

LAPORAN TEORI

PENGOLAHAN CITRA



INTELLIGENT COMPUTING

CH

NAMA : Satria Bima Anggara

NIM : 202331114

KELAS : B

DOSEN : Ir. Darma Rusjdi , M.Kom

NO.PC : 04

ASISTEN : 1. Davina Najwa Ermawan

2. Fakhrol Fauzi Nugraha Tarigan

3. Viana Salsabila Fairuz Syala

4. Muhammad Hanief Febriansyah

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2024/2025

1. Konsep dasar dari operasi konvolusi dalam pengolahan citra digital.
Operasi konvolusi adalah teknik matematis yang sangat penting dalam pengolahan citra digital. Konvolusi melibatkan penggabungan dua fungsi untuk menghasilkan fungsi ketiga, yang biasanya mencerminkan bagaimana satu fungsi mempengaruhi yang lainnya. Dalam konteks citra, fungsi pertama adalah citra itu sendiri, dan fungsi kedua adalah kernel (atau filter). Proses konvolusi dilakukan dengan cara menggeser kernel di atas citra, satu piksel pada satu waktu.
2. Apa yang dimaksud dengan metode ketetapan dalam pengolahan citra.
Metode ketetapan dalam pengolahan citra mencakup berbagai teknik yang dirancang untuk meningkatkan kualitas citra dan mengekstrak informasi yang relevan. Metode ini sangat penting dalam aplikasi seperti pengenalan wajah, analisis medis, dan sistem pemantauan. Beberapa metode ketetapan yang umum digunakan meliputi:
 - Pengurangan Noise: Menggunakan filter seperti Average atau Median untuk mengurangi noise yang dapat mengganggu analisis citra.
 - Peningkatan Kontras: Menggunakan histogram equalization untuk memperbaiki kontras citra, sehingga detail yang sebelumnya tersembunyi menjadi lebih terlihat.
3. Jelaskan prinsip kerja Average Filter dan median Filter
 - Average Filter: Filter rata-rata bekerja dengan menghitung rata-rata nilai piksel dalam area di sekitar piksel yang sedang diproses. Ini menghasilkan citra yang lebih halus, tetapi dapat mengurangi detail penting, khususnya tepi. Average filter paling efektif untuk menghilangkan noise acak, tetapi bisa mengaburkan fitur penting.
 - Median Filter: Filter median menggantikan nilai piksel dengan nilai median dari piksel-piksel di sekitarnya. Ini sangat efektif untuk menghilangkan speckle noise, karena median lebih tahan terhadap outlier dibandingkan rata-rata. Median filter membantu mempertahankan tepi dan detail penting dalam citra, menjadikannya pilihan yang lebih baik dalam banyak aplikasi.
4. Apa Itu Speckle noise dan bagaimana ia berbeda dari Gaussian noise?
Speckle Noise: Speckle noise adalah jenis noise yang umum ditemui dalam citra yang diambil menggunakan metode radar atau ultrasonik. Noise ini muncul sebagai titik-titik acak yang dapat mengganggu detail penting dalam citra. Speckle noise biasanya bersifat impulsif dan dapat merusak kualitas citra, membuatnya sulit untuk analisis lebih lanjut.

Gaussian Noise: Ini adalah noise yang dihasilkan dari fluktuasi acak dalam sinyal, biasanya mengikuti distribusi normal. Gaussian noise cenderung merata di seluruh citra dan lebih mudah dihapus dengan teknik pengolahan citra seperti filter Gaussian. Noise ini lebih umum dalam citra yang diambil dalam kondisi pencahayaan rendah atau dengan sensor yang memiliki sensitivitas tinggi.
5. Uraikan manfaat dari operasi konvolusi dalam bidang computer visi
Pengenalan Pola: Konvolusi memungkinkan sistem untuk mengenali pola dalam citra, seperti bentuk dan tekstur, yang penting dalam aplikasi seperti pengenalan wajah dan objek.
Deteksi Objek: Dengan menerapkan filter khusus, sistem dapat mendeteksi objek dalam citra dengan akurasi tinggi, yang berguna dalam pengawasan dan analisis citra medis.

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA



NAMA : Satria Bima Anggara

NIM : 202331114

KELAS : B

DOSEN : Ir. Darma Rusjdi , M.Kom

NO.PC : 04

ASISTEN : 1. Davina Najwa Ermawan

2. Fakhrol Fauzi Nugraha Tarigan

3. Viana Salsabila Fairuz Syala

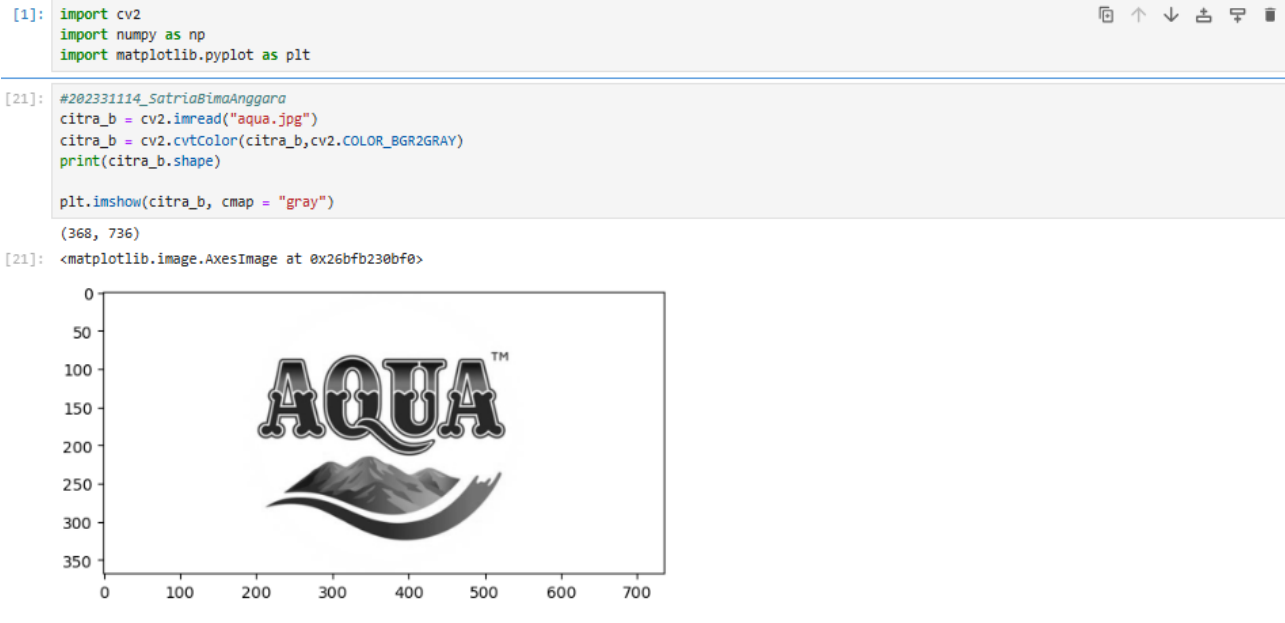
4. Muhammad Hanief Febriansyah

INSTITUT TEKNOLOGI PLN

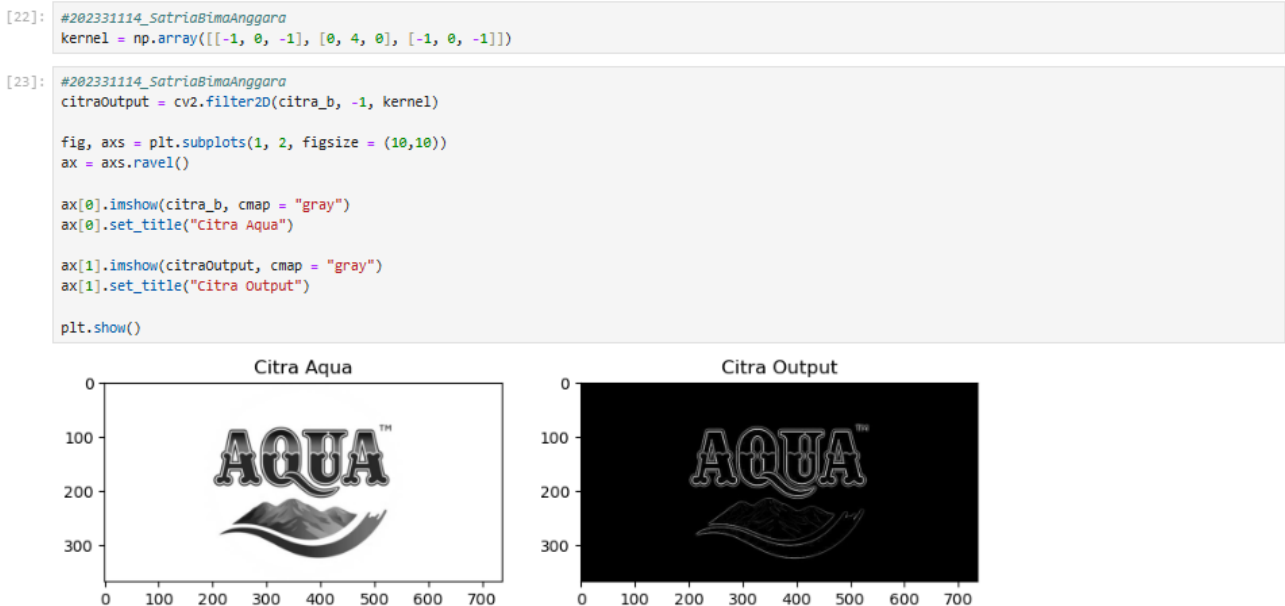
TEKNIK INFORMATIKA

2024/2025

Praktikum 3



cv2: Library OpenCV untuk pengolahan citra. `cv2.imread("aqua.jpg")`: Membaca file gambar boneka. `kernel = np.array([[-1, 0, -1], [0, 4, 0], [-1, 0, -1]])`, Kernel ini digunakan untuk proses konvolusi (filtering).

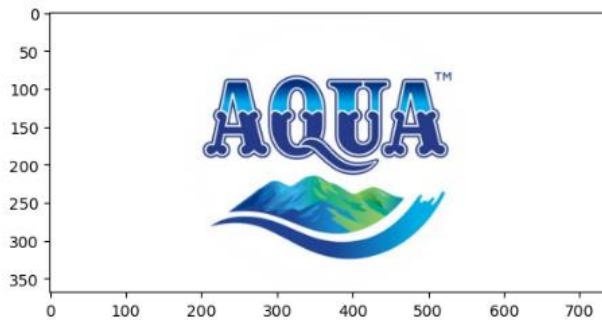


Kernel ini digunakan untuk proses konvolusi (filtering). `cv2.filter2D`: Fungsi untuk mengaplikasikan filter 2D (konvolusi). `citraOutput`: Hasil gambar setelah difilter. `plt.subplots`: Membuat tampilan 2 gambar berdampingan.

Praktikum 3

```
[24]: #202331114_SatriaBimaAnggara
citra_r = cv2.imread("aqua.jpg")
citra_r = cv2.cvtColor(citra_r, cv2.COLOR_BGR2RGB)

plt.imshow(citra_r)
plt.show()
```



Membaca gambar "aqua.jpg" dan menampilkannya dalam mode RGB. Kode Ini untuk menampilkan citra berwarna asli, bukan grayscale.

```
[6]: from skimage.util import random_noise

[25]: #202331114_SatriaBimaAnggara
noise_img_snp = random_noise(citra_r, mode = "s&p", amount = 0.2)

noise_img_gaussian = random_noise(citra_r, mode = "gaussian", mean = 0, var = 0.01)

noise_img_speckle = random_noise(citra_r, mode = "speckle")

noise_img_snp = np.array(255*noise_img_snp, dtype = 'uint8')
noise_img_gaussian = np.array(255*noise_img_gaussian, dtype = 'uint8')
noise_img_speckle = np.array(255*noise_img_speckle, dtype = 'uint8')

[26]: #202331114_SatriaBimaAnggara
fig, axs = plt.subplots(4, 1, figsize = (10, 40))
ax = axs.ravel()

ax[0].imshow(citra_r, cmap = "gray")
ax[0].set_title('Citra Asli')

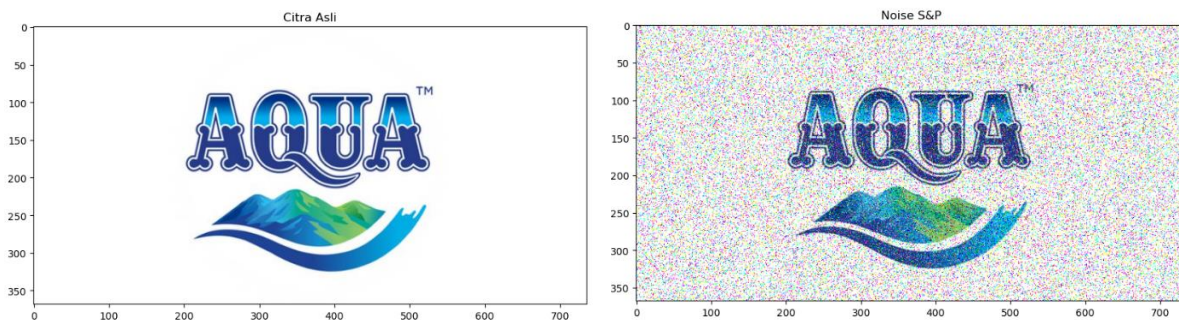
ax[1].imshow(noise_img_snp, cmap = "gray")
ax[1].set_title('Noise S&P')

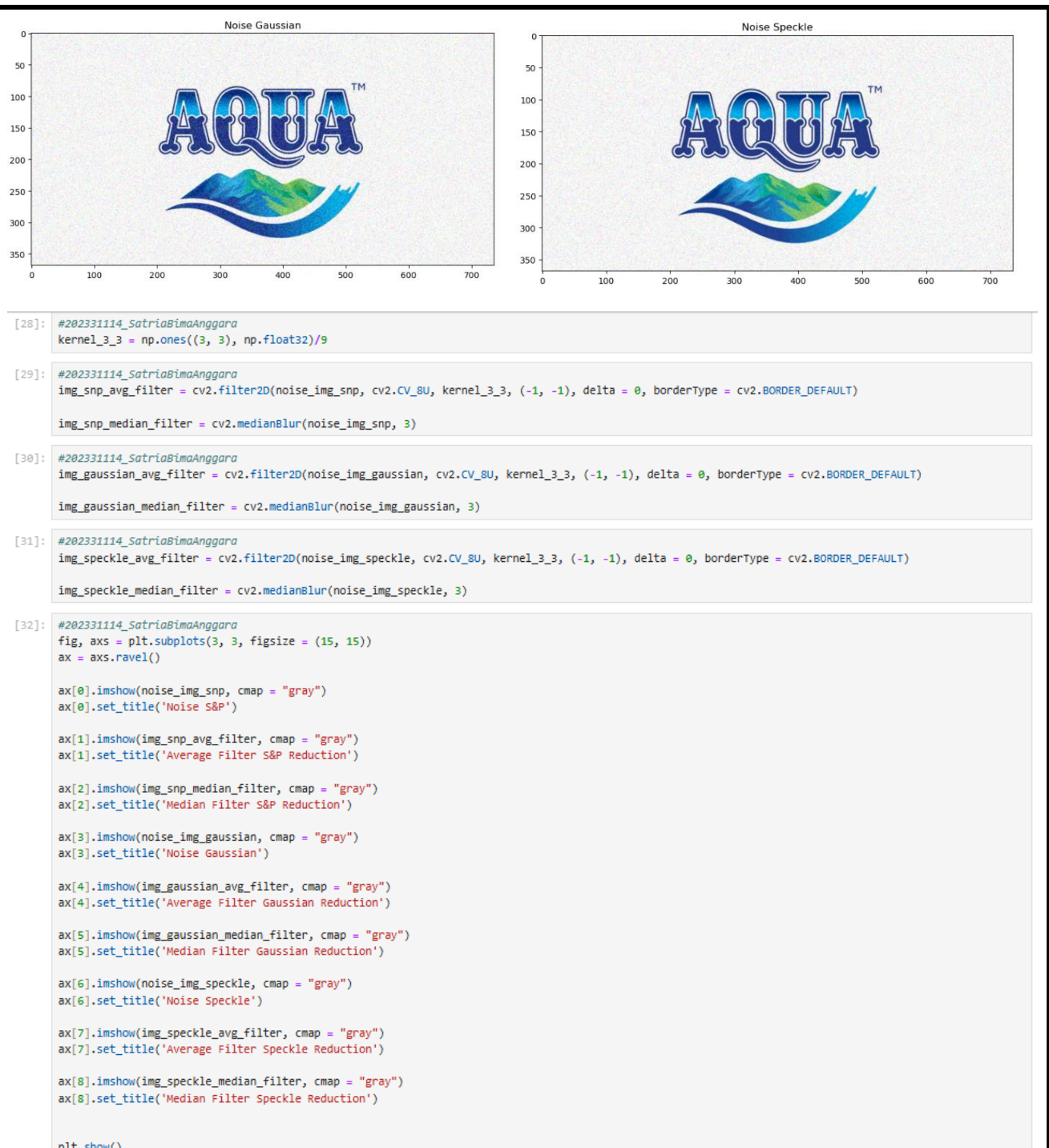
ax[2].imshow(noise_img_gaussian, cmap = "gray")
ax[2].set_title('Noise Gaussian')

ax[3].imshow(noise_img_speckle, cmap = "gray")
ax[3].set_title('Noise Speckle')

plt.show()
```

andom_noise: Fungsi dari skimage untuk menambahkan noise pada citra. Salt & Pepper: Noise acak berupa titik hitam dan putih. Gaussian: Noise acak mengikuti distribusi normal. Speckle: Noise berbentuk bintik-bintik menyebar. uint8 agar cocok untuk ditampilkan dan diproses oleh OpenCV.



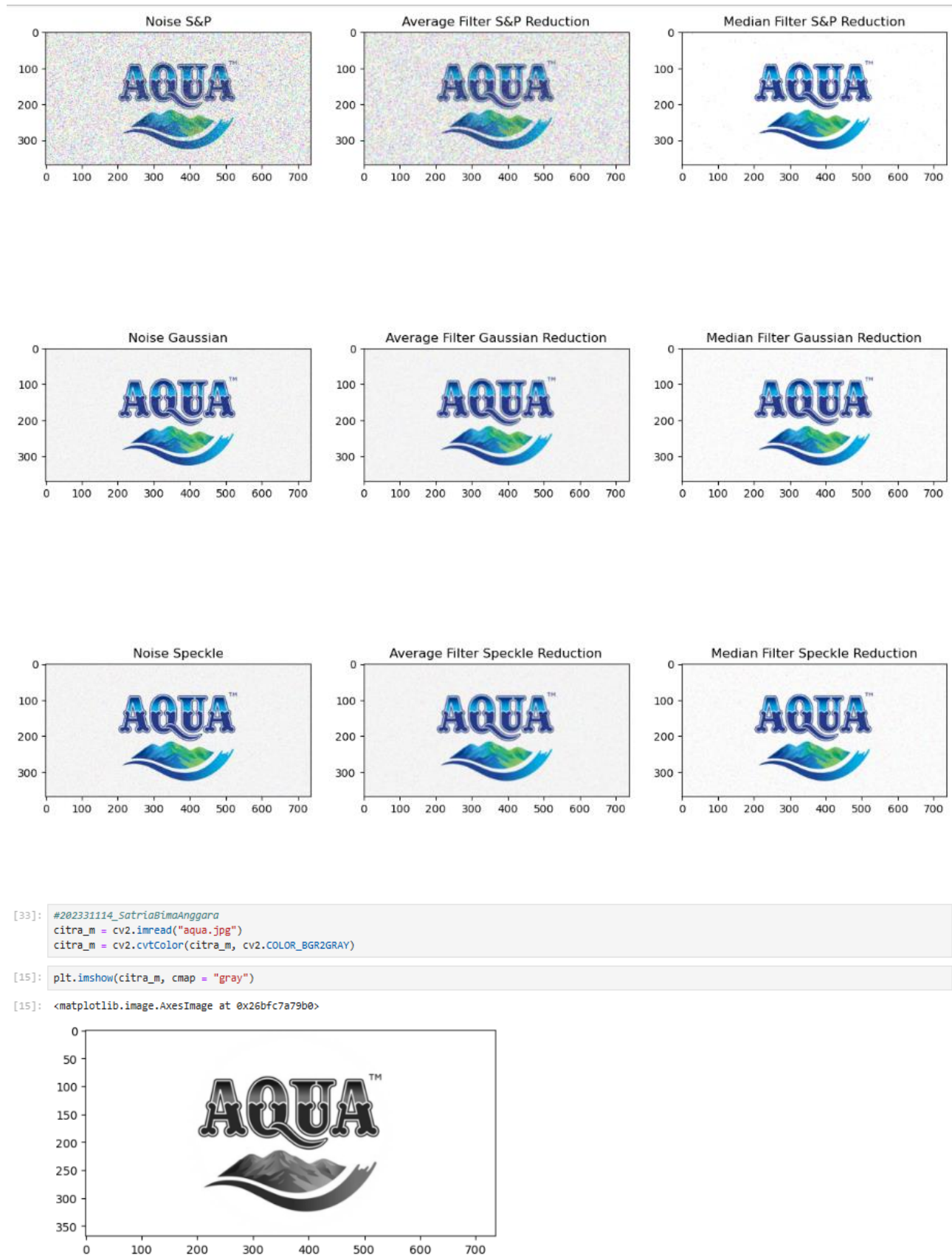


Average Filter (filter rata-rata): Menghaluskan gambar dengan merata-ratakan nilai pixel dalam jendela 3x3. Median Filter: Mengganti nilai pixel dengan nilai median dalam jendela, efektif untuk menghilangkan salt & pepper noise.

fig, axs = plt.subplots(3, 3, figsize=(15, 15)) Menampilkan perbandingan:

- Gambar dengan noise.
- Gambar setelah di-filter dengan average.
- Gambar setelah di-filter dengan median.

Output masing masing citra:



Cv2.imread("aqua.jpg") untuk memasukan gambar ke kode, gambar ini akan di ubah sesuai dengan noise noise.

Praktikum 3

```
[34]: #202331114_SatriaBimaAnggara
copyCitra1 = citra_m.copy().astype(float)

m1, n1 = copyCitra1.shape
output1 = np.empty([m1, n1])

print("Shape copy citra 1 : ", copyCitra1.shape)
print("Shape output citra 1 : ", output1.shape)

print("m1 : ", m1)
print("n1 : ", n1)

print()

Shape copy citra 1 : (368, 736)
Shape output citra 1 : (368, 736)
m1 : 368
n1 : 736

[35]: #202331114_SatriaBimaAnggara
for baris in range(0, m1-1):
    for kolom in range(0, n1-1):
        a1 = baris
        b1 = kolom
        jumlah = copyCitra1[a1-1, b1-1] + copyCitra1[a1-1, b1] + copyCitra1[a1-1, b1+1] + \
        copyCitra1[a1, b1-1] + copyCitra1[a1, b1] + copyCitra1[a1, b1+1] + \
        copyCitra1[a1+1, b1-1] + copyCitra1[a1+1, b1] + copyCitra1[a1+1, b1+1]
        output1[a1,b1] = (1/9 * jumlah)

output1 = np.asarray(output1, dtype = np.uint8)

[36]: #202331114_SatriaBimaAnggara
fig, axs = plt.subplots(1, 2, figsize = (10, 10))
ax = axs.ravel()

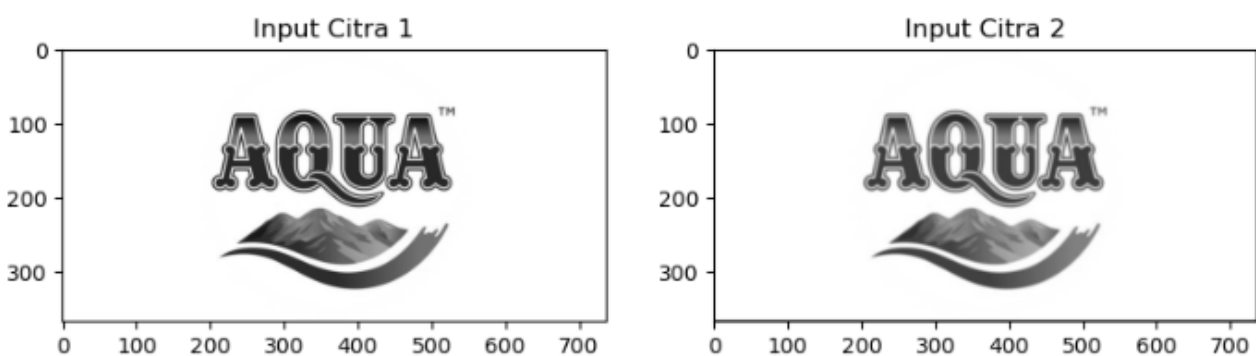
ax[0].imshow(citra_m, cmap = "gray")
ax[0].set_title('Input Citra 1')

ax[1].imshow(output1, cmap = "gray")
ax[1].set_title('Input Citra 2')

plt.show()
```

Kode di atas melakukan beberapa langkah pemrosesan citra, termasuk mengubah citra menjadi format float, mengekstrak bagian tertentu dari citra, dan menampilkan hasilnya dalam skala abu-abu. Ini bisa digunakan untuk berbagai aplikasi dalam pengolahan citra, seperti filter atau segmentasi.

Output dari kode:



Praktikum 3

```
[37]: #Membuat filter median
#202331114_SatriaBimaAnggara
citra_m2 = citra_m
img_median = citra_m2.copy()
img_median_after = cv2.medianBlur(img_median, 5)

fig, axs = plt.subplots(1, 2, figsize = (10, 10))
ax = axs.ravel()

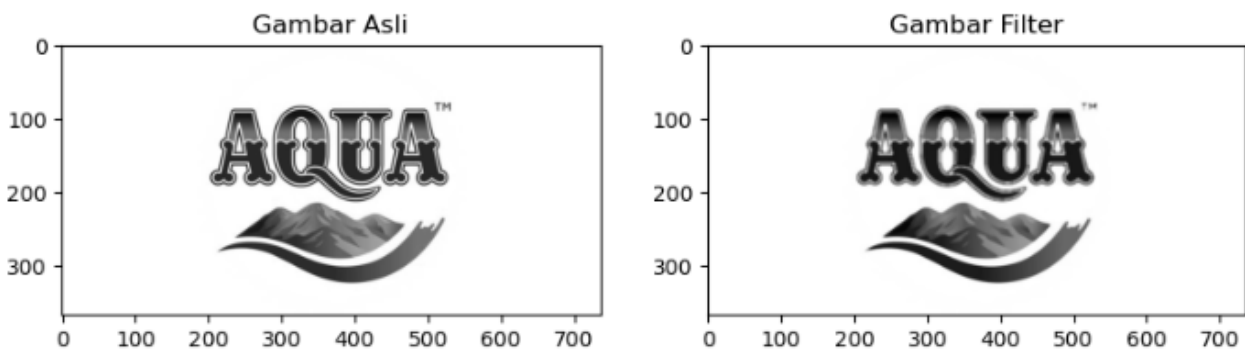
ax[0].imshow(img_median, cmap = "gray")
ax[0].set_title('Gambar Asli')

ax[1].imshow(img_median_after, cmap = "gray")
ax[1].set_title('Gambar Filter')

plt.show()
```

Kode ini menerapkan filter median pada citra untuk mengurangi noise, kemudian menampilkan citra asli dan citra yang telah diproses berdampingan. Filter median efektif dalam menghilangkan noise impulsif sambil mempertahankan tepi citra.

Output dari gambar asli & gambar filter :



```
[38]: #Membuat filter batas
#202331114_SatriaBimaAnggara
citra_m3 = citra_m
copyCitra3 = citra_m3.copy().astype(float)

m1, n1 = copyCitra3.shape
output3 = np.empty([m1-1, n1-1])

for baris in range(1, m1-1):
    for kolom in range(1, n1-1):
        a1 = baris
        b1 = kolom
        arr = np.array([copyCitra3[a1-1, b1-1], copyCitra3[a1-1, b1], copyCitra3[a1-1, b1+1],
                        copyCitra3[a1, b1-1], copyCitra3[a1, b1], copyCitra3[a1, b1+1],
                        copyCitra3[a1+1, b1], copyCitra3[a1+1, b1+1]])

        minPiksel = np.amin(arr)
        maxPiksel = np.amax(arr)

        if (copyCitra3[baris, kolom] < minPiksel):
            output3[baris, kolom] = minPiksel
        else :
            if (copyCitra3[baris, kolom] > maxPiksel):
                output3[baris, kolom] = maxPiksel
            else :
                output3[baris, kolom] = copyCitra3[baris, kolom]

output3 = np.asarray(output3, dtype = np.uint8)

fig, axs = plt.subplots(1, 2, figsize = (10, 10))
ax = axs.ravel()

ax[0].imshow(citra_m3, cmap = "gray")
ax[0].set_title('Gambar Asli')

ax[1].imshow(output3, cmap = "gray")
ax[1].set_title('Gambar Filter Batas')

plt.show()
```

Praktikum 3

ini mengimplementasikan filter sederhana yang menghitung nilai maksimum dan minimum dari setiap piksel (meskipun logika untuk menjelajahi piksel di sekitarnya tampaknya tidak lengkap). Hasilnya adalah citra yang tertapis, yang ditampilkan berdampingan dengan citra asli.

Output dari Gambar Asli & Gambar Filter Batas :

