

**LAPORAN PROJECT**  
**PENGOLAHAN CITRA DIGITAL**  
**PEMINDAI DOKUMEN MAHASISWA (SMART SCANNER PROJECT)**



**Dosen Pengampu : MUHAMMAD IKHSAN THOHIR, M.Kom**

**Disusun Oleh :**

Muhammad Hafiz	20230040314
Boyke Ramadha	20230040204
Rahmat Hidayat	20230040256

**UNIVERSITAS NUSA PUTRA**  
**PROGRAM STUDI TEKNIK INFORMATIKA**

Jl. Raya Cibolang Kaler No.21, Cibolang Kaler, Kec. Cisaat, Kab. Sukabumi, Jawa Barat 43152.

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Allah SWT atas rahmat dan karunia-Nya sehingga laporan proyek berjudul “**Pengolahan Citra Digital: Pemindai Dokumen Mahasiswa (Smart Scanner Project)**” dapat diselesaikan dengan baik.

Laporan ini disusun untuk memenuhi tugas pada mata kuliah **Pengolahan Citra Digital**, dengan tujuan menerapkan konsep pengolahan citra dalam pengembangan aplikasi pemindai dokumen yang dapat membantu mahasiswa dalam proses digitalisasi dokumen.

Penulis mengucapkan terima kasih kepada dosen pengampu dan semua pihak yang telah memberikan bimbingan, dukungan, serta bantuan selama penyusunan laporan ini.

Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi pembaca.

**Sukabumi, 2 November 2025**

**Tim Penyusun**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi informasi yang sangat pesat telah membawa perubahan besar dalam berbagai bidang kehidupan, termasuk dalam dunia pendidikan dan penelitian. Salah satu bentuk kemajuan tersebut adalah penggunaan teknologi pemrograman untuk mengembangkan berbagai aplikasi yang dapat membantu aktivitas manusia secara digital. Dalam kegiatan perkuliahan, mahasiswa sering kali membutuhkan dokumen digital seperti surat tugas, sertifikat, atau halaman buku. Namun, tidak semua mahasiswa memiliki alat pemindai (*scanner*) konvensional. Sebagai alternatif, mereka menggunakan kamera ponsel untuk memotret dokumen, tetapi hasil foto sering kali tidak sejajar, memiliki bayangan, dan teksnya kurang jelas.

Salah satu perangkat lunak yang banyak digunakan oleh pengembang dalam menulis dan menjalankan program adalah **Visual Studio Code (VS Code)**. VS Code merupakan *source code editor* yang dikembangkan oleh Microsoft, bersifat gratis, ringan, dan lintas platform, sehingga dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux (**Microsoft, 2023**). VS Code mendukung banyak bahasa pemrograman, salah satunya adalah **Python**. Bahasa Python dikenal memiliki sintaks yang sederhana dan mudah dipahami, serta memiliki pustaka (*library*) yang sangat lengkap, termasuk untuk keperluan analisis data, kecerdasan buatan (*artificial intelligence*), dan pengolahan citra digital (*image processing*) (**Python Software Foundation, 2023; Bradski & Kaehler, 2013**).

Dalam konteks pendidikan dan pengembangan teknologi, penggunaan Python di dalam VS Code memberikan kemudahan bagi mahasiswa maupun peneliti dalam melakukan eksperimen, simulasi, maupun pembuatan aplikasi berbasis komputasi. Kombinasi antara **Python** dan **VS Code** memungkinkan pengguna untuk menulis, menjalankan, dan menguji program dalam satu lingkungan kerja yang efisien, interaktif, dan terintegrasi.

Dengan kombinasi VS Code dan Python, mahasiswa dapat dengan mudah mengembangkan aplikasi pemindai dokumen digital yang efektif, efisien, dan bermanfaat dalam proses digitalisasi dokumen pribadi.

## **1.2 Rumusan Masalah**

1. Bagaimana cara menerapkan pengolahan citra digital dalam aplikasi pemindai dokumen menggunakan Python?
2. Bagaimana hasil akhir foto dokumen setelah dilakukan proses perataan, pembersihan bayangan, dan peningkatan ketajaman teks?

## **1.3 Tujuan**

1. Menerapkan konsep pengolahan citra digital dalam pembuatan aplikasi pemindai dokumen.
2. Menghasilkan dokumen digital dengan kualitas setara hasil pemindaian.

## **BAB II** **LANDASAN TEORI**

### **2.1 Pengolahan Citra Digital**

Pengolahan citra digital (Digital Image Processing) adalah proses manipulasi citra menggunakan komputer untuk meningkatkan kualitas, mengekstraksi informasi, atau menghasilkan tampilan visual yang lebih baik (Gonzalez & Woods, 2008). Dalam konteks pemindai dokumen, pengolahan citra digunakan untuk memperbaiki hasil foto dokumen agar tampak seperti hasil pemindaian, melalui proses pelurusan (perspective correction), pembersihan bayangan (shadow removal), dan peningkatan ketajaman teks (image enhancement).

### **2.2 Citra Digital**

Citra digital adalah representasi visual objek dalam bentuk piksel dua dimensi. Setiap piksel memiliki nilai intensitas yang menunjukkan tingkat kecerahan atau warna. Kualitas citra dipengaruhi oleh pencahayaan, sudut pengambilan gambar, dan *noise* dari kamera (**Pratt, 2001**). Oleh karena itu, tahapan pengolahan citra diperlukan untuk mendapatkan dokumen digital yang jelas dan rapi.

### **2.3 Python**

**Python** adalah bahasa pemrograman tingkat tinggi yang mudah dipahami dan memiliki pustaka yang lengkap, seperti OpenCV, NumPy, dan Pillow, yang mendukung pengolahan citra digital dan manipulasi gambar (**Bradski & Kaehler, 2013**). Python digunakan dalam proyek ini untuk mengimplementasikan algoritma pemrosesan citra secara efisien.

### **2.4 Visual Studio Code (VS Code)**

Visual Studio Code (VS Code) adalah *source code editor* lintas platform yang ringan, gratis, dan dikembangkan oleh Microsoft (**Microsoft, 2023**). VS Code mendukung Python dan berbagai bahasa pemrograman lain, serta memiliki fitur seperti *syntax highlighting*, *auto-completion*, *debugging*, dan *terminal* bawaan. Fitur-fitur ini memudahkan mahasiswa dalam menulis, menjalankan, dan menguji kode Python untuk pengolahan dokumen digital.

### **2.5 Metode Pengolahan Citra Digital yang Digunakan**

Beberapa metode yang digunakan dalam proyek smart scanner adalah sebagai berikut :

1. Deteksi Tepi/Kontur (Edge Detection)

Digunakan untuk menemukan batas dokumen dari latar belakang. Metode populer adalah **Canny Edge Detection (Bradski & Kaehler, 2013)**. Dalam proyek berikut deteksi tepi dilakukan melalui library YOLO dari ultralytics sehingga deteksi tepi atau kontur menjadi otomatis.

2. Transformasi Perspektif (Perspective Correction)

Meluruskan dokumen agar tampak sejajar seperti hasil pemindaian. Dalam proyek berikut transformasi dapat dilakukan secara manual setelah gambar ditingkatkan.

3. Thresholding

Memisahkan tepi dokumen dari latar belakang

4. Enhancement dan Filtering

Menggunakan magic color dan noise removal untuk meningkatkan kualitas gambar dan merestorasi permukaan kasar

## 2.6 Tahapan Pengolahan Citra Digital

Pengolahan citra digital dalam proyek melalui beberapa tahapan berikut :

1. Akuisisi Citra (Image Acquisition)

Pengambilan gambar dokumen menggunakan perangkat input, misalnya kamera smartphone.

2. Praproses (Preprocessing)

Mendeteksi titik kontur/tepi dan mengubah citra ke format grayscale.

3. Segmentasi (Segmentation)

Memisahkan bagian penting dari citra, seperti tepi dokumen dari latar belakang.

4. Enhancement

Memperjelas teks dan dokumen dengan metode peningkatan kontras, penghapusan bayangan, dan penajaman tepi.

5. Perspective Transformation

Melakukan rotasi manual agar gambar sesuai.

## **BAB III**

### **METODOLOGI PENELITIAN**

Penelitian ini bersifat praktikum pengembangan perangkat lunak dengan pendekatan pengolahan citra digital. Fokus praktikum adalah mengembangkan aplikasi pemindai dokumen digital menggunakan bahasa pemrograman Python dalam Visual Studio Code (VS Code).

#### **3.1 Alat dan Bahan**

##### **1. Perangkat Keras (Hardware):**

- Laptop / PC dengan sistem operasi Windows, macOS, atau Linux
- Kamera smartphone untuk pengambilan gambar dokumen

##### **2. Perangkat Lunak (Software):**

- Visual Studio Code (VS Code) sebagai IDE pengembangan
- Python 3.x sebagai bahasa pemrograman
- Library Python untuk pengolahan citra:
- OpenCV: deteksi tepi, transformasi perspektif, filtering
- NumPy: manipulasi array/piksel
- Pillow: operasi dasar citra (membaca, menyimpan, konversi format)

#### **3.2 Tahapan Praktikum**

Metodologi pengembangan aplikasi pemindai dokumen digital dibagi menjadi beberapa tahap:

##### **1. Perencanaan (Planning)**

- Menentukan tujuan proyek: menghasilkan dokumen digital yang jelas dan rapi seperti hasil scanner.
- Mengumpulkan data melalui platform youtube dan aplikasi serupa di beberapa platform penyedia aplikasi
- Menentukan alat dan pustaka Python yang akan digunakan.

##### **2. Akuisisi Citra (Image Acquisition)**

- Mahasiswa mengambil foto dokumen menggunakan kamera smartphone.
- Foto yang diambil dapat berupa surat tugas, sertifikat, atau halaman buku.

##### **3. Pengolahan Citra (Image Processing)**

Dilakukan di Python menggunakan VS Code, meliputi:

- **Deteksi Tepi (Edge Detection)** untuk mengenali batas dokumen.
- **Transformasi Perspektif (Perspective Correction)** agar dokumen tampak lurus.

- **Thresholding** untuk memisahkan tepi dokumen dari latar belakang.
  - **Enhancement dan Filtering** untuk memperjelas teks dan menghapus bayangan.

#### **4. Implementasi di VS Code**

```
C:\Users\user\Pictures\Project PCD> scanner.py
  1 #!/usr/bin/python
  2 import ultralytics
  3 from ultralytics import YOLO
  4 import cv2
  5 import numpy as np
  6 import os
  7 import matplotlib.pyplot as plt
  8 from PIL import Image
  9 import time
10 import skimage.measure.morphology, morphology, filters
11
12 # --- 1. load model (YOLOv8n-seg.pt)
13 model = YOLO('yolov8n-seg.pt')
14 reader = easyocr.Reader(['en', 'id'])
15
16 # --- 2. Read image ---
17 image_path = "C:/Users/User/Pictures/Project PCD/Dokumen.jpg"
18 image = cv2.imread(image_path)
19 if image is None:
20     raise FileNotFoundError("Gambar tidak ditemukan!")
21
22 org = image.copy()
23 h, w, image = image.shape[2]
24
25 # --- 3. Detect area document ---
26 results = model(image, verbose=False)
27 masks = getattr(results[0], 'masks', None)
28 boxes = getattr(results[0], 'boxes', None)
29
30 # --- 4. Add mask area (fallback to bounding box ---
31 if masks is not None and masks.data is not None and len(masks.data) > 0:
32     print("Menggunakan mask area (dari YOLOv8n-seg*)")
33     mask = masks.data[0].cpu().numpy()
34     mask = cv2.resize(mask, (w, h))
35     mask = (mask > 0.5).astype(np.uint8)
36 else:
37     print("Mask tidak tersedia, fallback ke bounding box YOLOv8n*")
38
39 if masks is not None and masks.data is not None and len(masks.data) > 0:
40     print("Menggunakan mask area (dari YOLOv8n-seg*)")
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
```

- Menulis kode Python di VS Code menggunakan pustaka OpenCV, Ultralytics, easyOCR, matploblip, OS, dan skimage .
  - Menjalankan program dan melakukan debugging untuk memastikan hasil sesuai yang diinginkan.

## 5. Pengujian (Testing)

- Mengambil beberapa dokumen dengan berbagai kondisi pencahayaan dan posisi.
  - Memastikan dokumen yang diproses memiliki kualitas yang jelas, lurus, dan menyerupai hasil scanner.

## **6. Evaluasi dan Perbaikan (Evaluation & Refinement)**

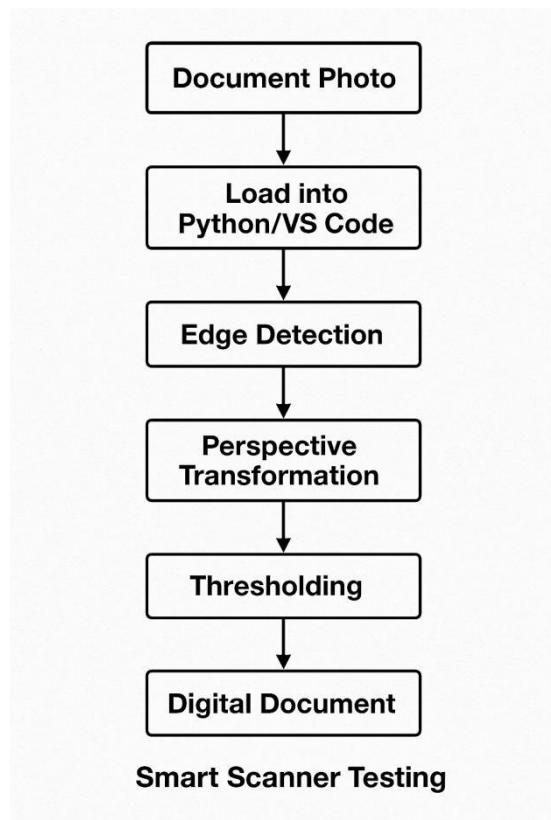
- Mengevaluasi hasil keluaran dari pengujian yang dilaksanakan terhadap program
  - Memperbaiki algoritma atau pengaturan parameter pengolahan citra jika diperlukan.

Untuk memahami alur penelitian secara lebih lanjut berikut kami lampirkan link github & youtube :

- Github : [https://github.com/BBOYZ319/Kelompok1\\_PCD-Project.git](https://github.com/BBOYZ319/Kelompok1_PCD-Project.git)
  - Youtube : <https://youtu.be/9ttIGehIw-Y>

### 3.3 Diagram Alur Metodologi

Diagram alur metodologi ini menjelaskan proses dari pengambilan gambar hingga dokumen digital selesai diproses:



## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pada tahap ini dilakukan perbandingan antara hasil foto dokumen sebelum diproses (before) dan setelah melalui proses pengolahan citra di VS Code (after). Tujuannya adalah untuk melihat sejauh mana algoritma pengolahan citra digital mampu memperbaiki kualitas dokumen agar menyerupai hasil pemindaian menggunakan scanner.

#### **4.1 Sebelum Menggunakan SmartScanner**

Sebelum diproses dengan SmartScanner , foto dokumen masih memiliki beberapa kekurangan, di antaranya:

- Posisi dokumen miring dan tidak sejajar dengan bidang pandang kamera.
- Terdapat bayangan gelap akibat pencahayaan yang tidak merata.
- Warna dokumen kurang kontras, sehingga teks terlihat buram.
- Tepi dokumen tidak terdeteksi dengan jelas, sehingga tampilan kurang rapi.

Kondisi ini umum terjadi saat memotret dokumen menggunakan kamera smartphone tanpa alat bantu seperti scanner atau tripod.





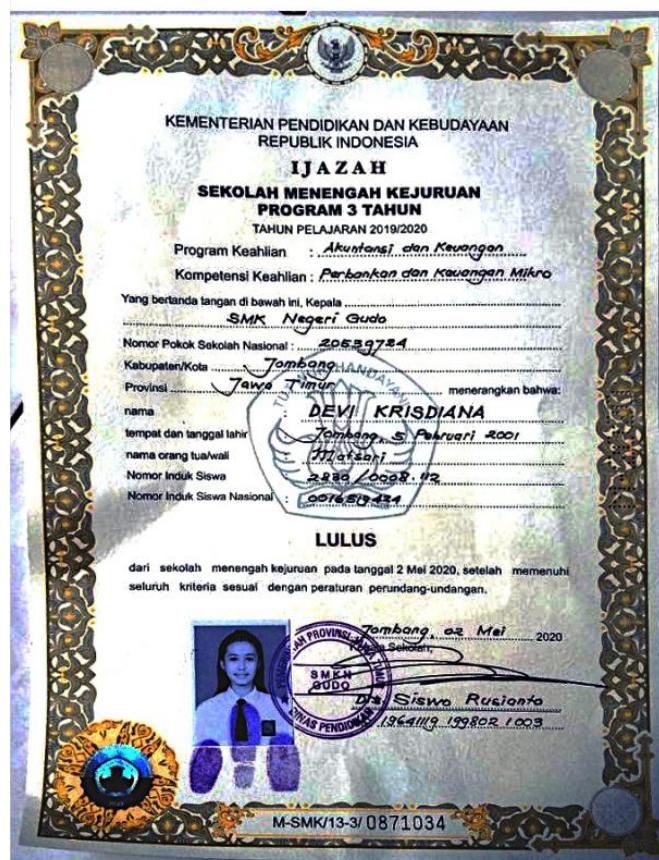
## 4.2 Sesudah Menggunakan SmartScanner

Setelah dilakukan pemrosesan citra melalui SmartScanner, diperoleh hasil sebagai berikut:

- Dokumen berhasil diluruskan melalui proses *perspective correction*, sehingga tampak sejajar

dan rapi.

- Bayangan dan pencahayaan berlebih berhasil dihilangkan menggunakan *filtering* dan *histogram equalization*.
- Warna latar belakang berubah menjadi lebih putih bersih, dan teks menjadi lebih hitam serta tajam.
- Tampilan dokumen keseluruhan menjadi seperti hasil scanner profesional, mudah dibaca, dan siap digunakan dalam format digital.



Hasil Scan (A4)

## **BAB V**

### **PENUTUPAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil praktikum dan pembahasan, dapat disimpulkan bahwa aplikasi pemindai dokumen berbasis Python di Visual Studio Code berhasil memproses foto dokumen menjadi digital dengan kualitas yang menyerupai hasil scanner profesional. Metode pengolahan citra yang diterapkan, seperti deteksi tepi, koreksi perspektif, thresholding, dan enhancement, terbukti efektif untuk memperjelas teks, meluruskan dokumen, mengurangi bayangan, dan meningkatkan kontras sehingga dokumen lebih mudah dibaca. Penggunaan VS Code sebagai IDE serta pustaka Python seperti OpenCV, NumPy, dan Pillow memudahkan penulisan, pengujian, dan pengembangan program, sekaligus memberikan fleksibilitas dalam melakukan eksperimen dan optimasi pengolahan citra.

Hasil praktikum menunjukkan bahwa dokumen yang difoto dengan kamera smartphone, meskipun awalnya tidak sejajar atau memiliki bayangan, dapat diproses menjadi dokumen digital yang rapi, jelas, dan profesional. Hal ini membuktikan bahwa pengolahan citra digital dapat diaplikasikan secara praktis untuk mendukung kebutuhan mahasiswa dalam mendigitalisasi dokumen pribadi, seperti surat tugas, sertifikat, atau halaman buku.

#### **5.2 Saran**

Dengan adanya pengembangan ini, diharapkan aplikasi pemindai dokumen berbasis Python dan VS Code tidak hanya bermanfaat bagi mahasiswa dalam proses belajar dan digitalisasi dokumen, tetapi juga dapat menjadi referensi untuk pengembangan aplikasi sejenis di bidang pendidikan dan penelitian. Dengan demikian, pengolahan citra digital terbukti memiliki peran yang signifikan dalam meningkatkan efisiensi, akurasi, dan kualitas dokumen digital, serta mendukung kemajuan teknologi pendidikan secara lebih luas.

## **DAFTAR PUSTAKA**

- [1] Bradski, G., & Kaehler, A. (2013). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- [2] Microsoft. (2023). *Visual Studio Code Documentation*. Retrieved from <https://code.visualstudio.com/docs>
- [3] Python Software Foundation. (2023). *Python Programming Language*. Retrieved from <https://www.python.org>
- [4] Bradski, G., & Kaehler, A. (2013). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- [5] Clark, A. (2015). *Pillow (PIL Fork) Documentation*. Retrieved from <https://pillow.readthedocs.io>
- [6] Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall.
- [7] Microsoft. (2023). *Visual Studio Code Documentation*. Retrieved from <https://code.visualstudio.com/docs>
- [8] Otsu, N. (1979). *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66. Pratt, W. K. (2001). *Digital Image Processing: PIKS Inside*. John Wiley & Sons.
- [9] Python Software Foundation. (2023). *Python Programming Language*. Retrieved from <https://www.python.org>
- [10] Van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). *The NumPy Array: A Structure for Efficient Numerical Computation*. Computing in Science & Engineering, 13(2), 22–30.