```
#*************************************************************************************************#
#********                    PROGRAM PREDIKSI GEOMETRI MOLEKUL VERSI VESPR (GUI)                  *********#
#********                                        Dibuat Oleh :                                    *********#
#********                                       Djukarna                                          *********#
#********                                     Tanggal : 15 OKTOBER 2025                           *********#
#********                                   Fakultas Keguruan & Pendidikan                        *********#
#********                                   Universitas Katolik Parahyangan                       *********#
#********                                      Bandung - Jawa Barat                               *********#
#*************************************************************************************************#

import tkinter as tk
from tkinter import *
from tkinter import ttk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
import numpy as np

#---------------------------------#
# Konfigurasi Jendela Utama
#---------------------------------#
root = tk.Tk()
root.geometry("1200x650")
root.title("Vesper")
root.iconbitmap("icon.ico")

#---------------------------------#
# Frame Utama
#---------------------------------#
frame_kontrol = tk.Frame(root)
frame_kontrol.pack(side="left", fill="y", padx=10, pady=10)

frame_model = tk.Frame(root, bg="#ADD8E6")
frame_model.pack(side="right", fill="both", expand=True, padx=10, pady=10)

#---------------------------------#
# Fungsi-fungsi Program
#---------------------------------#

def exitProgram():
    print("Keluar dari Program")
    root.destroy()

def tampilkan_tetrahedral():                                    # jadikan bentuk OOP dan update untuk jenis molekul lainnya
    # Hapus canvas lama jika ada
    for widget in frame_model.winfo_children():
        widget.destroy()

    # Buat figure dan axis 3D
    fig = plt.figure(figsize=(4, 4), facecolor="#ADD8E6")
    ax = fig.add_subplot(111, projection='3d')
    ax.set_facecolor("#ADD8E6")

    # Koordinat atom pusat (C)
    pusat = np.array([0, 0, 0])
    ax.scatter(*pusat, color='black', s=200, label='C')

    # Koordinat atom H (tetrahedral)
    bond_length = 1.0
    arah = np.array([
        [1, 1, 1],
        [-1, -1, 1],
        [-1, 1, -1],
        [1, -1, -1]
    ])
```

```
64        arah = arah / np.linalg.norm(arah, axis=1)[:, None] * bond_length
65
66        # Gambar atom H dan ikatannya
67        for vektor in arah:
68            ax.scatter(*(pusat + vektor), color='white', edgecolor='black', s=100)
69            ax.plot([pusat[0], pusat[0]+vektor[0]],
70                    [pusat[1], pusat[1]+vektor[1]],
71                    [pusat[2], pusat[2]+vektor[2]],
72                    color='gray', linewidth=2)
73
74        ax.set_xlim(-1.5, 1.5)
75        ax.set_ylim(-1.5, 1.5)
76        ax.set_zlim(-1.5, 1.5)
77        ax.axis('off')
78        ax.set_title("Model Tetrahedral (CH□)", color='navy')
79
80        # Tampilkan ke Tkinter
81        canvas = FigureCanvasTkAgg(fig, master=frame_model)
82        canvas.draw()
83        canvas.get_tk_widget().pack(expand=True, fill='both')
84
85
86    def hitung_geometri():
87        # Ambil input pengguna
88        pusat = atom_pusat.get().strip().capitalize()         #atom dari input atom pusat
89        terikat = atom_terikat.get().strip().capitalize()     #atom terikat dari input
          atom terikat
90        jumlah_atom = n_ikatan.get()                          #jumlah atom terikat
91
92        # Kamus valensi sederhana
93        valensi = {
94        "H"  : 1, "He": 2,
95        "Li" : 1, "Be": 2,  "B": 3,  "C": 4,  "N": 5,  "O": 6, "F" : 7, "Ne": 8,
96        "Na" : 1, "Mg": 2, "Al": 3, "Si": 4,  "P": 5,  "S": 6, "Cl": 7, "Ar": 8,
97        "K"  : 1, "Ca": 2, "Ga": 3, "Ge": 4, "As": 5, "Se": 6, "Br": 7, "Kr": 8,
98        "Rb" : 1, "Sr": 2, "In": 3, "Sn": 4, "Sb": 5, "Te": 6, "I" : 7, "Xe": 8,
99        "Cs" : 1, "Ba": 2, "Tl": 3, "Pb": 4, "Bi": 5, "Po": 6, "At": 7, "Rn": 8,
100       }
101
102       elektronegatif = {
103       "H"  : 2.1, "He": 0,
104       "Li" : 1.0, "Be": 1.5,  "B": 2.0,  "C": 2.5,  "N": 3.0,  "O": 3.5, "F" : 4.0, "Ne"
          : 0,
105       "Na" : 0.9, "Mg": 1.2, "Al": 1.5, "Si": 1.8,  "P": 2.1,  "S": 2.5, "Cl": 3.0, "Ar"
          : 0,
106       "K"  : 0.8, "Ca": 1.0, "Ga": 1.6, "Ge": 1.8, "As": 2.0, "Se": 2.4, "Br": 2.8, "Kr"
          : 3,     # gas mulai Kripton Kasus khusus, hehehee belum sepenuhya mulia jika
          ketemu Si Cantik F
107       "Rb" : 0.8, "Sr": 1.0, "In": 1.7, "Sn": 1.8, "Sb": 1.9, "Te": 2.1, "I" : 2.5, "Xe"
          : 2.6,   # gas mulai Xenon Kasus khusus, hehehee belum sepenuhya mulia jika
          ketemu Si Cantik F
108       "Cs" : 0.7, "Ba": 0.9, "Tl": 1.8, "Pb": 1.8, "Bi": 1.9, "Po": 2.0, "At": 2.2, "Rn"
          : 2.2,   # gas mulai Radon Kasus khusus, hehehee belum sepenuhya mulia jika
          ketemu Si Cantik F
109       }
110
111       bentuk_vsepr = {
112       (2, 0): ("AX2E0", "Linear", 180),
113       (2, 1): ("AX2E", "Linear", 180),
114       (3, 0): ("AX3E0", "Trigonal Planar", 120),
115       (4, 0): ("AX4E0", "Tetrahedral", 109.5),
116       (3, 1): ("AX3E1", "Trigonal Pyramidal", 107),
117       (2, 2): ("AX2E2", "Bent (V-shaped)", 109.5),
118       (6, 0): ("AX6E0", "Oktahedral", 90),
119       (5, 0): ("AX5E0", "Triigonal bipiramida", 120),
120       (4, 1): ("AX4E", "Tetrahedral Terdistorsi", 90),
121       (3, 2): ("AX3E2", "T-Planar", 120),
122       (2, 3): ("AX2E3", "Linear II", 180),
123       (6, 0): ("AX6", "Oktahedral", 90),
124       (5, 1): ("AX5E", "Tetragonal Bipiramida", 90),
125       (4, 2): ("AX4E2", "Square Planar", 90),
126       (2, 4): ("AX2E4", "Linear", 180),
```

```
        }

        elektron_valensi_pusat = valensi.get(pusat, 0)
        elektron_valensi_terikat = valensi.get(terikat, 0)
        total_elektron_valensi = (jumlah_atom * elektron_valensi_terikat) +
        elektron_valensi_pusat

        N_pusat = elektronegatif.get(pusat, 0)
        N_ikat = elektronegatif.get(terikat, 0)
        delta_N = abs(N_pusat - N_ikat)

        print("elektron valensi atom pusat: ", elektron_valensi_pusat)
        print("elektron valensi atom terikat: ", elektron_valensi_terikat)
        print("Keelektronagatifan atom pusat: ", N_pusat)
        print("Keelektronagatifan atom terikat: ", N_ikat)
        print("Keelektronagatifan senyawa: ",delta_N)

        if (delta_N < 1.7):
            total_EV = (jumlah_atom * terikat)+ pusat
            PEI = jumlah_atom
            elektron_valensi_tersisa = (total_elektron_valensi - PEI*2)
            PEB = int((elektron_valensi_pusat - jumlah_atom)/2)
            #PEB = (elektron_valensi_pusat - jumlah_atom)/2
            PEI_var.set(PEI)
            PEB_var.set(PEB)
            key = (PEI, PEB)
            if key in bentuk_vsepr:
                simbol, bentuk, sudut = bentuk_vsepr[key]
                rumus_molekul.set(bentuk)
                rumus_geometri.set(simbol)
                sudut_geometri.set(sudut)
        elif(delta_N > 1.7):
            rumus_molekul.set("Ikatan Ionik")
            rumus_geometri.set("Ikatan Ionik  ")
            sudut_geometri.set("Ikatan Ionik ")


    #----------------------------------#
    # GUI Kontrol
    #----------------------------------#

    label1 = Label(frame_kontrol, text="PREDIKSI GEOMETRI MOLEKUL VSEPR", font=("Verdana",
     16))
    label1.grid(row=0, column=0, columnspan=3, pady=10)

    # Logo UNPAR
    img = PhotoImage(file="logo_UNPAR.png")
    img1 = img.subsample(2,2)
    Label(frame_kontrol, image=img1).grid(row=0, column=3, padx=10)

    # Input data
    Label(frame_kontrol, text="Simbol Atom Pusat:", font=("Verdana", 12)).grid(row=2,
    column=0, sticky="w")
    atom_pusat = tk.StringVar()
    Entry(frame_kontrol, textvariable=atom_pusat, width=6, font=('calibre', 12)).grid(row=
    2, column=1)

    Label(frame_kontrol, text="Simbol Atom Terikat:", font=("Verdana", 12)).grid(row=3,
    column=0, sticky="w")
    atom_terikat = tk.StringVar()
    Entry(frame_kontrol, textvariable=atom_terikat, width=6, font=('calibre', 12)).grid(
    row=3, column=1)

    Label(frame_kontrol, text="Jumlah Ikatan:", font=("Verdana", 12)).grid(row=4, column=0
    , sticky="w")
    n_ikatan = tk.IntVar()
    Entry(frame_kontrol, textvariable=n_ikatan, width=6, font=('calibre', 12)).grid(row=4,
     column=1)

    # Tombol Hitung
    Button(frame_kontrol, text="Hitung!", command=hitung_geometri, font=("Arial", 12,
    "bold"), bg="lightgray").grid(row=5, column=1, pady=10)
```

```
190
191   # Output hasil
192   Label(frame_kontrol, text="Pasangan Elektron Ikatan (PEI):", font=("Verdana", 12)).
      grid(row=6, column=0, sticky="w")
193   PEI_var = tk.IntVar()
194   Entry(frame_kontrol, textvariable = PEI_var, width=8).grid(row=6, column=1)
195
196   Label(frame_kontrol, text="Pasangan Elektron Bebas (PEB):", font=("Verdana", 12)).grid
      (row=7, column=0, sticky="w")
197   PEB_var = tk.IntVar()
198   Entry(frame_kontrol, textvariable = PEB_var, width=8).grid(row=7, column=1)
199
200   Label(frame_kontrol, text="Geometri Molekul:", font=("Verdana", 12)).grid(row=8,
      column=0, sticky="w")
201   rumus_molekul = tk.StringVar()
202   Entry(frame_kontrol, textvariable = rumus_molekul, width=20).grid(row=8, column=1)
203
204   Label(frame_kontrol, text="Rumus Geometri Molekul:", font=("Verdana", 12)).grid(row=9,
       column=0, sticky="w")
205   rumus_geometri = tk.StringVar()
206   Entry(frame_kontrol, textvariable = rumus_geometri, width=20).grid(row=9, column=1)
207
208   Label(frame_kontrol, text="Sudut Geometri Molekul:", font=("Verdana", 12)).grid(row=10
      , column=0, sticky="w")
209   sudut_geometri = tk.StringVar()
210   Entry(frame_kontrol, textvariable = sudut_geometri, width=20).grid(row=10, column=1)
211
212   # Tombol Modelkan
213   Button(frame_kontrol, text="Modelkan!", command=tampilkan_tetrahedral, font=("Arial",
      12, "bold"), bg="lightgray").grid(row=12, column=1, pady=10)  #Jika claas model sudah
      banyak ubah untuk menampilkan akses ke setiap kelas
214
                                                                    #menurut jenis geometri
      molekulnya, buat class sendiri untuk programnya
215
216   # Tombol Exit
217   Button(frame_kontrol, text="Keluar", command=exitProgram, font=("Arial", 12, "bold"),
      bg="#ff8080").grid(row=13, column=1, pady=10)
218
219   root.mainloop()
220
```