**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**

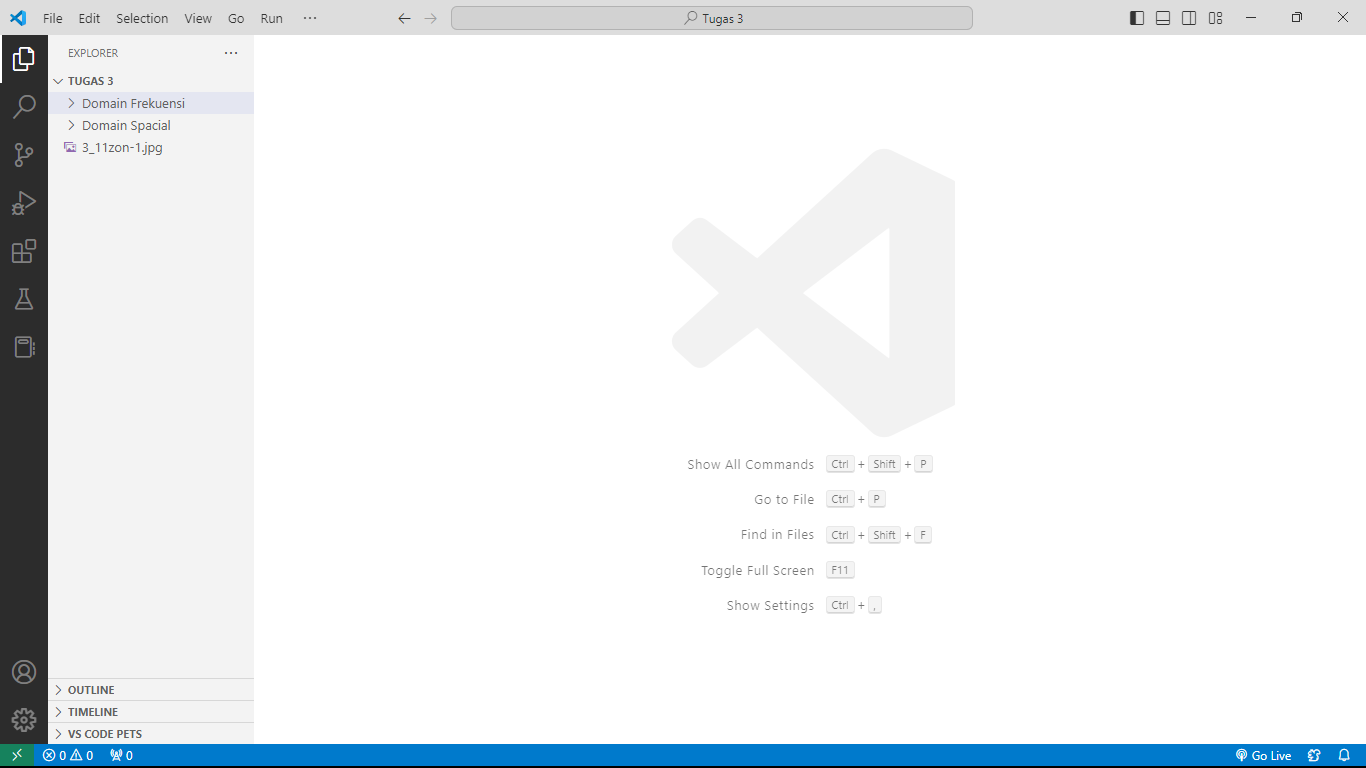
**PALU 2024**

1. **TEORI DASAR**

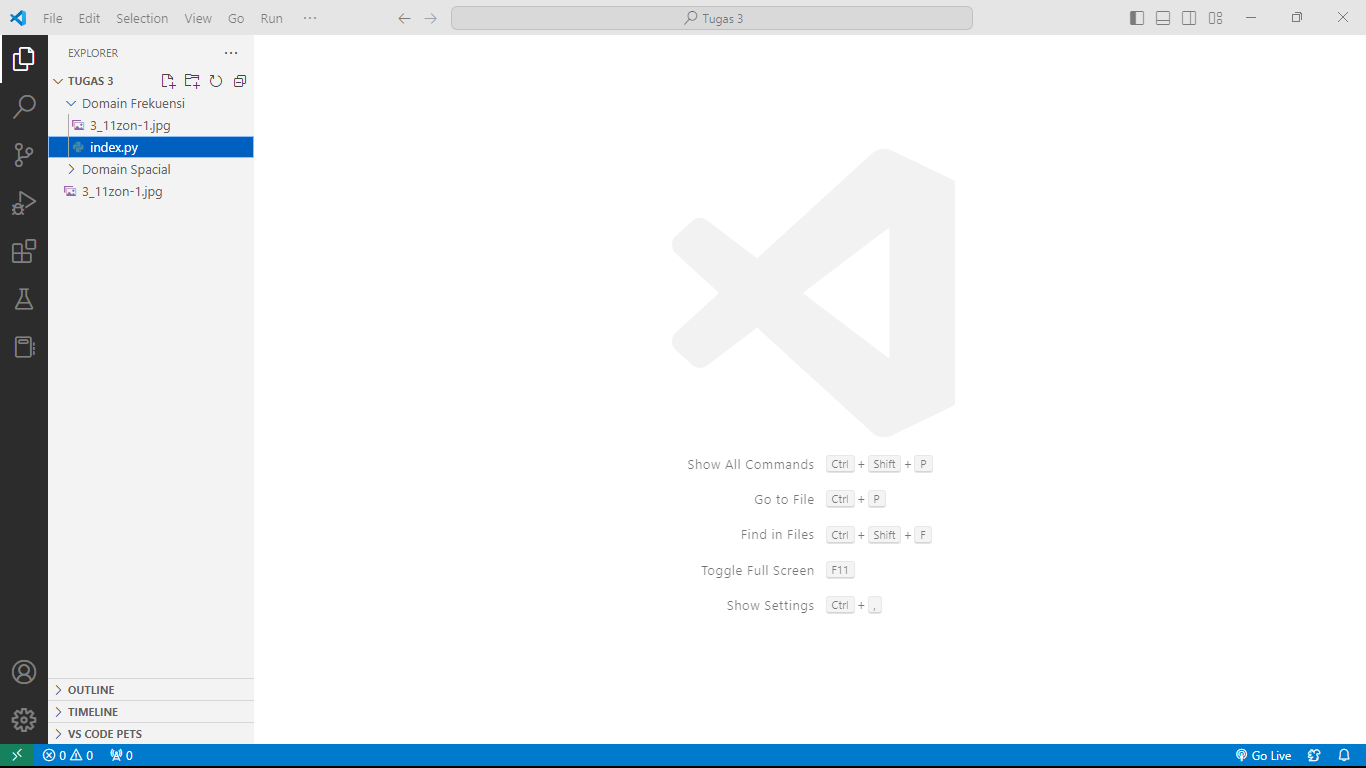
Domain frekuensi adalah cara representasi sinyal atau data yang berbeda dari domain waktu, dan merupakan konsep penting dalam analisis sinyal, pengolahan gambar, dan berbagai bidang teknik dan sains. Jika dalam domain waktu, sinyal diuraikan berdasarkan waktu atau urutan kejadian, dalam domain frekuensi, 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.

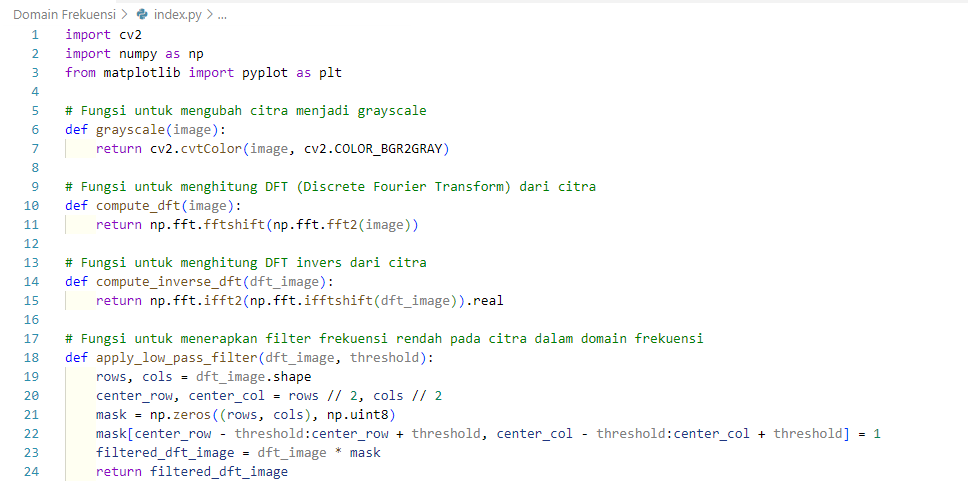
1. **LANGKAH KERJA**
   * 1. Jalankan aplikasi *VSCode.*

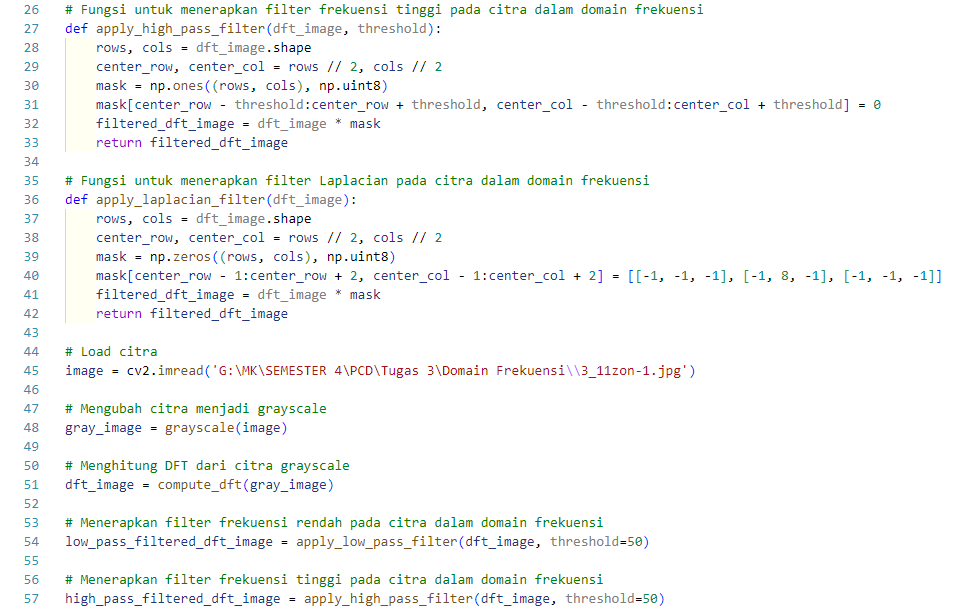


* + 1. Selanjutnya menambahkan file dengan nama ‘index.py’

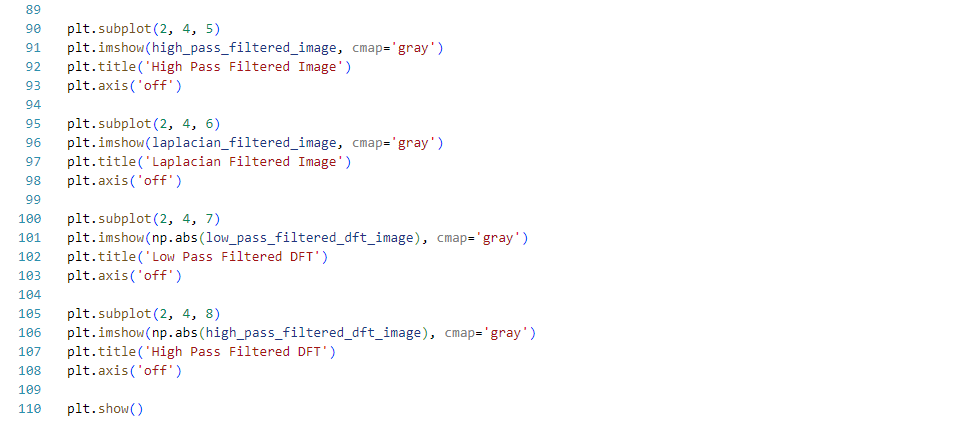


* + 1. Masukkan Kode Program Berikut.

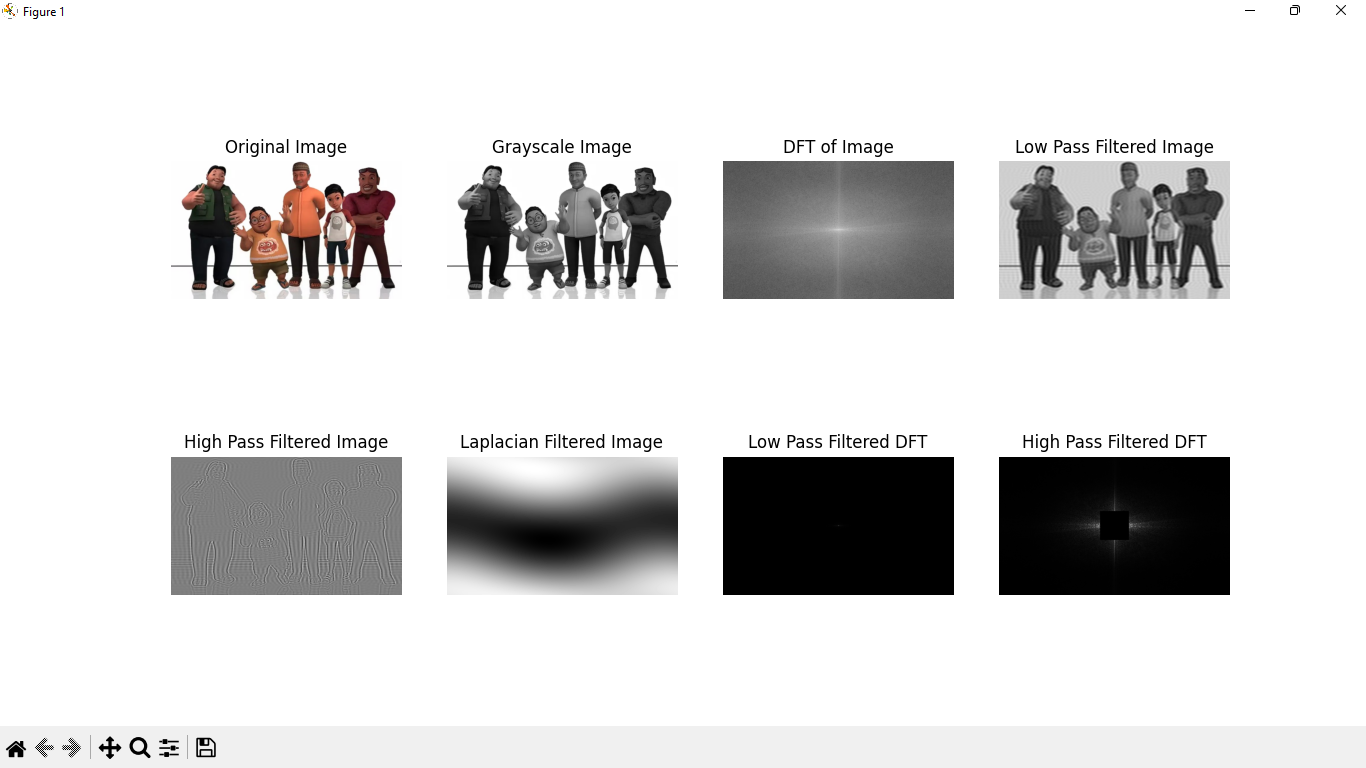








1. **HASIL PERCOBAAN**

****

1. **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 ditampikan 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.