Implementasi Algoritma Scale-Invariant Feature Transform (SIFT) untuk Image Stitching Panorama

Daffa Rafly Muhammad
Teknik Informatika
Universitas Darussalam Gontor
Ponorogo, Indonesia
daffaraflymuhammad21@student.unida,gontor.ac.id

***Abstract*—Proyek akhir ini bertujuan untuk mengimplementasikan algoritma Scale-Invariant Feature Transform (SIFT) dalam aplikasi penyambungan citra (image stitching) untuk membentuk panorama. SIFT dipilih karena ketangguhannya terhadap perubahan skala, rotasi, dan pencahayaan. Eksperimen dilakukan menggunakan bahasa pemrograman Python dan pustaka OpenCV. Tahapan meliputi deteksi keypoint, ekstraksi deskriptor, pencocokan fitur (feature matching), dan estimasi homografi menggunakan RANSAC. Hasil pengujian menunjukkan bahwa metode ini mampu menyambungkan dua citra terpisah dengan sudut pandang berbeda menjadi satu kesatuan panorama yang presisi, meskipun terdapat tantangan dalam biaya komputasi.**

***Index Terms*—SIFT, Image Stitching, Panorama, Computer Vision, Feature Matching.**

I. Pendahuluan

Visi komputer adalah bidang yang memungkinkan komputer untuk memproses dan menganalisis citra digital layaknya penglihatan manusia. Salah satu tantangan utama dalam visi komputer adalah mengenali fitur objek yang sama pada citra yang berbeda, terlepas dari perubahan sudut pandang atau skala.

Dalam proyek ini, saya memilih topik *Image Stitching* atau Panorama. Pembuatan panorama memerlukan deteksi titik-titik penting (*keypoints*) yang akurat antara dua gambar yang saling tumpang tindih (*overlap*). Algoritma *Scale-Invariant Feature Transform* (SIFT) yang diperkenalkan oleh David Lowe digunakan sebagai metode utama karena kemampuannya menghasilkan fitur yang invarian (kokoh) terhadap transformasi geometri. Laporan ini akan membahas metodologi, hasil implementasi, serta analisis kinerja SIFT dalam membentuk panorama dari dataset citra lingkungan rumah (dapur).

II. Dasar Teori

A. Scale-Invariant Feature Transform (SIFT)

SIFT adalah algoritma untuk mendeteksi dan mendeskripsikan fitur lokal pada citra. Tahapan utamanya meliputi:

- **Scale-space Extrema Detection:** Menggunakan *Difference of Gaussian* (DoG) untuk mencari potensi keypoint yang invarian terhadap skala.
- **Keypoint Localization:** Menentukan lokasi akurat keypoint dan menghilangkan titik yang kontrasnya rendah atau berada di tepi (edge).
- **Orientation Assignment:** Memberikan orientasi pada keypoint berdasarkan gradien lokal agar invarian terhadap rotasi.
- **Keypoint Descriptor:** Membentuk vektor deskriptor (biasanya 128 dimensi) untuk setiap keypoint sebagai identitas uniknya.

B. Feature Matching dan Homografi

Setelah deskriptor didapat, dilakukan pencocokan fitur antar dua citra. *Brute-Force Matcher* atau FLANN digunakan untuk mencari pasangan fitur terdekat. Untuk menyaring pasangan yang salah (*outliers*), metode *Lowe's Ratio Test* diterapkan.

Selanjutnya, matriks Homografi dihitung menggunakan algoritma RANSAC (*Random Sample Consensus*) untuk memetakan perspektif satu citra ke citra lainnya agar dapat digabungkan menjadi panorama.

III. Metodologi

A. Perangkat dan Pustaka

Implementasi dilakukan menggunakan bahasa pemrograman Python. Pustaka utama yang digunakan adalah OpenCV untuk pemrosesan citra, NumPy untuk operasi matriks, dan Matplotlib untuk visualisasi hasil.

B. Dataset

Dataset yang digunakan adalah citra pribadi yang diambil menggunakan kamera ponsel. Objek yang diambil adalah area dapur dengan pencahayaan lampu ruangan. Dua gambar diambil dengan sudut pandang bergeser namun tetap memiliki area *overlap* sekitar 30-50% untuk memungkinkan pencocokan fitur.

C. Alur Implementasi

1) **Pra-pemrosesan:** Citra dimuat dan dikonversi ke *grayscale*.
2) **Deteksi SIFT:** Objek `cv2.SIFT_create()` digunakan untuk mendeteksi *keypoints* dan menghitung deskriptor pada kedua citra.
3) **Pencocokan:** Fitur dicocokkan menggunakan `BFMatcher` dengan norma L2.
4) **Penyaringan:** Penerapan *ratio test* (threshold 0.75) untuk mengambil *good matches*.

5) **Stitching:** Menghitung homografi dari *good matches*, melakukan *warping* citra, dan menggabungkannya dalam kanvas yang diperluas.

IV. Hasil dan Analisis

A. Visualisasi Keypoints

Langkah pertama adalah mendeteksi keypoint. Gambar 1 (kiri) menunjukkan visualisasi garis-garis pencocokan fitur (*matches*) yang menghubungkan titik-titik identik pada citra kiri dan kanan. SIFT berhasil mengidentifikasi pola keramik dinding dan sudut ruangan sebagai fitur yang kuat.

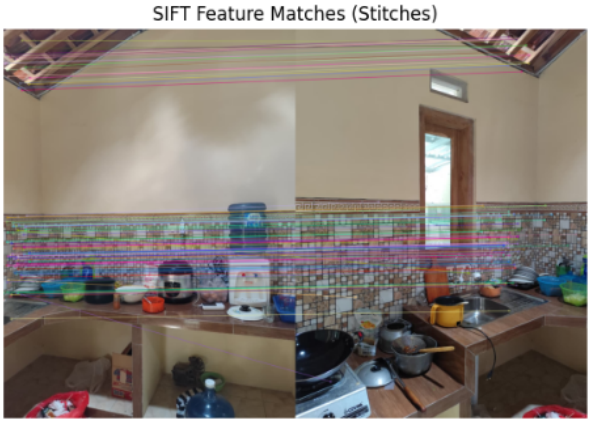

Fig. 1. Visualisasi Feature Matching antar dua citra dapur.

B. Hasil Panorama

Setelah homografi diterapkan, citra kanan ditransformasikan perspektifnya mengikuti citra kiri. Hasil penggabungan dapat dilihat pada Gambar 2.

Fig. 2. Hasil akhir Panorama (Image Stitching).

Terlihat adanya area hitam di sekeliling citra hasil (*expanded canvas*). Hal ini terjadi karena proses transformasi perspektif (*warping*) mengubah bentuk geometris gambar asli. Secara visual, penyambungan pada area dinding dan meja dapur terlihat mulus, menunjukkan akurasi matriks homografi yang dihasilkan oleh SIFT dan RANSAC.

C. Analisis Kegagalan dan Keberhasilan

Metode ini berhasil karena tekstur keramik dinding memberikan banyak fitur unik. Namun, potensi kegagalan dapat terjadi pada area dinding yang polos (tanpa tekstur) atau jika perubahan sudut pengambilan gambar terlalu ekstrem, yang menyebabkan fitur tidak lagi dikenali.

V. Penjelasan Pribadi Mahasiswa

Berikut ini adalah penjelasan dari pengalaman saya selama mempelajari dan mengerjakan matakuliah dan project ini.

A. Pemahaman Konsep SIFT

Selama mengerjakan proyek ini, saya memahami bahwa SIFT bukan cuma mencari sudut, tapi mencari titik yang stabil di berbagai skala dan angle kamera yang harus tepat. Konsep *Scale-space* sangat krusial agar komputer bisa mengenali objek yang sama meski kita memotretnya dari jarak jauh atau dekat.

B. Kendala dan Solusi

Kendala utama yang saya hadapi adalah saat instalasi pustaka OpenCV versi lama yang mendukung SIFT (sebelum patennya kadaluarsa), namun ternyata di versi OpenCV terbaru SIFT sudah terintegrasi secara gratis. Solusinya adalah memastikan versi ‘opencv-contrib-python‘ sudah terbarui. Selain itu, menyesuaikan foto/dataset agar gambar tidak terpotong juga cukup sulit. Dan juga salah satu kendala saya adalah belum terlalu paham memakai aplikasi PyCharm. Solusinya adalah bertanya dengan teman sebaya, tutorial, dan juga AI.

C. Perubahan Pemahaman

Sebelum proyek ini, saya mengira penyambungan gambar hanya sekadar menempelkan dua foto. Sekarang saya mengerti bahwa ada proses matematis kompleks (Homografi) untuk meluruskan perspektif agar sambungannya terlihat alami.

D. Kelebihan dan Keterbatasan SIFT

Menurut saya, kelebihan SIFT adalah akurasinya yang sangat tinggi untuk pencocokan detail. Namun, keterbatasannya adalah prosesnya relatif lambat dibandingkan algoritma lain seperti ORB, sehingga mungkin kurang cocok untuk aplikasi *real-time* di perangkat mobile berspesifikasi rendah.

VI. Kesimpulan

Implementasi SIFT untuk pembuatan panorama telah berhasil dilakukan. Algoritma ini terbukti efektif dalam mengenali fitur pada citra ruangan dan menyambungkannya dengan presisi. Meskipun membutuhkan sumber daya komputasi yang lebih besar, hasil visual yang diberikan lebih stabil dibandingkan metode deteksi sudut sederhana. Pengembangan selanjutnya dapat dilakukan dengan menerapkan metode *blending* (seperti multi-band blending) untuk menghilangkan garis sambungan yang mungkin masih terlihat halus.

REFERENCES

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] OpenCV Documentation, "Feature Matching + Homography to find Objects," [Online]. Available: https://docs.opencv.org/.

[3] Analytics Vidhya, "OpenCV: Feature Matching + SIFT Algorithm (Scale-Invariant Feature Transform)," Medium, [Online]. Available: https://medium.com/analytics-vidhya/opencv-feature-matching-siftalgorithm-scale-invariant-feature-transform-16672eafb253

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.