

# Perbandingan Metode SIFT dan ORB untuk Pencocokan Fitur Citra

Note: This space intentionally left blank to comply with IEEE formatting.

Wafa Syaefurokhman  
Informatics Engineering Department  
Universitas Darussalam Gontor  
Ponorogo, Indonesia  
[wafabila@student.cs.unida.gontor.ac.id](mailto:wafabila@student.cs.unida.gontor.ac.id)

*Abstract—Penelitian ini bertujuan untuk membandingkan performa algoritma Scale-Invariant Feature Transform (SIFT) dan Oriented FAST and Rotated BRIEF (ORB) dalam tugas deteksi dan pencocokan fitur pada citra. Evaluasi dilakukan berdasarkan jumlah keypoints, jumlah good matches, inlier ratio menggunakan RANSAC, serta waktu eksekusi. Hasil eksperimen menunjukkan bahwa SIFT menghasilkan pencocokan yang lebih stabil dan inlier ratio lebih tinggi, sedangkan ORB memiliki keunggulan dalam efisiensi waktu komputasi. Penelitian ini menunjukkan adanya trade-off antara ketahanan terhadap transformasi geometris dan kecepatan pemrosesan.*

**Keywords**— SIFT, ORB, Feature Matching, RANSAC, Computer Vision.

## I. PENDAHULUAN

Perkembangan teknologi computer vision memungkinkan komputer untuk memahami dan menganalisis citra digital. Salah satu permasalahan penting dalam computer vision adalah bagaimana sistem dapat mengenali objek yang sama meskipun mengalami perubahan skala, rotasi, pencahayaan, maupun sudut pandang.

Metode Scale-Invariant Feature Transform (SIFT) merupakan salah satu algoritma populer yang dirancang untuk mendekripsi dan mendeskripsikan fitur lokal pada citra yang bersifat invariant terhadap skala dan rotasi. Selain SIFT, terdapat metode lain seperti ORB (Oriented FAST and Rotated BRIEF) yang dikembangkan untuk memberikan performa lebih cepat dengan kompleksitas komputasi lebih rendah.

Topik ini dipilih karena pencocokan fitur merupakan dasar dari banyak aplikasi seperti object recognition, image stitching, augmented reality, dan visual SLAM. Dengan membandingkan SIFT dan ORB, dapat diketahui kelebihan dan kekurangan masing-masing metode dalam hal akurasi dan efisiensi.

## II. DASAR TEORI

### A. Keypoint.

Keypoint adalah titik-titik penting pada citra yang memiliki karakteristik unik dan stabil terhadap perubahan tertentu. Keypoint biasanya berada pada sudut (corner),

tepi kuat, atau area dengan perubahan intensitas signifikan.

### B. Descriptor

Descriptor adalah representasi numerik dari area di sekitar keypoint. Descriptor digunakan untuk mencocokkan fitur antara dua citra.

### C. SIFT (Scale-Invariant Feature Transform)

SIFT bekerja melalui beberapa tahap:

#### 1. Scale-space extrema detection

Menggunakan Difference of Gaussian (DoG) untuk mendekripsi kandidat keypoint pada berbagai skala.

#### 2. Keypoint localization

Menghilangkan keypoint yang tidak stabil atau memiliki kontras rendah.

#### 3. Orientation assignment

Memberikan orientasi pada keypoint berdasarkan arah gradien dominan.

#### 4. Keypoint descriptor

Membuat vektor descriptor 128-dimensi berdasarkan distribusi gradien.

Keunggulan utama SIFT adalah invariansi terhadap:

- Skala
- Rotasi
- Perubahan pencahayaan moderat

### D. ORB

ORB merupakan kombinasi FAST detector dan BRIEF descriptor yang ditambahkan orientasi. ORB lebih cepat dibanding SIFT karena menggunakan descriptor biner, sehingga cocok untuk aplikasi real-time.

### III. METODOLOGI

#### A. Alur Sistem

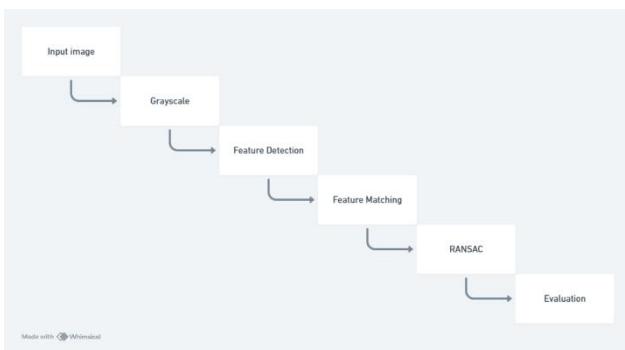


Fig. 1. Diagram alur sistem perbandingan SIFT dan ORB.

Alur sistem pada penelitian ini dimulai dengan memasukkan dua citra sebagai input, yang kemudian dikonversi ke dalam format grayscale untuk menyederhanakan proses komputasi karena algoritma SIFT dan ORB bekerja berdasarkan intensitas piksel. Selanjutnya dilakukan proses deteksi keypoint dan ekstraksi descriptor menggunakan kedua metode tersebut. Descriptor yang dihasilkan kemudian dicocokkan menggunakan Brute Force Matcher (BFMatcher), dan hasil pencocokan disaring menggunakan ratio test untuk memperoleh pasangan fitur terbaik. Pasangan fitur yang lolos seleksi digunakan untuk menghitung homografi dengan algoritma RANSAC [1] guna memisahkan inlier dan outlier berdasarkan konsistensi geometris. Setelah itu, dihitung nilai inlier ratio dan waktu komputasi sebagai parameter evaluasi performa masing-masing metode, kemudian seluruh hasil dianalisis secara kuantitatif dan visual untuk membandingkan tingkat akurasi serta efisiensi SIFT [2] dan ORB [3].

#### B. Dataset

Dataset terdiri dari dua citra objek yang sama dengan perbedaan sudut pandang dan pencahayaan.



Fig. 2. Citra pertama (img1).



Fig. 3. Citra kedua (img2).

#### C. Implementasi

Implementasi dilakukan menggunakan:

- Python 3.10
- OpenCV (opencv-contrib-python)
- NumPy
- Pandas

Eksperimen dijalankan pada lingkungan virtual (.venv) untuk menjaga konsistensi dependensi.

### IV. HASIL DAN ANALISIS

#### A. Visualisasi Keypoints



Fig. 4. Visualisasi keypoints menggunakan metode SIFT.



Fig. 5. Visualisasi keypoints menggunakan metode ORB.

Pada Gambar 4 terlihat bahwa SIFT menghasilkan jumlah keypoint yang lebih banyak dan tersebar merata pada area objek. Sedangkan pada Gambar 5, ORB menghasilkan keypoint yang lebih terbatas dan terkonsentrasi pada area kontras tinggi.

### B. Visualisasi Feature Matching



Fig. 6. Hasil feature matching menggunakan SIFT.

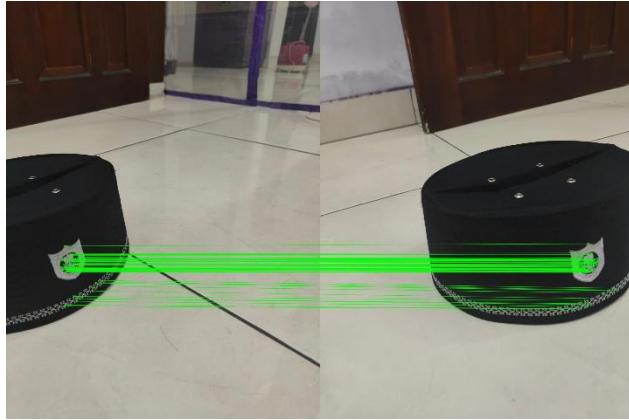


Fig. 7. Hasil feature matching menggunakan ORB.

Pada Gambar 6 terlihat bahwa SIFT menghasilkan garis pencocokan yang lebih konsisten dan terpusat pada area objek utama. Jumlah garis yang sesuai (inlier) lebih banyak dibanding ORB. Pada Gambar 7, ORB tetap menghasilkan pencocokan yang cukup baik, namun beberapa garis terlihat kurang stabil terutama pada area perubahan perspektif.

### C. Hasil Kuantitatif

Method	Keypoints Img1	Keypoints Img2	Good Matches	Inlier Ratio	Time (s)
SIFT	1031	1368	193	0.549	0.554
ORB	1000	1000	160	0.462	0.391

Table 1. hasil kuantitatif

### D. Analisis

SIFT menghasilkan jumlah keypoint dan good matches lebih banyak dibanding ORB. Nilai inlier ratio yang lebih tinggi menunjukkan bahwa kecocokan fitur SIFT lebih akurat dan stabil.

Namun dari sisi waktu komputasi, ORB lebih cepat karena menggunakan descriptor biner yang lebih ringan secara komputasi.

### E. Keberhasilan dan Kegagalan

#### Keberhasilan:

- SIFT berhasil mempertahankan akurasi pada perubahan skala kecil.
- ORB bekerja cepat dan tetap menghasilkan matching yang cukup baik.

#### Kegagalan:

- Pada area dengan tekstur rendah, kedua metode menghasilkan lebih sedikit keypoint.
- ORB lebih sensitif terhadap perubahan skala dibanding SIFT.

## V. PENJELASAN PRIBADI MAHASISWA

Sebelum mengerjakan project ini, saya memahami SIFT hanya sebagai algoritma deteksi fitur tanpa benar-benar mengetahui bagaimana prosesnya bekerja. Saya hanya mengetahui bahwa SIFT dapat mendeteksi titik-titik penting pada gambar.

Setelah mempelajari dan mengimplementasikan langsung, saya memahami bahwa SIFT memiliki tahapan kompleks seperti scale-space detection dan perhitungan orientasi gradien. Saya menjadi lebih paham mengapa SIFT disebut invariant terhadap skala dan rotasi.

Kendala yang saya hadapi adalah error saat instalasi OpenCV versi tertentu dan perbedaan hasil descriptor antara SIFT dan ORB. Selain itu, saya sempat kesulitan memahami konsep RANSAC untuk menghitung inlier ratio. Solusinya adalah membaca dokumentasi OpenCV dan mencoba beberapa eksperimen kecil untuk memahami alurnya.

Pemahaman saya berubah cukup signifikan. Awalnya saya menganggap semua metode feature detection hampir sama. Setelah eksperimen, saya menyadari bahwa setiap metode memiliki trade-off antara akurasi dan kecepatan.

Menurut saya, kelebihan SIFT adalah robust dan stabil terhadap transformasi geometris. Namun kekurangannya adalah waktu komputasi lebih lambat. ORB lebih ringan dan cocok untuk sistem real-time, tetapi kurang stabil terhadap perubahan skala besar.

Project ini membuat saya lebih memahami bagaimana algoritma computer vision [4] bekerja secara praktis, bukan hanya teori.

## VI. KESIMPULAN

Dari hasil eksperimen dapat disimpulkan bahwa:

1. SIFT memberikan hasil matching lebih akurat.
2. ORB memberikan performa lebih cepat.
3. Pemilihan metode harus disesuaikan dengan kebutuhan aplikasi.

Pengembangan selanjutnya dapat dilakukan dengan:

- Menambahkan metode SURF [5] sebagai pembanding
- Menguji pada dataset lebih besar
- Menguji pada variasi rotasi dan skala ekstrem

## VII. DAFTAR PUSTAKA

- [1] M. A. Fischler and R. C. Bolles, “Random sample consensus,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.
- [2] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, IEEE, Nov. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.
- [4] R. Szeliski, *Computer Vision*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-34372-9.
- [5] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” 2006, pp. 404–417. doi: 10.1007/11744023\_32.

## VIII. 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.