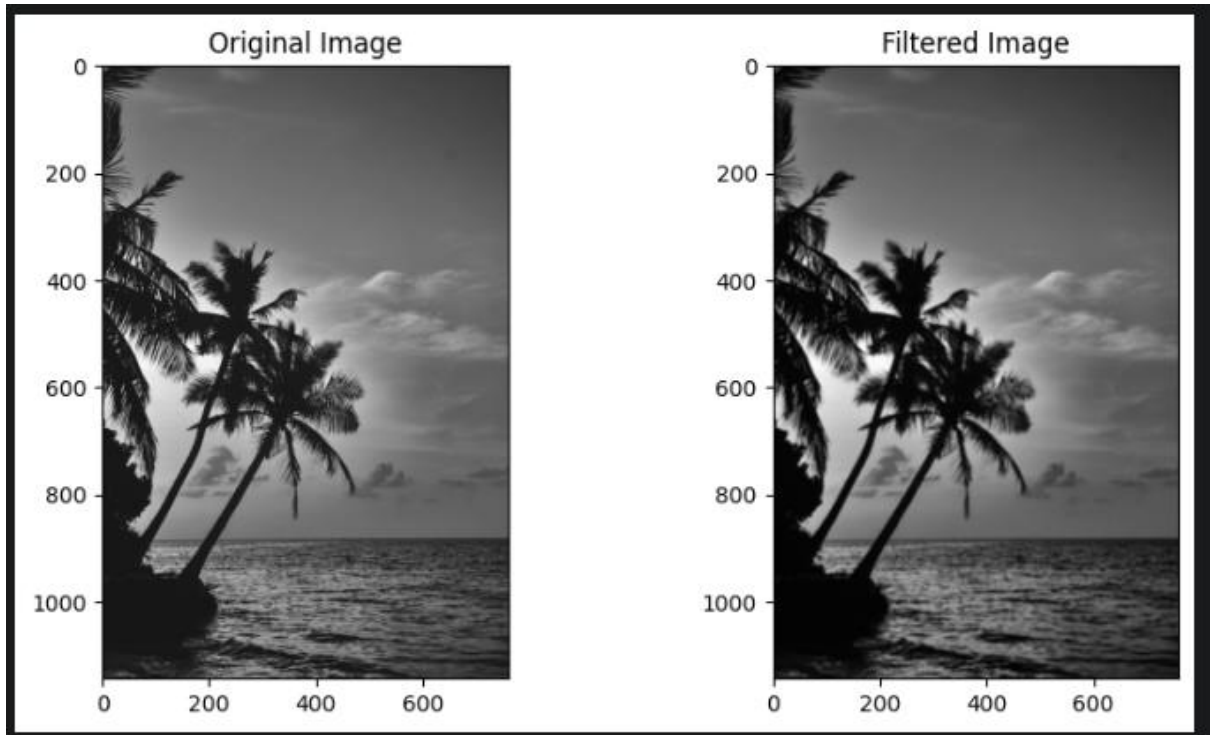


Ade Tirta Rahmat Hidayat – Analisis Simulasi

Link Google Colab:

https://colab.research.google.com/drive/1P41uFsODyPCZZ0f_qYV4iJ5qJYlu-282?usp=sharing

1. Pengaplikasian *Filter Moving Average*



Keterangan:

- Original Image:
Gambar pertama (kiri) adalah gambar asli dalam skala abu-abu. Semua detail gambar tetap utuh, termasuk detail tekstur, batasan objek, dan variasi intensitas.
- Filtered Image: Gambar kedua (kanan) menunjukkan hasil setelah diterapkannya filter moving average dengan kernel 5×5 . Filter ini bekerja dengan meratakan intensitas pixel di area lokal, menghasilkan efek seperti blur.

Efek dari *Filter Moving Average*:

- Pengurangan Noise:
Filter moving average berguna untuk mengurangi noise acak pada gambar, terutama noise dengan intensitas tinggi (salt-and-pepper noise). Namun, karena ini adalah filter sederhana berbasis rata-rata, ia tidak dapat mempertahankan detail atau batasan objek dengan baik.
- Efek Buram (Blur):
Detail halus, seperti tepi objek atau tekstur kecil, menjadi kurang tajam karena intensitas setiap piksel digantikan oleh rata-rata nilai intensitas di sekitarnya. Pada gambar tersebut, batas pohon dan tekstur awan menjadi kurang jelas dibandingkan gambar asli.
- Kernel Size:

Ukuran kernel (5×5) yang digunakan cukup besar, sehingga efek blurnya cukup nyata. Kernel yang lebih kecil (3×3) akan memberikan efek yang lebih halus, sedangkan kernel yang lebih besar (7×7 atau lebih) akan memberikan efek blur yang lebih kuat.

2. Deteksi Fitur dengan SIFT



SIFT

- SIFT adalah metode untuk mendeteksi dan mendeskripsikan fitur lokal (*keypoints*) pada gambar.
- Metode ini dirancang untuk *invariance* skala dan rotasi, artinya fitur yang dideteksi tetap konsisten meskipun terjadi perubahan ukuran atau rotasi pada gambar.

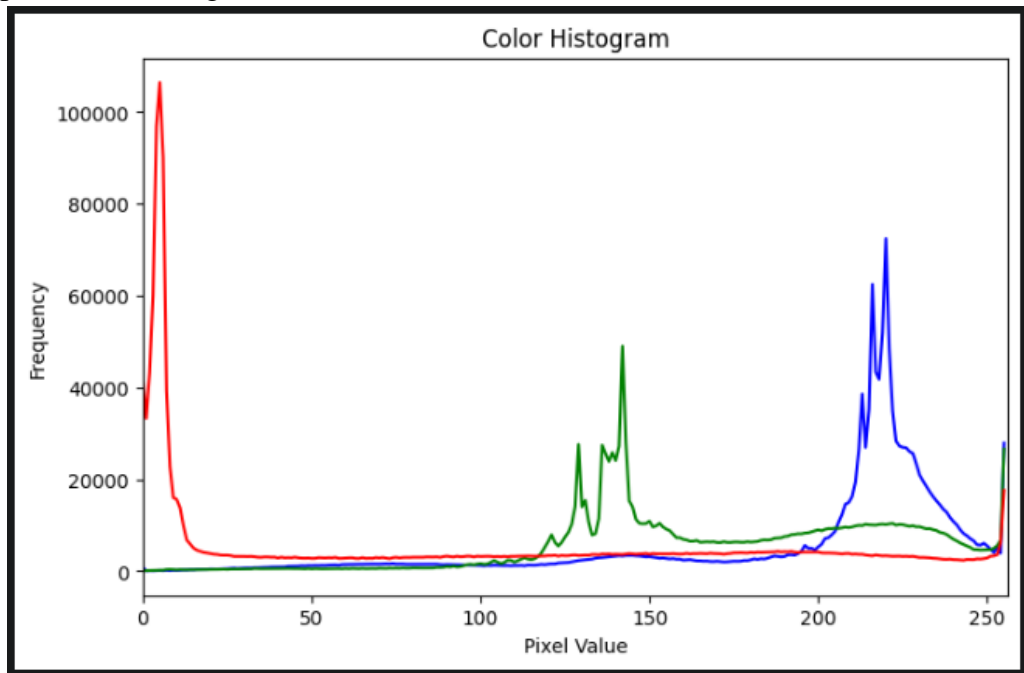
Analisis Keypoints

- Pada gambar, lingkaran-lingkaran berwarna menunjukkan lokasi keypoints yang telah terdeteksi oleh algoritma SIFT.
- Lingkaran dengan berbagai ukuran menunjukkan bahwa algoritma telah mendeteksi fitur di berbagai skala (dari fitur kecil hingga besar).
- Pusat lingkaran menandai posisi fitur, sementara ukuran lingkaran menggambarkan skala fitur tersebut.
- Arah panah kecil pada lingkaran (tidak terlalu terlihat jelas) menunjukkan orientasi dominan di sekitar fitur, yang digunakan untuk invariansi rotasi.

Pada gambar tersebut, sebagian besar keypoints terkonsentrasi di area dengan tekstur kompleks atau kontras tinggi, seperti:

- Ranting dan daun pohon kelapa.
- Ujung-ujung pohon yang berbatasan dengan langit.
- Area air laut yang memiliki variasi pola gelombang dan intensitas cahaya.
- Area yang datar atau memiliki perubahan intensitas yang kecil, seperti bagian langit yang halus, memiliki lebih sedikit keypoints.

3. Representasi Histogram Gambar



Output Histogram dengan keterangan:

- Sumbu X: Menunjukkan rentang nilai piksel (0-255) untuk setiap warna. Nilai 0 berarti intensitas warna minimum (hitam). Nilai 255 berarti intensitas warna maksimum (putih/warna penuh).
- Sumbu Y: Menunjukkan frekuensi kemunculan piksel untuk setiap nilai.
Kurva Warna: Kurva biru, hijau, dan merah menunjukkan distribusi intensitas masing-masing kanal warna.

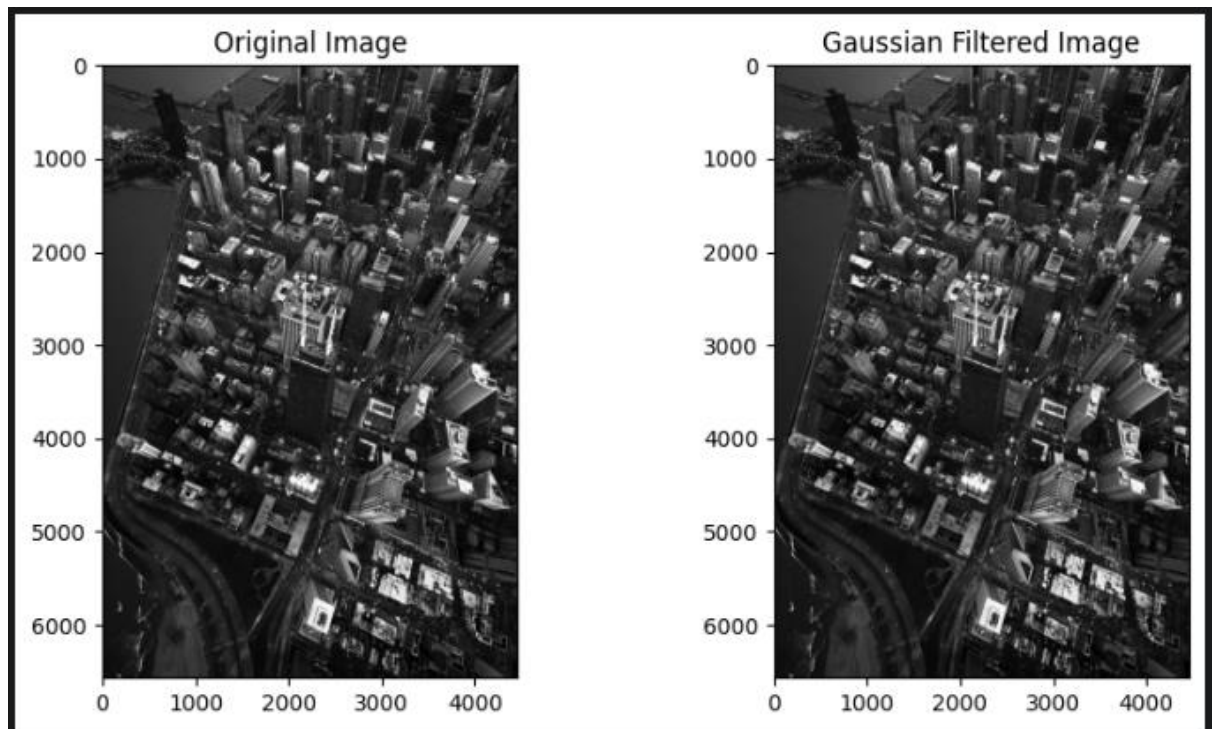
Dominasi Warna:

- Kurva warna biru (blue channel) terlihat memiliki lonjakan pada nilai piksel yang lebih tinggi (mendekati 200-250), menunjukkan bahwa ada area dominan dengan warna biru terang di gambar.
- Kanal hijau menunjukkan puncak di area nilai piksel menengah (sekitar 150), menunjukkan elemen hijau di gambar.
- Kanal merah memiliki distribusi yang lebih merata, meskipun dengan frekuensi yang lebih rendah dibandingkan dua kanal lainnya.

Fokus Warna:

- Lonjakan besar di kanal tertentu (misalnya biru) mengindikasikan adanya elemen dominan dari warna tersebut.
- Kanal merah yang lebih rendah frekuensinya dapat menunjukkan gambar lebih condong ke warna dingin (kebiruan atau kehijauan).

4. *Gaussian Smoothing*



Gaussian Blur

- Bertujuan untuk mengurangi noise, menghaluskan transisi, dan menyederhanakan detail tanpa menghilangkan informasi penting.
- Gaussian smoothing menggunakan kernel Gaussian untuk meratakan gambar.

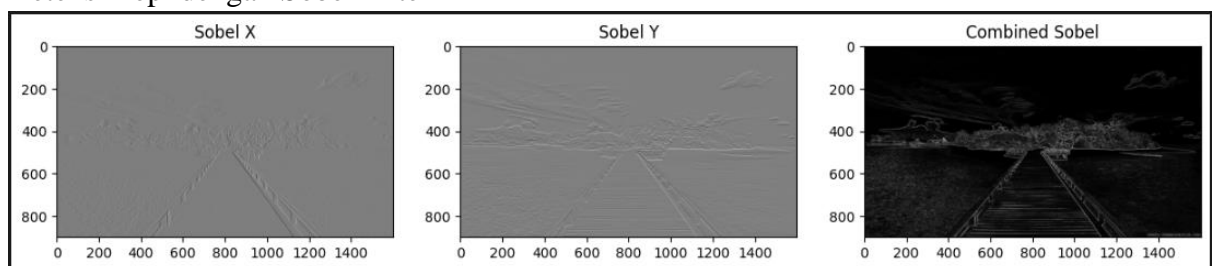
Parameter:

- o (15, 15): Ukuran kernel 15x15 (mengontrol area pengaruh smoothing).
- o $\sigma = 1$: Parameter deviasi standar Gaussian, mengontrol seberapa intens smoothing dilakukan

Analisis Gambar:

- Gambar Asli (Original Image):
 - o Menampilkan detail tajam dari gambar.
 - o Terdapat noise atau detail kecil yang terlihat jelas.
- Gambar Gaussian Filtered:
 - o Detail halus dan tajam di gambar asli diratakan.
 - o Noise berkurang, terutama pada area datar seperti langit atau bayangan.
 - o Beberapa detail tajam (seperti tepi bangunan) menjadi kabur akibat smoothing.

5. Deteksi Tepi dengan Sobel Filter



Sobel Filter

- Sobel filter adalah operator diferensial untuk mendeteksi tepi dalam gambar dengan menghitung perubahan intensitas piksel.
- Filter ini bekerja dengan dua kernel:
Sobel X: Mendeteksi perubahan horizontal (gradien pada arah x).
Sobel Y: Mendeteksi perubahan vertikal (gradien pada arah y).

Sobel X:

- Menunjukkan tepi yang sejajar secara vertikal, yaitu garis-garis yang mengalami perubahan intensitas dalam arah horizontal.
- Misalnya: Jembatan atau tepi objek yang memanjang dalam arah horizontal terlihat lebih jelas.

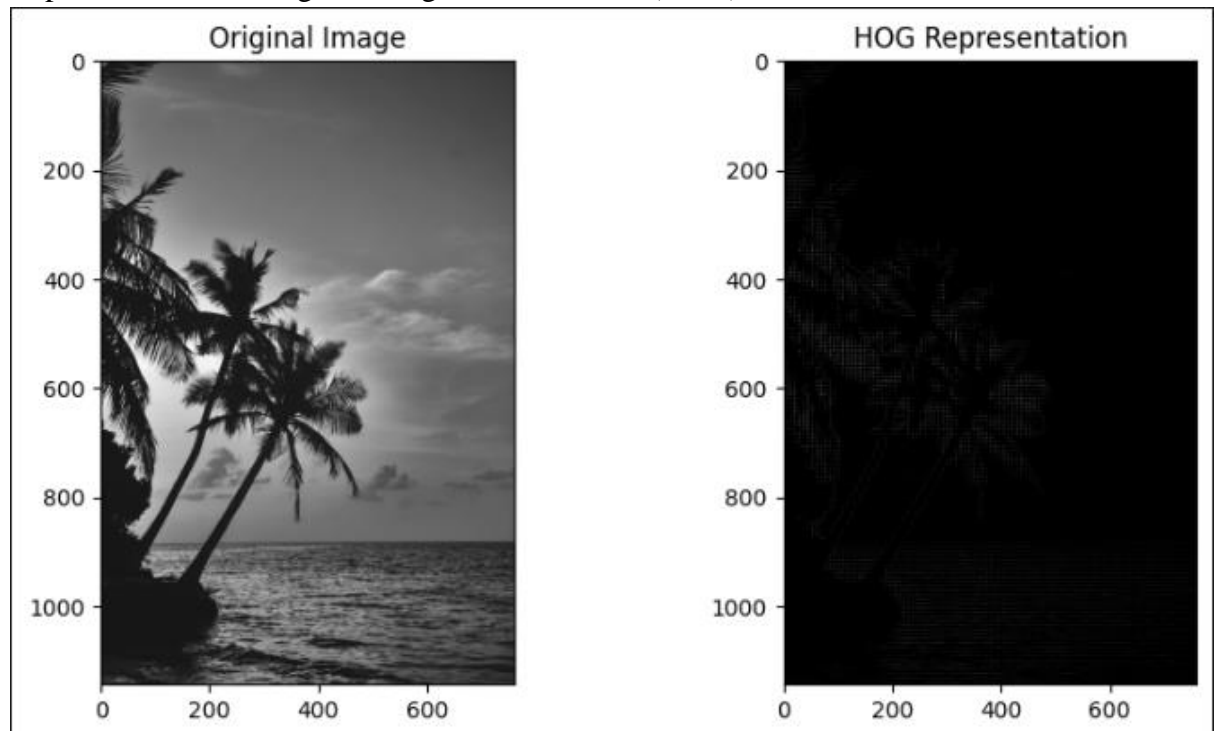
Sobel Y:

- Mendeteksi tepi yang sejajar secara horizontal, yaitu garis-garis yang mengalami perubahan intensitas dalam arah vertikal.
- Contoh: Garis vertikal seperti tepi tiang atau struktur vertikal lainnya lebih menonjol.

Combined Sobel:

- Kombinasi magnitudo gradien dari Sobel X dan Sobel Y memberikan deteksi tepi penuh.
- Area dengan perubahan intensitas yang signifikan dalam arah horizontal dan vertikal terlihat jelas.
- Hasil ini memberikan representasi keseluruhan tepi dalam gambar.

6. Representasi Fitur dengan Histogram of Oriented (HOG)



Histogram of Oriented Gradients (HOG)

- Pola intensitas menggambarkan gradien arah (orientasi) pada piksel-piksel gambar.

- Area yang memiliki tepi dan perubahan tekstur yang signifikan, seperti tepi pohon kelapa dan langit, lebih terlihat dalam representasi HOG.
- Bagian datar (seperti langit) kurang terwakili, karena gradien di area tersebut relatif kecil.

7. Visualisasi *Tracking* dengan OpenCv pada Webots

Visualisasi Tracking dalam konteks ini melibatkan penggunaan OpenCV untuk mendeteksi dan melacak objek dalam dunia virtual Webots. Menggunakan OpenCV untuk visualisasi tracking sangat efektif karena OpenCV memiliki berbagai algoritma deteksi yang dapat dioptimalkan untuk tracking objek dengan presisi tinggi. Dalam simulasi Webots, integrasi dengan OpenCV memungkinkan robot untuk berinteraksi dengan dunia secara dinamis, seperti mendeteksi objek atau menavigasi di sekitar hambatan.

8. *Document Scanner Simulation* pada Webots

Document Scanner Simulation dalam Webots adalah simulasi di mana robot dengan kamera diprogram untuk mendeteksi dan memindai dokumen secara otomatis. Biasanya, ini melibatkan penggunaan kamera robot untuk mendeteksi tepi dokumen dan kemudian memotongnya atau mengambil gambar dengan kualitas tinggi.

9. *Fruit Detection Robot* pada Webots

Fruit Detection Robot pada Webots adalah simulasi di mana robot dilengkapi dengan kamera untuk mendeteksi dan mengenali buah dalam dunia virtual. Biasanya, ini melibatkan color detection untuk mendeteksi berbagai jenis buah yang memiliki warna atau bentuk yang jelas.