PANDUAN PENGGUNAAN PIPELINE OCR

UNTUK DIGITALISASI ARSIP TANAH

Disusun oleh:

KKN-PPM UGM Periode 4 2024/2025

Kelurahan Randuacir, Kecamatan Argomulyo

Kota Salatiga, Provinsi Jawa Tengah

Tahun 2025

Pendahuluan

Proyek KKN PPM UGM 2024/2025 Periode 4 bertujuan untuk mendukung pengarsipan digital dokumen kepemilikan tanah di Kelurahan Randuacir dan Cebongan, Salatiga. Sebagai salah satu bagian dari tema besar "Pemberdayaan Masyarakat melalui Geoinformasi dan Literasi untuk Meningkatkan Kesejahteraan," kegiatan ini difokuskan pada pembuatan alat pencarian dokumen berbasis *Optical Character Recognition* (OCR). Dengan memanfaatkan teknologi ini, dokumen yang sebelumnya tersedia dalam bentuk fisik dapat diakses secara digital dan diolah lebih lanjut untuk berbagai kebutuhan.

Kelurahan Randuacir menyimpan ratusan dokumen kepemilikan tanah yang hingga saat ini dikelola dalam bentuk fisik. Tantangan yang muncul dari metode pengelolaan ini meliputi risiko kerusakan dokumen, sulitnya pencarian informasi tertentu, serta kurangnya efisiensi dalam pengelolaan data. Dengan memanfaatkan teknologi digital, masalah ini dapat diatasi melalui sistem yang memungkinkan digitalisasi, penyimpanan terstruktur, dan pencarian dokumen secara efisien.

Program kerja pertama dari kegiatan ini, berjudul “Alat Pencarian Dokumen Kepemilikan Tanah”, telah dilakukan uji coba pipeline pengolahan data dokumen tanah menggunakan OCR dengan satu sampel gambar, meskipun hasilnya masih memerlukan penyempurnaan. Dalam proker kedua, fokus diarahkan pada meningkatkan skala proyek agar mencakup seluruh file yang terdiri dari ratusan entri dokumen kepemilikan tanah.

Salah satu tantangan utama dalam pengarsipan dokumen di Kelurahan Randuacir adalah risiko kerusakan dokumen akibat faktor usia, kelembapan, atau kondisi lingkungan yang kurang mendukung. Dokumen-dokumen ini sering kali menjadi dasar dalam penyelesaian sengketa tanah atau urusan warisan, sehingga kondisi fisiknya yang rapuh dan rentan menimbulkan risiko kehilangan informasi penting. Selain itu, ketiadaan sistem digitalisasi yang memadai mengakibatkan staf kelurahan harus mengandalkan metode pencatatan manual, yang dapat memakan waktu dan rentan terhadap kesalahan. Volume dokumen yang besar semakin memperumit proses pengelolaan, sehingga diperlukan solusi teknologi yang mampu meningkatkan efisiensi, akurasi, dan perlindungan data secara keseluruhan.

Pipeline OCR yang dikembangkan dalam proyek ini bertujuan untuk mendigitalisasi dokumen fisik menjadi format digital yang terstruktur. Proses ini mencakup serangkaian langkah yang dimulai dengan persiapan lingkungan pengembangan, termasuk pengunduhan dan pembaruan library yang diperlukan setelah menginstal dependensi eksternal seperti Tesseract dan Pytesseract. Langkah-langkah tersebut dijelaskan secara lebih rinci dalam bagian berikutnya. Setelah semua library diimpor, pipeline dimulai dengan pembuatan subdirektori untuk menyimpan hasil pengolahan gambar. Subdirektori ini berfungsi sebagai tempat penyimpanan file PDF asli yang dikonversi menjadi gambar berformat PNG sebagai langkah awal digitalisasi.

Tahapan berikutnya melibatkan preprocessing gambar melalui beberapa langkah berikut:

1. Langkah 1.1: Konversi gambar ke grayscale.
2. Langkah 1.2: Peningkatan kontras menggunakan metode CLAHE.
3. Langkah 1.3: Penghapusan noise menggunakan median blurring.
4. Langkah 1.4: Normalisasi dimensi gambar.
5. Langkah 2.1: Deteksi struktur tabel.
6. Langkah 2.2: Penghapusan struktur tabel untuk mengekstraksi teks.
7. Langkah 3.1: Penerapan OCR untuk mengekstraksi teks dari gambar.
8. Langkah 3.2: Pembersihan hasil OCR untuk menghasilkan teks yang lebih terstruktur.
9. Langkah 3.3: Penyimpanan hasil OCR ke dalam file CSV dengan struktur yang sesuai.

Hasil dari setiap langkah preprocessing disimpan dalam sub-subdirektori terpisah untuk mempermudah evaluasi dan pengelolaan data. Setiap halaman PDF menghasilkan hingga 7 file PNG (sumber asli serta hasil konversi PDF ke PNG, dan dari Langkah 1.1 hingga Langkah 2.2), satu file .txt dari Langkah 3.1 dan Langkah 3.2, serta satu file CSV untuk Langkah 3.3. Total data yang dihasilkan mencapai hingga 6 GB, berdasarkan pengujian sebesar 5,68 GB, yang sudah termasuk seluruh file PDF dan direktori yang digunakan. Ukuran tersebut belum mencakup alat seperti VSCode, Python, library terkait, maupun Tesseract.

Pada komputer yang digunakan selama pengujian, keseluruhan pipeline memerlukan waktu sekitar 30 menit untuk diproses, yang dapat dijadikan acuan waktu eksekusi pada

perangkat dengan spesifikasi serupa. Spesifikasi komputer yang digunakan adalah sebagai berikut:

1. CPU: Intel i7-10750H @ 2.60GHz (12 CPU threads)
2. RAM: 16 GB DDR4 dual-channel
3. GPU: Nvidia GTX 1650 Ti
4. Penyimpanan: SSD 500 GB

Untuk mempercepat runtime pipeline, disarankan menggunakan perangkat dengan spesifikasi lebih tinggi, seperti prosesor dengan jumlah core lebih banyak, GPU yang lebih mumpuni untuk akselerasi komputasi paralel, RAM yang lebih besar untuk mendukung pemrosesan file dalam jumlah besar, serta SSD berkecepatan tinggi dengan ruang penyimpanan yang cukup.

Namun demikian, pipeline ini masih memiliki sejumlah keterbatasan yang perlu diperhatikan. Tantangan terbesar adalah kualitas dokumen yang kurang ideal, seperti tulisan tangan yang tipis dan sulit terbaca, format penulisan yang tidak teratur, serta kerusakan fisik seperti sobekan, lipatan, dan noda. Faktor-faktor tersebut memengaruhi akurasi hasil OCR. Oleh karena itu, pipeline ini dirancang sebagai dasar untuk pengembangan lebih lanjut, khususnya dalam hal penyempurnaan langkah preprocessing gambar dan optimasi parameter OCR, guna mencapai hasil yang lebih baik di masa mendatang.

Petunjuk penggunaan program ini akan dirinci dalam manual ini dan tutorial video. Panduan ini mencakup semua tahapan proses, mulai dari instalasi perangkat lunak, konfigurasi pipeline, hingga pengelolaan hasil digitalisasi. Dengan adanya dokumentasi ini, diharapkan staf kantor lurah Randuacir dapat melanjutkan dan mengelola proses digitalisasi.

Gambaran Umum Alur Kerja

Workflow ini dirancang untuk mengubah dokumen fisik menjadi format digital yang terstruktur dengan menggunakan teknologi OCR (Optical Character Recognition). Secara umum, alur kerjanya mencakup langkah-langkah berikut:

1. Konversi Dokumen PDF ke Gambar: Dokumen dalam format PDF diubah menjadi gambar untuk mempermudah pemrosesan lebih lanjut.

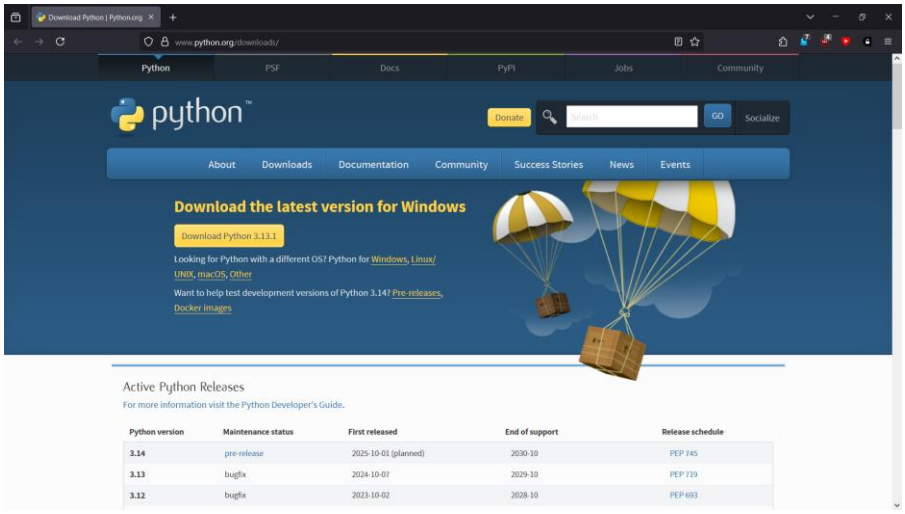
2. Preprocessing Gambar: Gambar yang dihasilkan melalui tahapan seperti konversi ke grayscale, peningkatan kontras, penghapusan noise, dan normalisasi dimensi.
3. Ekstraksi Teks menggunakan OCR: Gambar yang telah diproses digunakan untuk mengenali dan mengekstraksi teks menggunakan teknologi OCR.
4. Penyimpanan Hasil dalam CSV: Teks hasil OCR disimpan dalam file CSV yang memudahkan pencarian dan pengelolaan data.

Instalasi dan Persiapan

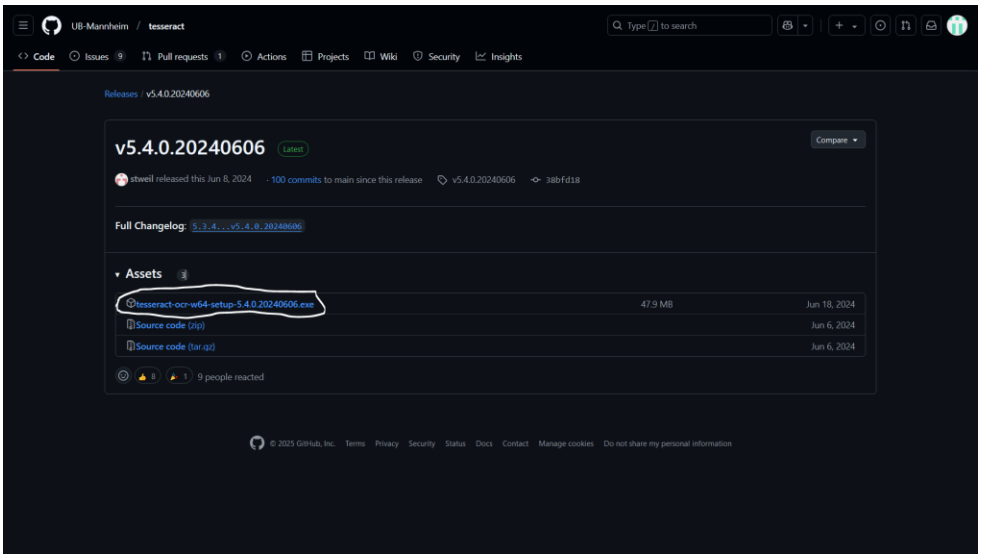
- Sistem Operasi

Program ini telah diuji pada Windows 10. Sistem operasi lain mungkin memerlukan konfigurasi tambahan.

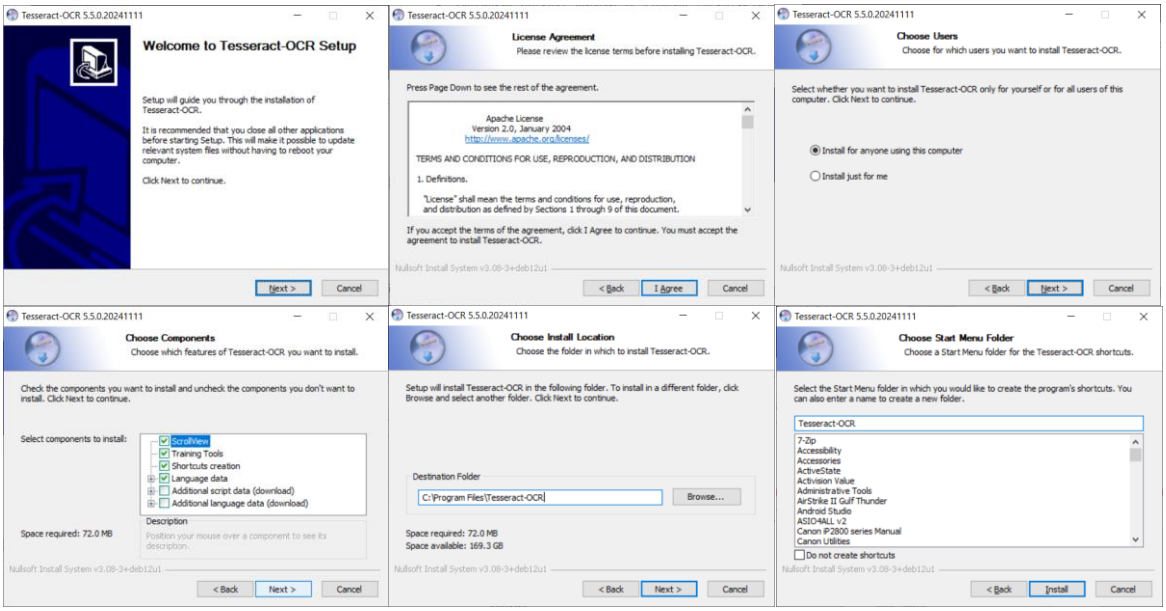
- Perangkat Lunak dan Alat
    - Python 3.10 atau versi lebih baru
        - Unduh Python dari situs resmi Python dan ikuti panduan instalasi



    - Tesseract OCR versi 5.4.0.20240606 atau lebih baru
        - Instal Tesseract OCR dari tautan GitHub ini:
            - Download file executable yang ditunjukkan gambar:



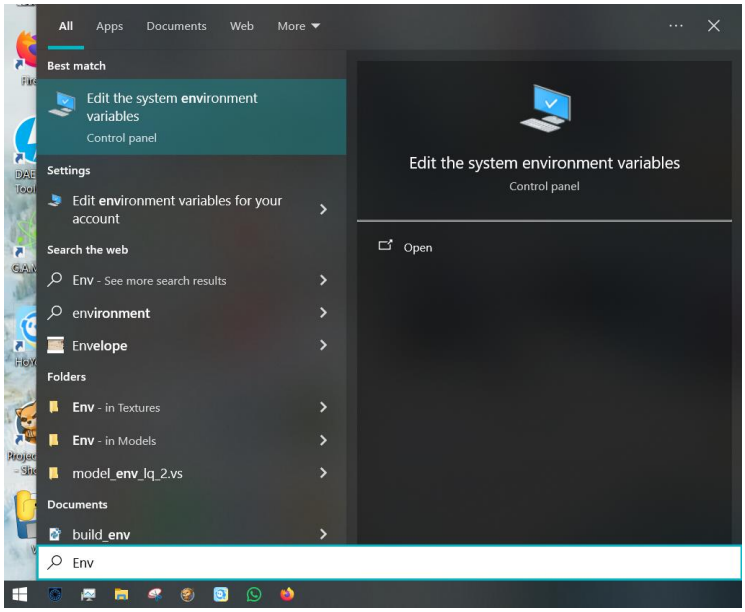
- Setelah selesai download, run file tersebut dan ikuti panduan instalasi sesuai gambar, kiri ke kanan:



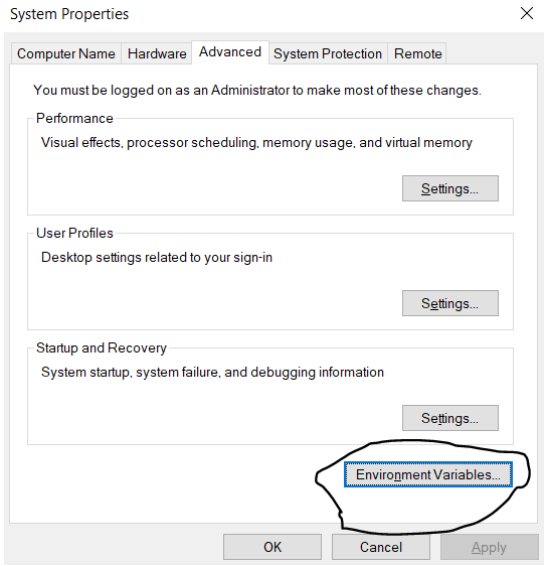
- Tekan tombol *Install* dan tunggu sampai instalasi selesai.

▪ Tambahkan direktori instalasi (C:\Program Files\Tesseract-OCR) ke dalam PATH sistem.

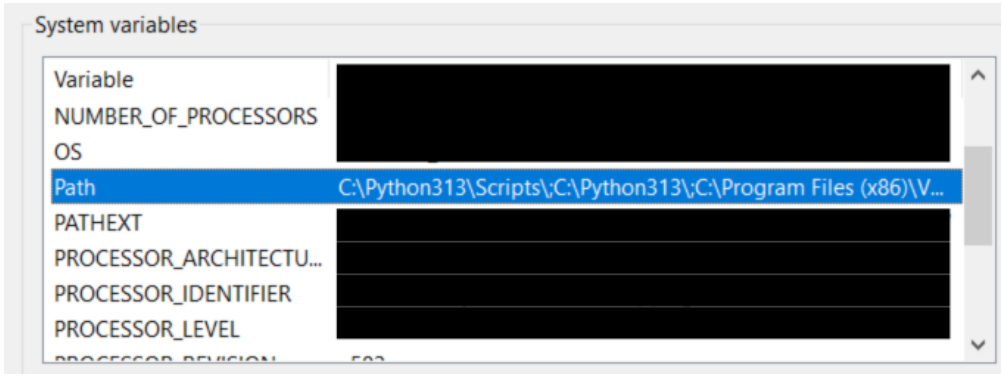
- Cari *Edit the system environment variables* pada menu Windows sesuai gambar



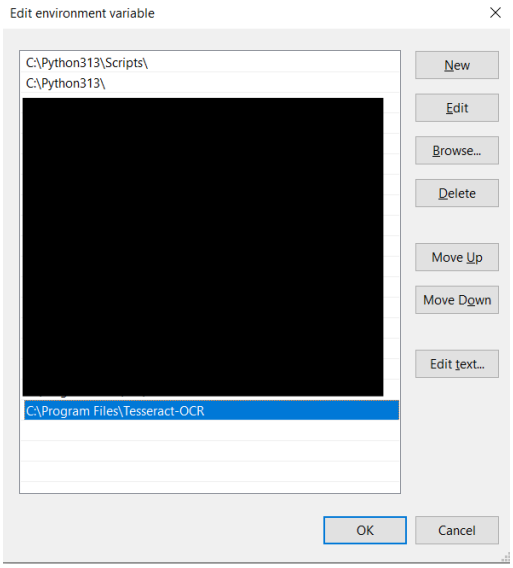
- Pencet tombol *Environment Variables*



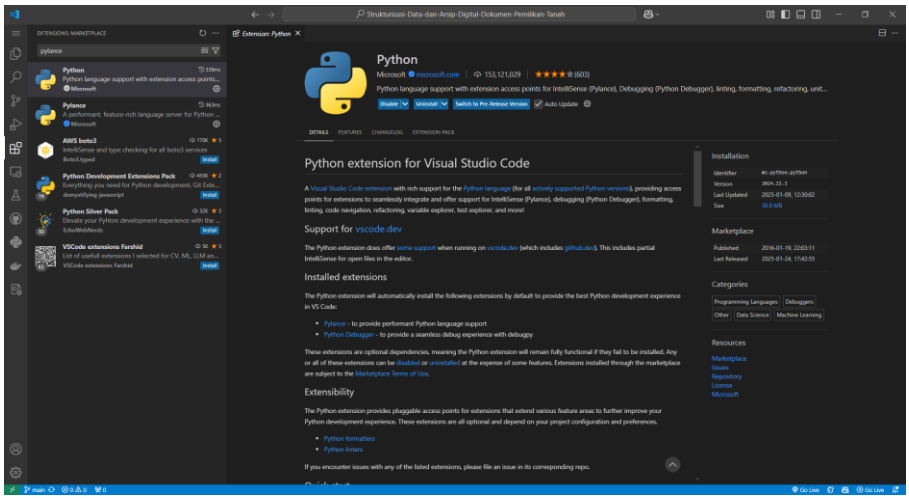
- Pada kolom *System Variables*, pencet *PATH* sesuai gambar



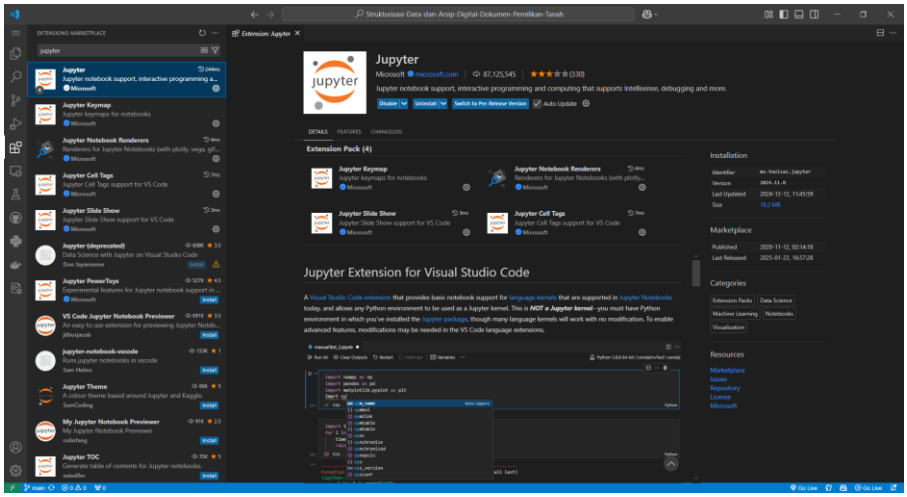
- Pencet tombol *New*, dan ketik C:\Program Files\Tesseract-OCR. Pencet *OK*, pencet *OK* pada menu *Environment* Variables, dan *Apply* pada menu *System Properties*.



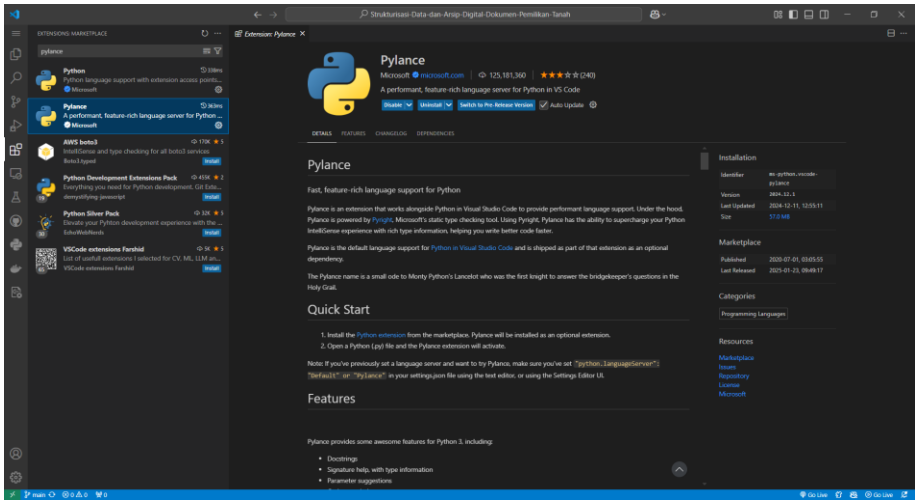
- Visual Studio Code (VSCode)
  - Unduh VSCode dan pastikan ekstensi berikut terinstal:
    - Python (Microsoft)



    - Jupyter

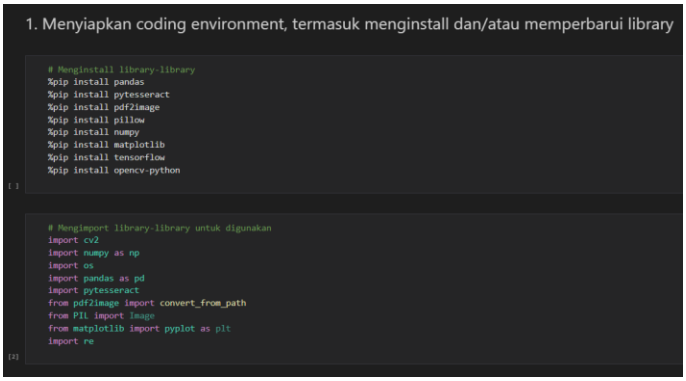


- Pylance



- Library Python

Library Python yang diperlukan akan otomatis diinstal melalui kode di sel pertama notebook. Anda hanya perlu menjalankan notebook untuk mengatur semuanya. Pastikan Anda memiliki koneksi internet yang stabil saat pertama kali menjalankan pipeline.



- File dan Folder Berkas Letter C

Untuk menjalankan pipeline OCR ini, diperlukan file dan folder dengan struktur dan nama yang spesifik. Folder utama di repositori ini adalah BERKAS LETTER C, dengan struktur sebagai berikut:

1. File PDF Letter C

File PDF ini adalah dokumen utama yang akan diproses. **Pastikan nama file sesuai dengan daftar berikut secara verbatim (case-sensitive)**:

a. berkas 0-100.pdf

b. berkas 100-200.pdf

c. berkas 200-300.pdf
d. berkas 300-400.pdf
e. berkas 400-500.pdf
f. berkas 500-520 EXAMPLE 2.pdf
g. berkas 500-543 EXAMPLE.pdf
h. berkas 500-600.pdf
i. berkas 600-700.pdf
j. berkas 700-800.pdf
k. berkas 800-900.pdf
l. berkas 900-968.pdf
m. berkas sisa sobekan 400-500.pdf

Nama file yang berbeda atau tidak sesuai dengan daftar di atas akan menyebabkan error saat menjalankan pipeline.

2. File .sys_id.dat

File ini adalah file sistem tambahan yang tidak mempengaruhi pipeline, tetapi disarankan untuk tidak dihapus.

- Spesifikasi Komputer yang Direkomendasikan

Untuk memastikan pipeline OCR berjalan dengan lancar dan waktu pemrosesan tetap efisien, berikut adalah spesifikasi komputer yang direkomendasikan:

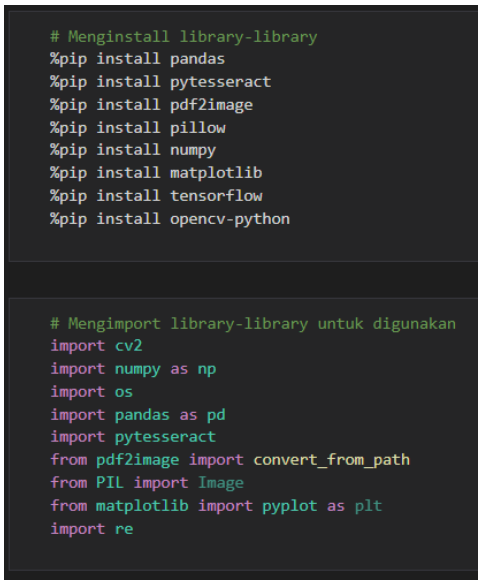
- Prosesor: Intel Core i5 generasi ke-8 atau setara
- RAM: 8 GB
- GPU: Tidak diperlukan (pipeline dapat dijalankan tanpa GPU)
- Penyimpanan: SSD dengan kapasitas kosong minimal 10 GB
- Sistem Operasi: Windows 10 atau keatas

Panduan Penggunaan Langkah Demi Langkah

Sebelum memulai pipeline, pastikan semua komponen berikut telah dipersiapkan:

- Instalasi Python dan Library:
  - Instal Python (versi 3.10 atau lebih baru).

- Jalankan cell pertama dan kedua di file ocr.ipynb untuk menginstal library: pandas, pytesseract, pdf2image, pillow, numpy, matplotlib, tensorflow, dan opencv-python, kemudian di import pada cell berikutnya.



```
# Menginstall library-library
%pip install pandas
%pip install pytesseract
%pip install pdf2image
%pip install pillow
%pip install numpy
%pip install matplotlib
%pip install tensorflow
%pip install opencv-python
```

```
# Mengimport library-library untuk digunakan
import cv2
import numpy as np
import os
import pandas as pd
import pytesseract
from pdf2image import convert_from_path
from PIL import Image
from matplotlib import pyplot as plt
import re
```

- Instalasi Tesseract OCR:
  - Pastikan Tesseract telah di install dan pathnya berada di *Environment Variables*.
- Penyiapan Folder dan File:
  - Pastikan semua file PDF berada di folder "BERKAS LETTER C" dengan nama sesuai (misalnya: berkas 0-100.pdf, berkas 100-200.pdf, dll.).
  - Jangan mengubah struktur folder dalam repositori.
- Cek Ruang Penyimpanan:
  - Sediakan ruang minimal 6 GB untuk hasil pipeline.

Mengoperasikan Program

- Cara Mengoperasikan Program
  - Pastikan semua file, folder, dan dependensi telah diatur sesuai dengan instruksi pada bagian sebelumnya.
  - Buka file ocr.ipynb di Visual Studio Code atau Jupyter Notebook.
  - Klik tombol Run All di bagian atas untuk menjalankan seluruh pipeline program.
    - Program akan memproses semua file PDF secara otomatis melalui pipeline OCR.

        - Hasil akan disimpan dalam subfolder yang sesuai untuk setiap langkah pemrosesan.

- Hasil yang Diharapkan

    Setelah menjalankan program, Anda akan mendapatkan output berikut:

    - Gambar pada Setiap Tahap Preprocessing:
        - Gambar-gambar yang menunjukkan hasil setiap langkah preprocessing, seperti grayscale, peningkatan kontras, dan sebagainya.
        - Gambar yang memuat struktur tabel dan versi tanpa tabel untuk setiap halaman PDF.
    - File Teks Hasil OCR:
        - File teks mentah yang diekstraksi dari setiap halaman dokumen.
        - File teks mentah yang dibersihkan paska ekstrasi.
        - File teks ini mungkin mengandung kesalahan akibat kualitas dokumen asli.
    - File CSV Akhir:
        - File CSV gabungan yang berisi teks hasil OCR dari semua halaman, disusun berdasarkan nama dokumen.

- Kualitas Hasil dan Harapan

    Meskipun pipeline dirancang untuk menangani berbagai jenis dokumen, hasil saat ini memiliki beberapa kekurangan, antara lain:

    - Kesalahan Ekstraksi Teks: Tulisan tangan yang tipis, posisi teks yang tidak rapi, atau format tabel yang tidak teratur dapat mengurangi akurasi OCR.
    - Kebutuhan Perbaikan Manual: Hasil dalam file CSV mungkin memerlukan revisi manual untuk meningkatkan kualitas.
    - Keterbatasan Dokumen Sumber: Dokumen yang robek, pudar, atau terkena lipatan dapat memengaruhi hasil akhir.

- Catatan Penting

    Proyek ini dirancang sebagai fondasi untuk pengembangan lebih lanjut. Pengguna diharapkan memberikan masukan atau mencoba parameter baru pada pipeline untuk meningkatkan kualitas hasil di masa depan.

FAQ dan Pemecahan Masalah

- Apakah saya harus menginstal semua dependensi secara manual?
    - Tidak. Semua dependensi Python yang diperlukan akan diinstal otomatis melalui cell pertama di file ocr.ipynb. Namun, Anda harus memastikan Tesseract OCR terinstal secara manual sesuai petunjuk di bagian sebelumnya.
- Berapa lama waktu yang dibutuhkan untuk memproses semua dokumen?
    - Waktu pemrosesan tergantung pada spesifikasi komputer Anda. Pada komputer dengan spesifikasi yang digunakan dalam testing (Intel i7-10750H, 16 GB RAM, SSD), seluruh pipeline memakan waktu sekitar 30 menit untuk memproses semua dokumen
- Mengapa hasil OCR tidak akurat?
    - Hasil OCR dapat terpengaruh oleh kualitas dokumen asli, seperti tulisan tangan yang tidak jelas, tabel yang tidak terdeteksi dengan sempurna, atau dokumen yang rusak. Perlu dilakukan penyesuaian pada parameter preprocessing atau OCR untuk meningkatkan akurasi.
- Apakah saya bisa memproses dokumen lain di luar file yang disediakan?
    - Bisa. Anda dapat mengganti file PDF sumber dengan dokumen lain, asalkan mengikuti struktur folder yang sama, dan mengedit kode untuk menggunakan nama folder/file baru.
- **Masalah**: Tesseract tidak terdeteksi oleh program.
    - **Solusi**: Pastikan Tesseract terinstal dengan benar dan lokasinya ditambahkan ke variabel PATH pada sistem Anda. Periksa petunjuk di bagian Instalasi dan Persiapan.
- **Masalah**: Program berhenti atau error saat dijalankan.
    - **Solusi**: Periksa apakah semua dependensi telah terinstal dengan benar. Jika error tetap terjadi, coba jalankan cell secara individual untuk mengidentifikasi tahap yang menyebabkan masalah.
- **Masalah**: Hasil preprocessing gambar tidak sesuai harapan.

    - **Solusi**: Cobalah untuk menyesuaikan parameter preprocessing, seperti nilai kontras pada langkah CLAHE atau ukuran kernel pada median blurring.
- **Masalah**: File CSV akhir tidak dibuat.
    - **Solusi**: Pastikan semua langkah sebelumnya telah berhasil dijalankan. Periksa juga apakah folder tujuan untuk menyimpan CSV memiliki izin akses.
- **Masalah**: Program berjalan sangat lambat.
    - **Solusi**: Pastikan komputer Anda memiliki spesifikasi yang memadai. Jika menggunakan komputer dengan spesifikasi rendah, pertimbangkan untuk memproses dokumen dalam jumlah yang lebih kecil.
- **Masalah**: Gambar tidak muncul di folder subdirektori.
    - **Solusi**: Periksa apakah folder dan nama file sudah diatur sesuai dengan struktur yang dijelaskan dalam manual ini.

Glossarium

- **A**
    - *Adaptive Histogram Equalization* (AHE): Metode pemrosesan gambar untuk meningkatkan kontras dengan membagi gambar menjadi bagian-bagian kecil.
    - Algoritma: Rangkaian instruksi atau langkah sistematis untuk menyelesaikan masalah tertentu.
- **C**
    - CLAHE (*Contrast Limited Adaptive Histogram Equalization*): Teknik pemrosesan gambar untuk meningkatkan kontras secara lokal tanpa memperbesar noise secara berlebihan.
    - CSV (*Comma-Separated Values*): Format file teks yang digunakan untuk menyimpan data dalam tabel yang dipisahkan oleh koma.
- **D**
    - Digitalisasi: Proses mengubah data atau dokumen fisik menjadi bentuk digital yang dapat diolah oleh komputer.
- **E**
    - *Environment Variables*: Variabel sistem yang digunakan untuk menentukan direktori dan konfigurasi perangkat lunak pada sistem operasi

- **F**
    - Folder: Direktori atau tempat penyimpanan di komputer yang digunakan untuk mengatur file.
- **G**
    - *Grayscale*: Format gambar yang hanya memiliki warna hitam, putih, dan berbagai tingkat abu-abu.
- **I**
    - *Image Preprocessing*: Serangkaian langkah untuk mempersiapkan gambar sebelum dianalisis atau diproses lebih lanjut, seperti meningkatkan kontras atau menghapus noise.
- **L**
    - *Library*: Kumpulan kode yang dirancang untuk digunakan kembali oleh pengembang dalam berbagai proyek pemrograman.
    - Lingkungan Kerja (*Working Environment*): Konfigurasi perangkat keras dan perangkat lunak yang digunakan untuk menjalankan program atau proyek.
- **M**
    - *Median Blurring*: Teknik penghapusan noise pada gambar dengan menggunakan nilai tengah dari piksel di sekitarnya.
    - *Modular*: Desain sistem yang dibagi menjadi beberapa bagian independen sehingga lebih mudah untuk dikembangkan atau diubah.
- **O**
    - OCR (*Optical Character Recognition*): Teknologi yang digunakan untuk mengenali teks dalam gambar dan mengubahnya menjadi format digital yang dapat diedit.
    - *Open-Source*: Perangkat lunak yang kode sumbernya tersedia untuk umum dan dapat dimodifikasi oleh siapa saja.
- **P**
    - Pipeline: Serangkaian langkah atau proses yang dirancang untuk mencapai tujuan tertentu dalam pengolahan data.
    - PNG (*Portable Network Graphics*): Format gambar yang mendukung transparansi dan memiliki kualitas lebih baik dibandingkan JPEG.

- **R**
    - Resolusi: Jumlah detail atau kepadatan piksel dalam gambar, biasanya diukur dalam DPI (*dots per inch*).
- **S**
    - Subdirektori: Folder di dalam folder utama yang digunakan untuk menyimpan data atau hasil pemrosesan tertentu.
- **T**
    - Tesseract: Perangkat lunak sumber terbuka untuk OCR.
    - *Thresholding*: Proses mengubah gambar berwarna atau *grayscale* menjadi gambar biner dengan memilih nilai ambang tertentu.

Referensi

- Tesseract OCR: https://github.com/UB-Mannheim/tesseract/releases/tag/v5.4.0.20240606
- Python: https://www.python.org
- Library Python:
    - pandas: https://pandas.pydata.org
    - Pytesseract: https://github.com/madmaze/pytesseract
    - pdf2image: https://github.com/Belval/pdf2image
    - Pillow: https://pillow.readthedocs.io
- Visual Studio Code (VSCode): https://code.visualstudio.com
- Dokumentasi CLAHE: https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html
- GitHub Repository: https://github.com/EchoNautilus/Strukturisasi-Data-dan-Arsip-Digital-Dokumen-Pemilikan-Tanah