

LAPORAN TEORI

PENGOLAHAN CITRA DIGITAL



NAMA : Putri Azzahra

NIM : 202331080

KELAS : B

DOSEN : Ir.Darma Rusjdi, M.Kom

NO.PC :

ASISTEN : 1. Davina Najwa Ermawan
2. Fakhrol Fauzi Nugraha Tarigan
3. Viana Salsabila Fairuz Syahla
4. Muhammad Hanief Febriansyah

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

1. Jelaskan konsep dasar dari operasi konvolusi dalam pengolahan citra digital
Konvolusi itu seperti cara sistem komputer “melihat” dan memahami gambar dengan memproses piksel-piksennya. Misalnya, kalau kita ingin gambar menjadi lebih halus atau mendeteksi garis pada gambar, maka kita pakai konvolusi. Caranya, kita pakai filter kecil (kotak angka, seperti 3x3) yang digeser ke seluruh gambar, lalu dihitung hasilnya untuk tiap piksel.
2. Apa yang dimaksud dengan metode ketetanggaan dalam pengolahan citra
Maksudnya adalah saat sistem melihat satu piksel, dia juga melihat piksel-piksel di sekitarnya (tetangga dekatnya). Nilai piksel baru akan dipengaruhi oleh nilai-nilai tetangganya itu. Misalnya, kalau tetangga sekitarnya terang semua, piksel tersebut juga ikut diterangi.
3. Jelaskan prinsip kerja average filter (filter rata-rata) dan median filter dalam mengurangi noise. Apa kelebihan dan kekurangannya?
 - **Average Filter:** mengambil nilai rata-rata dari sekeliling piksel. Cocok untuk menghaluskan gambar, tapi bisa bikin gambar jadi buram.
 - **Median Filter:** mengambil nilai tengah dari piksel-piksel tetangga. Lebih bagus untuk menghilangkan noise berbentuk bintik (seperti titik hitam/putih), dan tetap menjaga tepi gambar agar tidak terlalu buram.

4. Apa itu spackle noise dan bagaimana ia berbeda dari gaussian noise?

Speckle noise adalah gangguan seperti bintik-bintik kecil yang sering muncul di gambar hasil medis atau radar.

Gaussian noise seperti kabut atau butiran halus yang menyebar rata.

Perbedaannya, speckle noise muncul karena gangguan sistem, sedangkan Gaussian noise terjadi karena sensor atau proses pengambilan gambar.

5. Uraikan manfaat dari operasi konvolusi dalam bidang komputer visi (computer vision)

Konvolusi sangat berguna dalam komputer untuk mengenali gambar. Misalnya, agar komputer bisa mengenali wajah, benda, atau jalan dalam mobil otomatis. Tanpa konvolusi, komputer tidak bisa tahu mana bagian penting dari sebuah gambar.

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA DIGITAL



NAMA : Putri Azzahra

NIM : 202331080

KELAS : B

DOSEN : Ir.Darma Rusjdi, M.Kom

NO.PC :

ASISTEN : 1. Davina Najwa Ermawan

2. Fakhrol Fauzi Nugraha Tarigan

3. Viana Salsabila Fairuz Syahla

4. Muhammad Hanief Febriansyah

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

Jelaskan tiap – tiap langkah pengerjaan!

```
[32]: #202331080_Putri Azzahra
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

- Mengimpor pustaka yang dibutuhkan.
cv2 untuk pengolahan citra, numpy untuk operasi numerik, dan matplotlib.pyplot untuk menampilkan gambar.

```
[33]: #202331080_Putri Azzahra

citra_b = cv2.imread("Wonderful Indonesia.jpeg")
citra_b = cv2.cvtColor(citra_b, cv2.COLOR_BGR2GRAY)
print(citra_b.shape)

plt.imshow(citra_b, cmap = "gray")

(317, 474)

[33]: <matplotlib.image.AxesImage at 0x2274b1fc410>
```



- Baris ini untuk membaca gambar berwarna dari file "Wonderful Indonesia.jpeg" dan menyimpannya ke dalam variabel citra_b.
- citra_b = cv2.cvtColor(citra_b, cv2.COLOR_BGR2GRAY).
mengubah gambar yang tadinya berwarna (BGR) menjadi grayscale (abu-abu), sehingga hanya terdiri dari 1 channel intensitas.
- print(citra_b.shape)
Untuk Menampilkan ukuran dimensi dari gambar grayscale (jumlah baris dan kolom piksel).
- plt.imshow(citra_b, cmap="gray").
Menampilkan gambar grayscale dengan colormap abu-abu agar sesuai dengan tampilan citra hitam-putih.

```
[34]: #202331080_Putri Azzahra
kernel = np.array([[ -1,  0,  -1],[ 0,  4,  0],[ -1,  0,  -1],])
```

- Membuat kernel atau matriks filter 3x3. Kernel ini biasanya digunakan untuk mendeteksi tepi (edge detection), karena memiliki nilai tengah positif yang besar dan dikelilingi oleh nilai negatif.

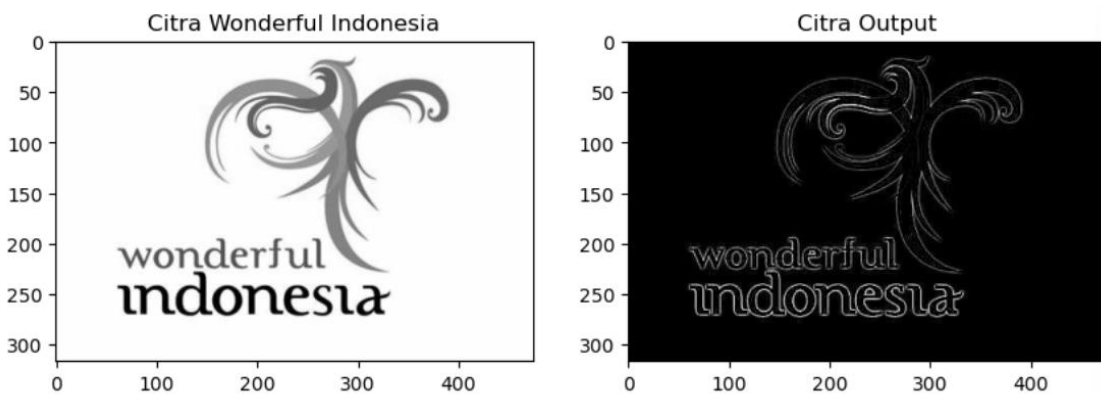
```
[35]: #202331080_Putri Azzahra
citraOutput = cv2.filter2D(citra_b, -1, kernel)

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

ax[0].imshow(citra_b, cmap = "gray")
ax[0].set_title("Citra Wonderful Indonesia")

ax[1].imshow(citraOutput, cmap = "gray")
ax[1].set_title("Citra Output")

plt.show()
```



- `citraOutput = cv2.filter2D(citra_b, -1, kernel)`.
Menerapkan kernel tersebut ke gambar `citra_b` menggunakan filter 2D (konvolusi), hasilnya disimpan di `citraOutput`.
- `fig, axs = plt.subplots(1, 2, figsize=(10,10))`
`ax = axs.ravel()`
Membuat dua kolom untuk menampilkan dua gambar berdampingan dengan ukuran tampilan 10x10.
- `ax[0].imshow(citra_b, cmap="gray")`
`ax[0].set_title("Citra Wonderful Indonesia")`.
Menampilkan gambar asli (grayscale) di kolom kiri dengan judul.
- `ax[1].imshow(citraOutput, cmap="gray")`
`ax[1].set_title("Citra Output")`.
Menampilkan hasil filter (citra setelah edge detection) di kolom kanan dengan judul.
- `plt.show()`.
Menampilkan semua plot yang sudah dibuat.

```
[37]: #202331080_Putri Azzahra
citra_r = cv2.imread("Wonderful Indonesia.jpeg")
citra_r = cv2.cvtColor(citra_r, cv2.COLOR_BGR2RGB)

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



- `citra_r = cv2.imread("Wonderful Indonesia.jpeg")`.
Membaca ulang gambar aslinya dalam format BGR.
- `citra_r = cv2.cvtColor(citra_r, cv2.COLOR_BGR2RGB)`.
Mengubah format warna dari BGR ke RGB agar tampilannya benar di matplotlib (karena OpenCV menggunakan BGR, sedangkan matplotlib pakai RGB).
- `plt.imshow(citra_r)`
`plt.show()`.
Menampilkan gambar berwarna asli ke layar.

```
[38]: #202331080_Putri Azzahra
      from skimage.util import random_noise
```

- Baris ini mengimpor fungsi `random_noise` dari pustaka `skimage.util`, yang digunakan untuk menambahkan noise (gangguan) pada gambar.

```
[39]: #202331080_Putri Azzahra
      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_speckle = np.array(255*noise_img_speckle, dtype = 'uint8')
```

- `noise_img_snp = random_noise(citra_r, mode="s&p", amount=0.2)`.
Menambahkan salt & pepper noise ke citra `citra_r`. Jenis noise ini berupa bintang putih (salt) dan hitam (pepper). `amount=0.2` berarti 20% piksel akan terganggu.
- `noise_img_gaussian = random_noise(citra_r, mode="gaussian", mean=0, var=0.01)`.
Menambahkan Gaussian noise, yaitu gangguan acak dengan distribusi normal (rata-rata 0 dan varians 0.01) ke citra.
- `noise_img_speckle = random_noise(citra_r, mode="speckle")`.

Menambahkan speckle noise, biasanya muncul pada citra radar atau medis dan bersifat seperti bintik halus.

- `noise_img_snp = np.array(255*noise_img_snp, dtype='uint8')`
`noise_img_speckle = np.array(255*noise_img_speckle, dtype='uint8')`.
 Hasil dari `random_noise` berbentuk float (0-1), baris ini mengalikannya dengan 255 lalu mengonversi ke uint8 agar bisa ditampilkan sebagai gambar 8-bit.

```
[40]: #202331080_Putri Azzahra
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')

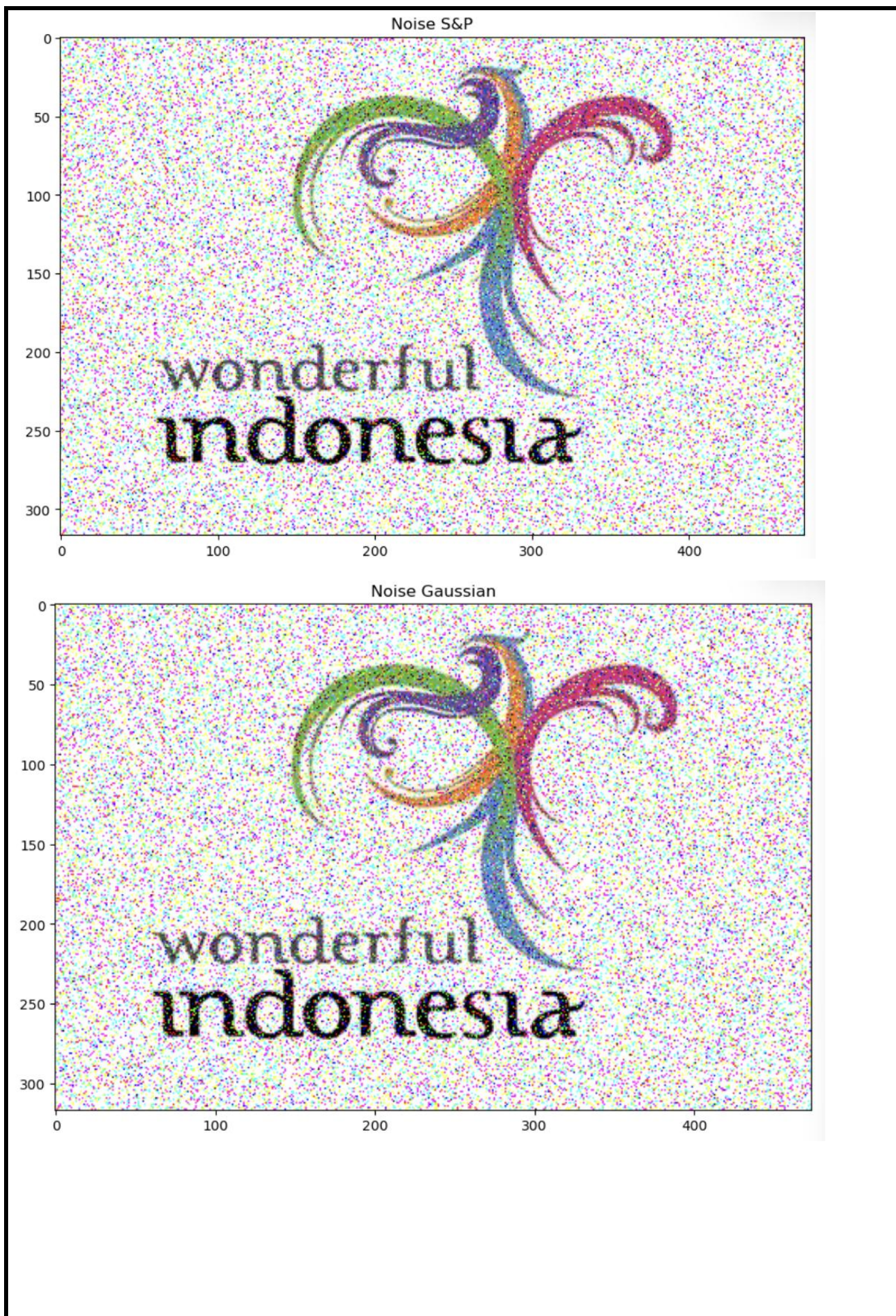
ax[1].imshow(noise_img_snp, cmap = "gray")
ax[1].set_title('Noise S&P')

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

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

plt.show()
```







- `fig, axs = plt.subplots(4, 1, figsize=(10, 40))`
`ax = axs.ravel()`
 Membuat 4 plot gambar dalam satu kolom (4 baris, 1 kolom), dan meratakan array `axs` agar bisa diakses dengan indeks `[0]`, `[1]`, dst.
- `ax[0].imshow(citra_r, cmap="gray")`
`ax[0].set_title('Citra Asli')`
 Menampilkan gambar asli dengan colormap abu-abu dan memberi judul "Citra Asli".
- `ax[1].imshow(noise_img_snp, cmap="gray")`
`ax[1].set_title('Noise S&P')`
 Menampilkan gambar yang telah ditambah noise salt & pepper.
- `ax[2].imshow(noise_img_snp, cmap="gray")`
`ax[2].set_title('Noise Gaussian')`
 Terjadi kesalahan kecil: harusnya ini `noise_img_gaussian`, tapi ditulis `noise_img_snp` lagi. Ini membuat tampilan Gaussian salah.
- `ax[3].imshow(noise_img_speckle, cmap="gray")`
`ax[3].set_title('Noise Speckle')`
 Menampilkan gambar dengan speckle noise.

```
[42]: #202331080_Putri Azzahra
      kernel_3_3 = np.ones((3, 3), np.float32)/9
```

- Membuat kernel 3x3 untuk filter rata-rata (average). Semua nilai kernel bernilai 1/9.

```
[43]: #202331080_Putri Azzahra
img_snp_avg_filter = cv2.filter2D(noise_img_snp, cv2.CV_8U, kernel_3_3, (-1, -1), delta = 0, borderType = cv2.BORDER_DEFAULT)

img_snp_median_filter = cv2.medianBlur(noise_img_snp, 3)
```

- `img_snp_avg_filter = cv2.filter2D(noise_img_snp, cv2.CV_8U, kernel_3_3, (-1, -1), delta=0, borderType=cv2.BORDER_DEFAULT)`
Menerapkan filter rata-rata pada gambar dengan S&P noise. Fungsi `cv2.filter2D` melakukan konvolusi.
- `img_snp_median_filter = cv2.medianBlur(noise_img_snp, 3).`
Menerapkan median filter ukuran 3x3 untuk mengurangi S&P noise. Median filter sangat efektif untuk noise jenis ini.

```
[44]: #202331080_Putri Azzahra
noise_img_gaussian_uint8 = np.clip(noise_img_gaussian, 0, 255).astype(np.uint8)

kernel_3_3 = np.ones((3, 3), np.float32) / 9
img_gaussian_avg_filter = cv2.filter2D(noise_img_gaussian_uint8, -1, kernel_3_3)

img_gaussian_median_filter = cv2.medianBlur(noise_img_gaussian_uint8, 3)
```

- `noise_img_gaussian_uint8 = np.clip(noise_img_gaussian, 0, 255).astype(np.uint8)`
Membatasi nilai pixel hasil Gaussian noise agar tetap di rentang 0–255 dan ubah ke uint8 agar bisa diproses lebih lanjut.
- `kernel_3_3 = np.ones((3, 3), np.float32) / 9`
Membuat kernel rata-rata 3x3.
- `img_gaussian_avg_filter = cv2.filter2D(noise_img_gaussian_uint8, -1, kernel_3_3)`
Menerapkan filter rata-rata pada gambar Gaussian.
- `img_gaussian_median_filter = cv2.medianBlur(noise_img_gaussian_uint8, 3)`
Menerapkan median filter ke gambar Gaussian.

```
[45]: #202331080_Putri Azzahra
img_speckle_avg_filter = cv2.filter2D(noise_img_speckle, cv2.CV_8U, kernel_3_3, (-1, -1), delta = 0, borderType = cv2.BORDER_DEFAULT)

img_speckle_median_filter = cv2.medianBlur(noise_img_speckle, 3)
```

- `img_speckle_avg_filter = cv2.filter2D(noise_img_speckle, cv2.CV_8U, kernel_3_3, (-1, -1), delta=0, borderType=cv2.BORDER_DEFAULT)`
Menerapkan filter rata-rata ke gambar speckle noise.
- `img_speckle_median_filter = cv2.medianBlur(noise_img_speckle, 3).`
Menerapkan median filter ke gambar dengan speckle noise.

Laporan 3

```
[46]: #202331080_Putri Azzahra
fig, axs = plt.subplots(3, 3, figsize = (15, 15))
ax = axs.ravel()

ax[0].imshow(noise_img_snp, cmap = "gray")
ax[0].set_title('Noise S&P')

ax[1].imshow(img_snp_avg_filter, cmap = "gray")
ax[1].set_title('Average Filter S&P Reduction')

ax[2].imshow(img_snp_median_filter, cmap = "gray")
ax[2].set_title('Median Filter S&P Reduction')

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

ax[4].imshow(img_gaussian_avg_filter, cmap = "gray")
ax[4].set_title('Average Filter Gaussian Reduction')

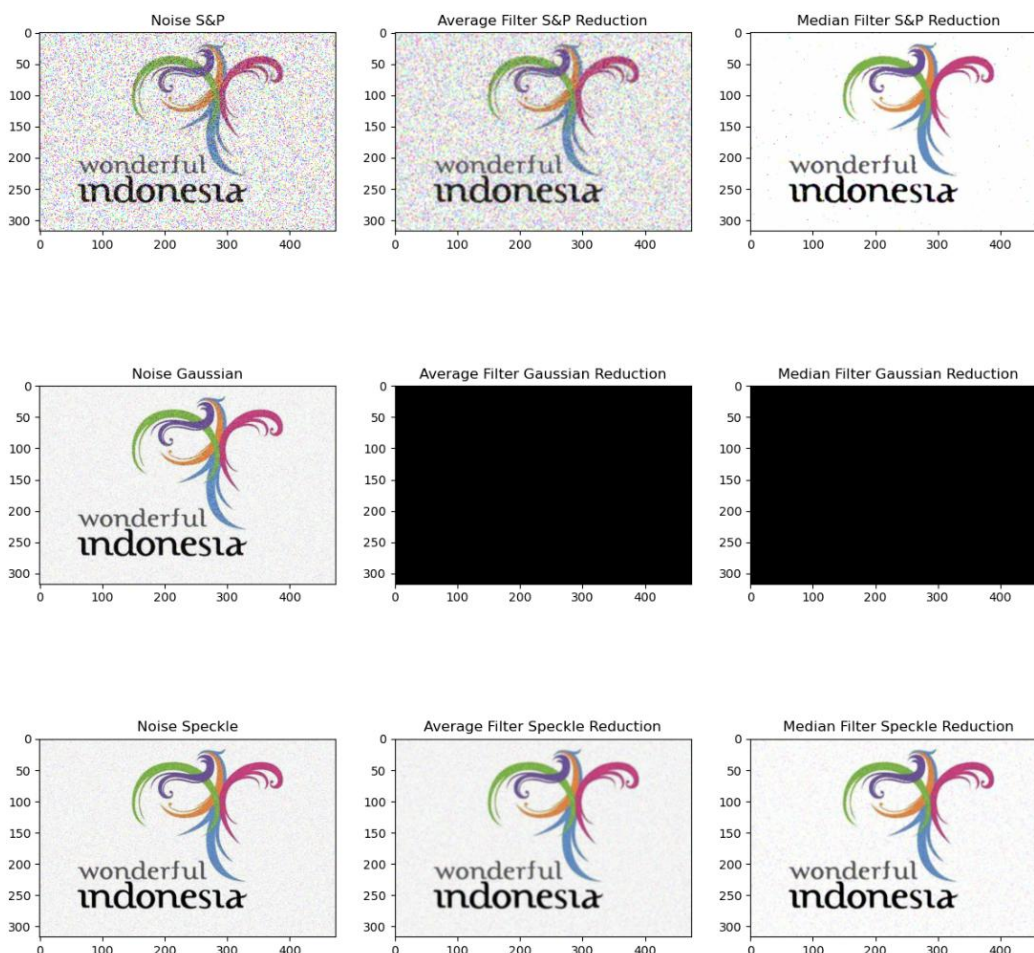
ax[5].imshow(img_gaussian_median_filter, cmap = "gray")
ax[5].set_title('Median Filter Gaussian Reduction')

ax[6].imshow(noise_img_speckle, cmap = "gray")
ax[6].set_title('Noise Speckle')

ax[7].imshow(img_speckle_avg_filter, cmap = "gray")
ax[7].set_title('Average Filter Speckle Reduction')

ax[8].imshow(img_speckle_median_filter, cmap = "gray")
ax[8].set_title('Median Filter Speckle Reduction')

plt.show()
```



- `fig, axs = plt.subplots(3, 3, figsize = (15, 15)).`
Membuat grid subplot berukuran 3 baris \times 3 kolom (total 9 kotak gambar), dengan ukuran total gambar 15x15 inci. `fig` adalah objek gambar utamanya, `axs` berisi array untuk semua subplot.
- `ax = axs.ravel().`
Mengubah struktur array 2D (`axs`) menjadi 1D agar mudah diakses dengan indeks [0] sampai [8].
- `ax[0].imshow(noise_img_snp, cmap = "gray")`
`ax[0].set_title('Noise S&P').`
Menampilkan gambar noise tipe Salt and Pepper (S&P) pada subplot pertama (pojok kiri atas), dengan colormap abu-abu. Judul subplot diberi nama "Noise S&P".
- `ax[1].imshow(img_snp_avg_filter, cmap = "gray")`
`ax[1].set_title('Average Filter S&P Reduction')`
Menampilkan hasil filter rata-rata pada gambar noise S&P di subplot kedua, dengan judul "Average Filter S&P Reduction".
- `ax[2].imshow(img_snp_median_filter, cmap = "gray")`
`ax[2].set_title('Median Filter S&P Reduction')`
Menampilkan hasil median filter pada gambar noise S&P di subplot ketiga, dengan judul "Median Filter S&P Reduction".
- `ax[3].imshow(noise_img_gaussian, cmap = "gray")`
`ax[3].set_title('Noise Gaussian')`
Menampilkan gambar dengan noise Gaussian di subplot keempat. Gaussian noise menyerupai noise acak normal.
- `ax[4].imshow(img_gaussian_avg_filter, cmap = "gray")`
`ax[4].set_title('Average Filter Gaussian Reduction')`
Menampilkan hasil average filter pada gambar noise Gaussian di subplot kelima.
- `ax[5].imshow(img_gaussian_median_filter, cmap = "gray")`
`ax[5].set_title('Median Filter Gaussian Reduction')`
Menampilkan hasil median filter pada gambar Gaussian noise di subplot keenam.
- `ax[6].imshow(noise_img_speckle, cmap = "gray")`
`ax[6].set_title('Noise Speckle')`
Menampilkan gambar dengan noise speckle (bintik-bintik seperti pada gambar ultrasound) di subplot ketujuh.
- `ax[7].imshow(img_speckle_avg_filter, cmap = "gray")`
`ax[7].set_title('Average Filter Speckle Reduction')`
Menampilkan hasil average filter pada gambar noise speckle di subplot kedelapan.
- `ax[8].imshow(img_speckle_median_filter, cmap = "gray")`
`ax[8].set_title('Median Filter Speckle Reduction')`
Menampilkan hasil median filter pada gambar noise speckle di subplot kesembilan.
- `plt.show().`
Menampilkan semua subplot yang sudah dibuat dan diisi dengan gambar di atas.

```
[47]: #202331080_Putri Azzahra
citra_m = cv2.imread("Wonderful Indonesia.jpeg")
citra_m = cv2.cvtColor(citra_m, cv2.COLOR_BGR2GRAY)
```

```
[48]: #202331080_Putri Azzahra
plt.imshow(citra_m, cmap = "gray")
```

```
[48]: <matplotlib.image.AxesImage at 0x2274a43f200>
```



- `citra_m = cv2.imread("Wonderful_Indonesia.jpeg")`
Fungsi `cv2.imread()` dari OpenCV digunakan untuk membaca file gambar berformat .jpeg.
- `citra_m = cv2.cvtColor(citra_m, cv2.COLOR_BGR2GRAY)`
Fungsi `cv2.cvtColor()` digunakan untuk konversi warna, dan `cv2.COLOR_BGR2GRAY` adalah konversi dari format BGR ke grayscale (hitam putih).
- `plt.imshow(citra_m, cmap = "gray")`
`cmap="gray"` berarti warna yang digunakan untuk menampilkan

```
[49]: #202331080_Putri Azzahra
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 : (317, 474)
Shape output citra 1 : (317, 474)
m1 : 317
n1 : 474
```


- `copyCitra1 = citra_m.copy().astype(float)`
Menyalin gambar `citra_m` ke variabel baru `copyCitra1`, lalu mengubah tipe datanya menjadi float agar proses perhitungan (seperti pembagian) bisa lebih presisi.
- `m1, n1 = copyCitra1.shape`
Menyimpan ukuran (jumlah baris dan kolom) dari gambar ke variabel `m1` (tinggi) dan `n1` (lebar).
- `output1 = np.empty([m1, n1])`
Membuat matriks kosong (dengan ukuran sama seperti gambar) yang akan dipakai untuk menyimpan hasil akhirnya.
- `print("Shape copy citra 1 : ", copyCitra1.shape)`
`print("Shape output citra 1 : ", output1.shape)`
`print("m1 : ", m1)`
`print("n1 : ", n1)`
Menampilkan informasi ukuran gambar dan matriks hasil ke layar, supaya bisa dicek apakah sesuai.

```
[50]: #202331080_Putri Azzahra
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)
```

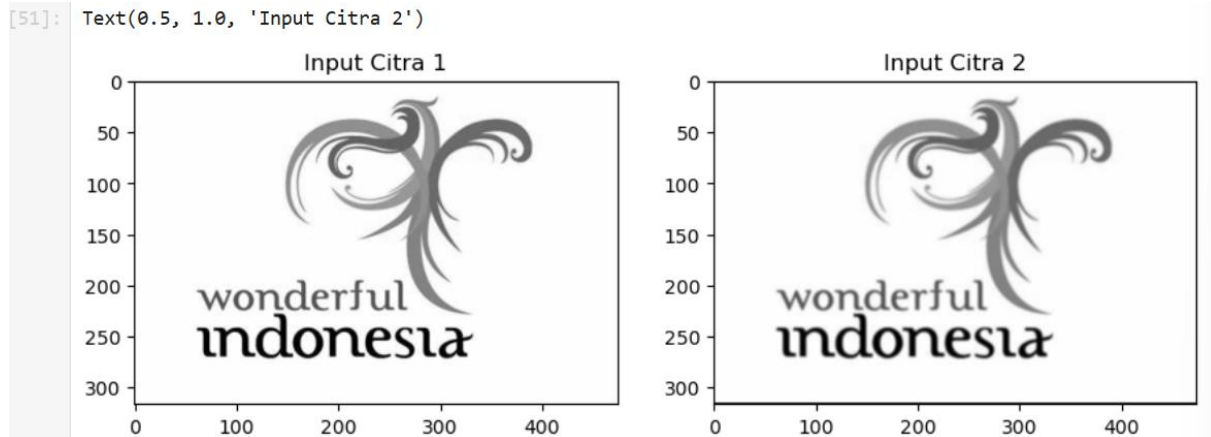
- `for baris in range(0, m1-1):`
 `for kolom in range(0, n1-1):`
Melakukan perulangan baris dan kolom di setiap piksel gambar, kecuali baris terakhir dan kolom terakhir (supaya tidak keluar batas saat mengambil tetangga piksel).
- `a1 = baris`
 `b1 = kolom`
Menyimpan posisi piksel saat ini ke variabel `a1` dan `b1` agar mudah dipakai dalam perhitungan.
- `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]`
Mengambil 9 piksel tetangga (termasuk pusat) dalam area 3x3 dan menjumlahkan semua nilainya.
- `output1[a1, b1] = (1/9 * jumlah)`
Menghitung rata-rata nilai dari area 3x3 tadi, lalu menyimpannya ke `output1` di posisi (`a1`, `b1`).
- `output1 = np.asarray(output1, dtype=np.uint8)`
Mengubah tipe data matriks hasil (`output1`) menjadi `uint8` (nilai piksel gambar 0–255), agar bisa ditampilkan atau disimpan sebagai gambar.

```
[51]: #202331080_Putri Azzahra
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')
```

```
[51]: Text(0.5, 1.0, 'Input Citra 2')
```



- 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')

```
[52]: #202331080_Putri Azzahra
#membuat Filter median

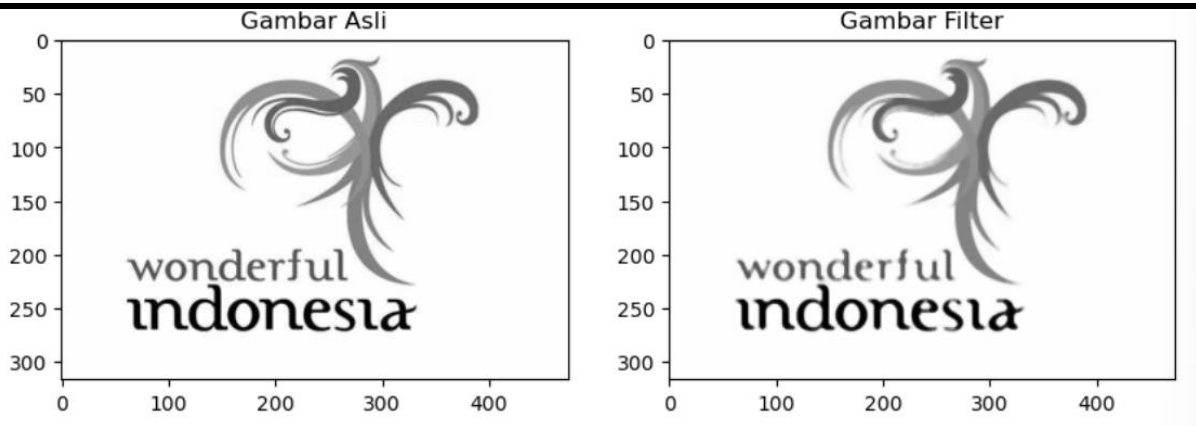
citra_m2 = citra_m
img_median = citra_m2.copy()
img_median_after = cv2.medianBlur(img_median, 5)

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()
```



- `img_median_after = cv2.medianBlur(img_median, 5)`
Gambar Filter (kanan) adalah hasil dari *median filter* dengan kernel 5×5 .

```
[53]: #202331080_Putri Azzahra
#Membuat citra batas
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+1], copyCitra3[a1+1, 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()
```

- `arr = np.array([...])` # ambil 8 tetangga
`minPiksel = np.amin(arr)`
`maxPiksel = np.amax(arr)`
`if (copyCitra3[baris, kolom] < minPiksel):`
 `output3[baris, kolom] = minPiksel`
`elif (copyCitra3[baris, kolom] > maxPiksel):`
 `output3[baris, kolom] = maxPiksel`
`else:`
 `output3[baris, kolom] = copyCitra3[baris, kolom]`
Gambar Asli: Menampilkan logo *Wonderful Indonesia* dalam grayscale.

