Libraries and upload files:

from flask import Flask, render\_template, request, redirect, url\_for, flash

from googlesearch import search

import PyPDF2

import os

from werkzeug.utils import secure\_filename

path = os.getcwd()

UPLOAD\_FOLDER = os.path.join(path, 'uploads')

ALLOWED\_EXTENSIONS = {'pdf'}

app = Flask(\_\_name\_\_)

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

app.add\_url\_rule(

    "/uploads/<name>", endpoint="download\_file", build\_only=True

)

app.config['MAX\_CONTENT\_LENGTH'] = 16 \* 1000 \* 1000

app.config['SECRET\_KEY'] = 'super secret key'

def allowed\_file(filename):

    return '.' in filename and \

           filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

index file:

@app.route('/', methods=['GET', 'POST'])

Def index():...

The main plagiarism function:

@app.route('/plagiarism/<name>', methods=['GET', 'POST'])

def plagiarism(name):

domain part for google searches:

    domain = "co.id"

variables to store output and plagiarism detected files:

    link\_output = []

    hasil\_plagiarism = []

    hasil\_link = []

    hasil\_persen = 0

    inputan\_mentah = ""

    inputan = []

    filename = ""

    text = ""

    hasil\_plagiarism\_final = []

    hasil\_link\_final = []

blocked sites from being considered in the plagiarism detection process:

    link\_blocked = ["id.linkedin.com", "linkedin.com", "youtube.com", "instagram.com", "facebook.com", "tokopedia.com",

                    "twitter.com", "reddit.com", "bukalapak.com", "shopee.com", "blibli.com"]

Checking if data submitted or uploaded or not:

        if request.form['text'] != '' and request.files['file'].filename == '':