangan fitur dengan distorsi jarak Euclidean minimal yang dipertahankan sebagai *Good Matches*.

4. **Estimasi Homografi dan *Warping*:** Matriks Homografi (\$H\$) diestimasi berdasarkan titik-titik korespondensi yang valid. Untuk menangani keberadaan *outliers* (data pencocokan yang salah), digunakan algoritma RANSAC (*Random Sample Consensus*) dengan toleransi reproyeksi 5.0 piksel. Citra target kemudian ditransformasikan (*warp perspective*) menggunakan matriks \$H\$ tersebut dan disatukan dengan citra referensi pada kanvas yang telah diperluas dimensinya untuk mengakomodasi perubahan geometri akibat rotasi perspektif.

## IV. HASIL DAN PEMBAHASAN

Eksperimen dilakukan menggunakan dataset citra lingkungan desa yang merepresentasikan kondisi dunia nyata dengan variasi tekstur alami yang kompleks, seperti vegetasi dan struktur bangunan.



Gambar 1. Visualisasi Tahapan SIFT: (a) Deteksi Keypoints pada Citra Kiri; (b) Deteksi Keypoints pada Citra Kanan; (c) Visualisasi Feature Matching yang valid (631 matches); (d) Hasil Akhir Panorama dengan koreksi perspektif.

### A. Analisis Deteksi Keypoints

Tahap awal melibatkan ekstraksi fitur lokal menggunakan SIFT. Seperti diilustrasikan pada **Gambar 1(a)** dan **1(b)**, algoritma berhasil memetakan ribuan titik fitur (*keypoints*) pada kedua citra uji. Distribusi keypoints terlihat sangat padat pada area dengan frekuensi spasial tinggi, seperti dedaunan pohon pisang dan pola genteng rumah. Sebaliknya, area dengan tekstur homogen seperti langit dan permukaan tanah yang rata memiliki densitas keypoints yang sangat rendah. Hal ini mengonfirmasi bahwa SIFT bekerja efektif dalam mengisolasi fitur yang distinctive (khas) dan mengabaikan informasi redundan yang tidak berkontribusi

pada proses pencocokan.

### B. Evaluasi Pencocokan Fitur (Feature Matching)

Proses pencocokan deskriptor antar citra divisualisasikan pada **Gambar 1(c)**. Penerapan *Lowe's Ratio Test* dengan ambang batas 0.75 terbukti krusial dalam menyaring pasangan fitur. Dari ribuan kandidat, diperoleh **631 pasangan fitur valid** (*inliers*) yang memiliki tingkat kemiripan tinggi. Visualisasi garis korespondensi menunjukkan pola paralel yang konsisten, menghubungkan objek yang sama (misalnya: batang pohon di sisi kiri terhubung presisi ke batang pohon di sisi kanan). Minimnya garis persilangan (*outliers*) mengindikasikan bahwa deskriptor SIFT memiliki diskriminabilitas tinggi meskipun terdapat sedikit perbedaan pencahayaan akibat bayangan awan.

### C. Rekonstruksi Panorama

Hasil akhir penyambungan citra ditampilkan pada **Gambar 1(d)**. Citra hasil jahitan (*stitched image*) memperlihatkan transisi yang mulus (*seamless*) pada area tumpang tindih, khususnya pada sambungan pagar kayu dan bayangan di tanah. Transformasi geometri terlihat jelas pada sisi kiri citra yang mengalami distorsi perspektif (*warping*). Distorsi ini merupakan konsekuensi matematis dari matriks Homografi yang diestimasi oleh RANSAC untuk menyelaraskan bidang pandang citra kiri agar sejajar dengan citra kanan. Keberhasilan rekonstruksi ini memvalidasi bahwa model geometri yang dihasilkan akurat dan mampu menangani rotasi kamera saat pengambilan gambar.

### V. Refleksi Pribadi

Sebelum mengerjakan proyek akhir ini, pemahaman saya tentang algoritma SIFT (*Scale-Invariant Feature Transform*) hanya terbatas pada teori di slide kuliah: bahwa SIFT bisa mengenali objek meskipun diperbesar atau diputar. Namun, setelah mengimplementasikannya secara langsung dalam kode Python, saya mendapatkan pemahaman yang jauh lebih konkret.

Saya menyadari bahwa "nyawa" dari SIFT terletak pada deskriptornya. Ketika saya melihat visualisasi *Keypoints* pada citra desa, saya takjub melihat bagaimana algoritma ini mengabaikan area langit yang polos dan fokus pada "pojokan" rumit seperti ujung daun pisang atau genteng. Ini menyadarkan saya bahwa bagi komputer, sebuah gambar hanyalah sekumpulan matriks angka, dan SIFT bertugas mengubah matriks tersebut menjadi "sidik jari" unik (vektor 128-dimensi) yang tahan banting terhadap perubahan geometri.

Kendala dan Solusi Tantangan terbesar yang saya hadapi bukanlah pada penulisan kode SIFT itu sendiri (karena pustaka OpenCV sangat membantu), melainkan pada tahap logika geometri saat penyatuan citra (*stitching*).

- **Kendala:** Awalnya, hasil panorama saya terpotong. Gambar sebelah kiri hilang sebagian setelah proses *warping*.
- **Analisis:** Setelah saya telusuri (*debugging*), ternyata transformasi matriks Homografi bisa menghasilkan koordinat negatif (keluar dari area gambar asli).

- **Solusi:** Saya belajar bahwa kita tidak bisa langsung menempel gambar. Saya harus menghitung ukuran "kanvas kosong" baru yang cukup besar untuk menampung gabungan kedua gambar, lalu menggunakan matriks translasi (pergeseran) untuk memindahkan gambar ke tengah kanvas agar tidak ada bagian yang terpotong. Ini mengajarkan saya bahwa Visi Komputer sangat erat kaitannya dengan Aljabar Linear.

Sebelum proyek ini, saya mengira pembuatan panorama itu sederhana: tinggal potong dan tempel. Sekarang saya paham bahwa di balik fitur "Panorama" di kamera HP kita, terdapat proses matematis yang sangat kompleks: mulai dari mencari ribuan titik fitur, membuang data sampah (*outliers*) dengan RANSAC, hingga melengkungkan perspektif gambar (*warping*) agar sejajar. Proyek ini mengubah cara pandang saya dari sekadar "pengguna teknologi" menjadi "pengembang" yang menghargai kerumitan di balik sebuah fitur sederhana.

Berdasarkan eksperimen saya di lingkungan *outdoor* (desa), kelebihan utama SIFT adalah **stabilitasnya**. Meskipun saya mengambil foto dengan tangan yang mungkin sedikit bergetar dan pencahayaan matahari yang tidak rata, SIFT tetap mampu menemukan lebih dari 600 titik kecocokan valid.

Namun, SIFT memiliki keterbatasan dalam hal **kecepatan**. Proses deteksi fitur memakan waktu beberapa detik pada laptop saya. Jika diterapkan untuk video *real-time*, mungkin SIFT akan terasa berat. Selain itu, saya menduga SIFT akan kesulitan jika dihadapkan pada objek dengan pola berulang yang ekstrem (seperti pagar besi yang seragam), karena deskriptornya akan menjadi ambigu.

Secara keseluruhan, proyek ini sangat memuaskan karena berhasil membuktikan teori yang dipelajari di kelas ke dalam aplikasi dunia nyata.

### VI. Kesimpulan

Implementasi algoritma *Scale-Invariant Feature Transform* (SIFT) untuk aplikasi penyambungan citra panorama pada lingkungan pedesaan telah berhasil dilaksanakan. Berdasarkan hasil eksperimen dan analisis yang dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. **Ketangguhan SIFT:** Algoritma SIFT terbukti sangat andal dalam mendeteksi fitur pada objek dengan tekstur alami yang kompleks (seperti vegetasi dan atap bangunan). Deteksi fitur konsisten meskipun terdapat variasi skala dan sedikit perubahan sudut pandang akibat pengambilan gambar secara *handheld* (tanpa tripod).
2. **Akurasi Pencocokan:** Penerapan *Lowe's Ratio Test* (dengan ambang batas 0.75) dikombinasikan dengan RANSAC sangat efektif dalam mengeliminasi *outliers*. Hal ini menghasilkan matriks Homografi yang presisi, sehingga garis sambungan pada panorama terlihat menyatu tanpa pergeseran yang signifikan.
3. **Koreksi Geometri:** Proses *warping* berhasil mengoreksi distorsi perspektif pada citra kiri, memungkinkan penggabungan yang mulus dengan

citra kanan pada kanvas yang diperluas.

Secara keseluruhan, penelitian ini memvalidasi bahwa meskipun SIFT memiliki beban komputasi yang relatif tinggi, stabilitas dan akurasi menjadi metode yang superior untuk rekonstruksi panorama otomatis dibandingkan metode deteksi sudut sederhana.

#### VII. Pernyataan Penggunaan Ai

Saya menyatakan bahwa penggunaan Artificial Intelligence (AI) dalam penyusunan laporan dan pengembangan kode pada tugas ini hanya digunakan sebagai alat bantu, seperti untuk memahami sintaks, debugging, atau referensi umum.

Seluruh analisis, pemahaman konsep, interpretasi hasil, dan penulisan penjelasan pribadi merupakan hasil pemikiran dan pekerjaan saya sendiri. Saya bertanggung jawab penuh atas isi laporan ini.

#### DAFTAR PUSTAKA

- [1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.  
+1
- [2] M. Brown and D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [3] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [4] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Springer, 2022.  
+1
- [5] OpenCV Documentation, "Feature Matching + Homography to find Objects," [Online]. Available: <https://docs.opencv.org/>. Accessed: Feb. 14, 2026.

