

LAPORAN TUGAS
PENGOLAHAN CITRA DIGITAL
DOMAIN FREKUENSI



OLEH :

Ahmad Afil

F 551 22 050

KELAS B

PROGRAM STUDI S1 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TADULAKO
PALU 2024

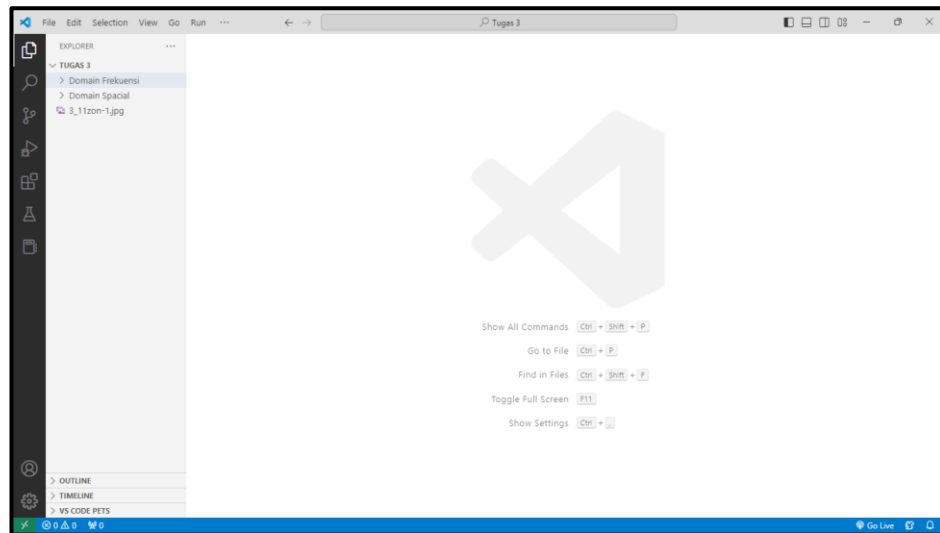
I. TEORI DASAR

Domain frekuensi adalah cara representasi sinyal atau data yang berbeda dari domain waktu, dan merupakan konsep penting dalam analisis sinyal, pengolahan gambar, dan berbagai bidang teknik dan sains. Jika dalam domain waktu, sinyal diuraikan berdasarkan waktu atau urutan kejadian, dalam domain frekuensi, sinyal diuraikan berdasarkan komponen frekuensinya. Konversi antara domain waktu dan frekuensi biasanya dilakukan dengan transformasi Fourier, yang mengubah sinyal dari representasi waktu menjadi representasi frekuensi.

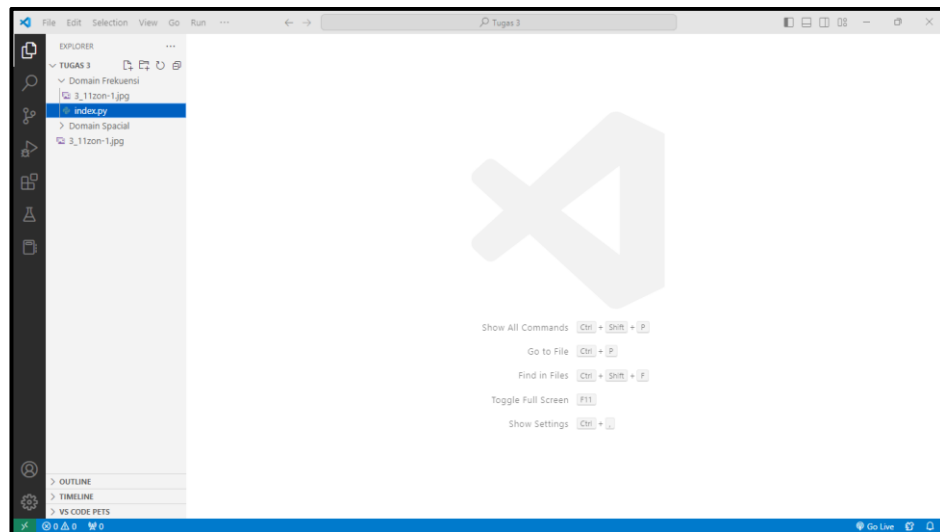
Konsep dasar domain frekuensi didasarkan pada ide bahwa setiap sinyal atau data bisa dianggap sebagai gabungan dari sinyal-sinyal sinusoidal pada berbagai frekuensi, amplitudo, dan fase. Transformasi Fourier memungkinkan kita untuk menganalisis sinyal berdasarkan frekuensi-frekuensi komponennya, mengidentifikasi komponen frekuensi yang dominan, dan memahami bagaimana sinyal berubah dalam domain frekuensi. Analisis ini sangat berguna untuk memahami karakteristik sinyal atau data yang mungkin tidak terlihat dalam domain waktu.

II. LANGKAH KERJA

A. Jalankan aplikasi VSCode.



B. Selanjutnya menambahkan file dengan nama 'index.py'



C. Masukkan Kode Program Berikut.

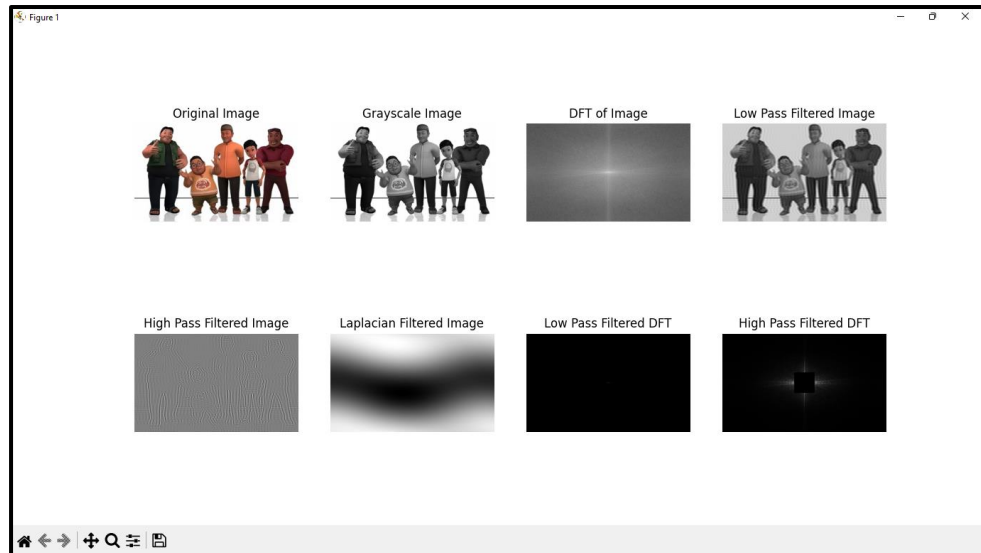
```
Domain Frekuensi > index.py > ...
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 # Fungsi untuk mengubah citra menjadi grayscale
6 def grayscale(image):
7     return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
8
9 # Fungsi untuk menghitung DFT (Discrete Fourier Transform) dari citra
10 def compute_dft(image):
11     return np.fft.fftshift(np.fft.fft2(image))
12
13 # Fungsi untuk menghitung DFT invers dari citra
14 def compute_inverse_dft(dft_image):
15     return np.fft.ifft2(np.fft.ifftshift(dft_image)).real
16
17 # Fungsi untuk menerapkan filter frekuensi rendah pada citra dalam domain frekuensi
18 def apply_low_pass_filter(dft_image, threshold):
19     rows, cols = dft_image.shape
20     center_row, center_col = rows // 2, cols // 2
21     mask = np.zeros((rows, cols), np.uint8)
22     mask[center_row - threshold:center_row + threshold, center_col - threshold:center_col + threshold] = 1
23     filtered_dft_image = dft_image * mask
24     return filtered_dft_image
```

```

26 # Fungsi untuk menerapkan filter frekuensi tinggi pada citra dalam domain frekuensi
27 def apply_high_pass_filter(dft_image, threshold):
28     rows, cols = dft_image.shape
29     center_row, center_col = rows // 2, cols // 2
30     mask = np.ones((rows, cols), np.uint8)
31     mask[center_row - threshold:center_row + threshold, center_col - threshold:center_col + threshold] = 0
32     filtered_dft_image = dft_image * mask
33     return filtered_dft_image
34
35 # Fungsi untuk menerapkan filter Laplacian pada citra dalam domain frekuensi
36 def apply_laplacian_filter(dft_image):
37     rows, cols = dft_image.shape
38     center_row, center_col = rows // 2, cols // 2
39     mask = np.zeros((rows, cols), np.uint8)
40     mask[center_row - 1:center_row + 2, center_col - 1:center_col + 2] = [[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]
41     filtered_dft_image = dft_image * mask
42     return filtered_dft_image
43
44 # Load citra
45 image = cv2.imread('G:\MK\SEMESTER 4\PCD\Tugas 3\Domain Frekuensi\3_11zon-1.jpg')
46
47 # Mengubah citra menjadi grayscale
48 gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
49
50 # Menghitung DFT dari citra grayscale
51 dft_image = np.fft.fft2(gray_image)
52
53 # Menerapkan filter frekuensi rendah pada citra dalam domain frekuensi
54 low_pass_filtered_dft_image = apply_low_pass_filter(dft_image, threshold=50)
55
56 # Menerapkan filter frekuensi tinggi pada citra dalam domain frekuensi
57 high_pass_filtered_dft_image = apply_high_pass_filter(dft_image, threshold=50)
58
59 # Menerapkan filter Laplacian pada citra dalam domain frekuensi
60 laplacian_filtered_dft_image = apply_laplacian_filter(dft_image)
61
62 # Menghitung DFT invers dari citra dalam domain frekuensi yang telah difilter
63 low_pass_filtered_image = np.fft.ifft2(low_pass_filtered_dft_image)
64 high_pass_filtered_image = np.fft.ifft2(high_pass_filtered_dft_image)
65 laplacian_filtered_image = np.fft.ifft2(laplacian_filtered_dft_image)
66
67 # Menampilkan citra-citra hasil pemrosesan
68 plt.figure(figsize=(12, 8))
69
70 plt.subplot(2, 4, 1)
71 plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
72 plt.title('Original Image')
73 plt.axis('off')
74
75 plt.subplot(2, 4, 2)
76 plt.imshow(gray_image, cmap='gray')
77 plt.title('Grayscale Image')
78 plt.axis('off')
79
80 plt.subplot(2, 4, 3)
81 plt.imshow(np.log(1 + np.abs(dft_image)), cmap='gray')
82 plt.title('DFT of Image')
83 plt.axis('off')
84
85 plt.subplot(2, 4, 4)
86 plt.imshow(low_pass_filtered_image, cmap='gray')
87 plt.title('Low Pass Filtered Image')
88 plt.axis('off')
89
90 plt.subplot(2, 4, 5)
91 plt.imshow(high_pass_filtered_image, cmap='gray')
92 plt.title('High Pass Filtered Image')
93 plt.axis('off')
94
95 plt.subplot(2, 4, 6)
96 plt.imshow(laplacian_filtered_image, cmap='gray')
97 plt.title('Laplacian Filtered Image')
98 plt.axis('off')
99
100 plt.subplot(2, 4, 7)
101 plt.imshow(np.abs(low_pass_filtered_dft_image), cmap='gray')
102 plt.title('Low Pass Filtered DFT')
103 plt.axis('off')
104
105 plt.subplot(2, 4, 8)
106 plt.imshow(np.abs(high_pass_filtered_dft_image), cmap='gray')
107 plt.title('High Pass Filtered DFT')
108 plt.axis('off')
109
110 plt.show()

```

III. HASIL PERCOBAAN



IV. ANALISIS

Program di atas merupakan contoh implementasi pemrosesan citra dalam domain frekuensi menggunakan Transformasi Fourier Diskret (DFT) dengan bantuan library OpenCV dan NumPy. Pada awal program, beberapa library yang diperlukan diimpor, termasuk cv2 untuk pemrosesan citra, numpy untuk operasi matriks, dan matplotlib.pyplot untuk menampilkan gambar. Selanjutnya, program mendefinisikan beberapa fungsi yang akan digunakan dalam pemrosesan citra.

Pertama, terdapat fungsi grayscale yang bertugas untuk mengubah citra menjadi citra grayscale (keabuan). Kemudian terdapat fungsi compute_dft yang digunakan untuk menghitung DFT (Discrete Fourier Transform) dari citra grayscale. Fungsi compute_inverse_dft digunakan untuk menghitung DFT invers dari citra dalam domain frekuensi. Selanjutnya, terdapat tiga fungsi untuk menerapkan filter pada citra dalam domain frekuensi, yaitu apply_low_pass_filter untuk filter frekuensi rendah, apply_high_pass_filter untuk filter frekuensi tinggi, dan apply_laplacian_filter untuk filter Laplacian.

Setelah definisi fungsi-fungsi, program memuat citra menggunakan fungsi cv2.imread dengan path file citra yang diberikan. Citra tersebut kemudian diubah menjadi citra grayscale menggunakan fungsi grayscale. Selanjutnya, DFT dari citra grayscale dihitung menggunakan fungsi compute_dft.

Program melanjutkan dengan menerapkan filter frekuensi rendah, filter frekuensi tinggi, dan filter Laplacian pada citra dalam domain frekuensi. Setiap filter diaplikasikan dengan memanggil fungsi yang sesuai, yaitu `apply_low_pass_filter`, `apply_high_pass_filter`, dan `apply_laplacian_filter`.

Setelah citra-citra dalam domain frekuensi telah difilter, DFT invers dari masing-masing citra tersebut dihitung menggunakan fungsi `compute_inverse_dft`. Hasil DFT invers dari masing-masing filter ditampilkan menggunakan fungsi `plt.imshow` dari `matplotlib.pyplot` pada subplot yang sesuai. Subplot-subplot tersebut menampilkan citra asli, citra grayscale, DFT dari citra, citra hasil filter frekuensi rendah, citra hasil filter frekuensi tinggi, citra hasil filter Laplacian, serta DFT dari masing-masing citra hasil filter.

Terakhir, hasil pemrosesan citra ditampilkan dengan menggunakan `plt.show()`. Dengan demikian, program ini mengilustrasikan proses pemrosesan citra dalam domain frekuensi dengan menerapkan filter frekuensi rendah, filter frekuensi tinggi, dan filter Laplacian pada citra grayscale.