Berikut adalah penjelasan dari setiap baris kode:

1. **import numpy as np** dan **import cv2 as cv**: Mengimpor library NumPy (untuk manipulasi array) dan OpenCV (untuk pengolahan citra dan deteksi wajah).
2. **wajah = cv.CascadeClassifier('haarcascade\_frontalface\_default.xml')**: Membaca file XML yang berisi model klasifikasi Cascade untuk deteksi wajah. File ini berisi pola yang digunakan untuk mengenali wajah manusia.
3. **img = cv.imread('Rizki.jpg')**: Membaca gambar dengan nama file "Rizki.jpg" dan menyimpannya dalam variabel img.
4. **img\_gray = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)**: Mengubah citra berwarna (BGR) menjadi citra grayscale. Deteksi wajah biasanya dilakukan pada citra grayscale karena lebih efisien.
5. **deteksi\_wajah = wajah.detectMultiScale(img\_gray, 1.1, 5)**: Mendeteksi wajah dalam citra grayscale menggunakan metode deteksi Haar Cascade. Fungsi **detectMultiScale** mengembalikan daftar koordinat (x, y, lebar, tinggi) dari setiap wajah yang terdeteksi.
6. **font = cv.FONT\_HERSHEY\_SIMPLEX** dan **jumlah = 0**: Mendefinisikan jenis font yang akan digunakan untuk menampilkan teks dan menginisialisasi variabel **jumlah** sebagai counter untuk menghitung jumlah wajah yang terdeteksi.
7. **for (x, y, w, h) in deteksi\_wajah:**: Melakukan iterasi pada setiap wajah yang terdeteksi.
8. Di dalam loop, terdapat beberapa operasi:
   * **jumlah = jumlah + 1**: Meningkatkan counter jumlah wajah.
   * **cv.putText(img, "Wajah", (x, y-10), font, 0.75, (0, 0, 255), 2, cv.LINE\_AA)**: Menampilkan teks "Wajah" di atas setiap kotak deteksi wajah.
   * **cv.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)**: Menggambar kotak persegi di sekitar setiap wajah yang terdeteksi.
   * **roi\_gray = img\_gray[y:y+h, x:x+w]** dan **roi\_color = img[y:y+h, x:x+w]**: Mendapatkan citra grayscale dan citra berwarna dari setiap wajah yang terdeteksi.
9. **cv.putText(img, "Jumlah wajah ada: " + str(jumlah) + " buah", (10, 30), font, 1, (0, 0, 0), 2, cv.LINE\_AA)**: Menampilkan jumlah wajah yang terdeteksi di pojok kiri atas gambar.
10. **cv.imshow('img', img)**: Menampilkan gambar yang telah diberi kotak deteksi wajah dan teks jumlah wajah.
11. **cv.waitKey(0)**: Menunggu hingga tombol apa pun ditekan pada jendela tampilan gambar.
12. **cv.destroyAllWindows()**: Menutup semua jendela tampilan gambar setelah tombol ditekan.

Tujuan dari proyek codingan ini adalah untuk mendeteksi wajah manusia dalam sebuah gambar dan menandai wajah tersebut dengan kotak persegi serta menampilkan jumlah wajah yang terdeteksi pada gambar.