

## MODEL APLIKASI DOCUMENT SCANNER MENGGUNAKAN OPERATOR CANNY DAN CONTOUR PADA OPEN CV BERBASIS DESKTOP

Muhamad Rizki Pratama<sup>1</sup>, Erlangga Putra Bhayangkara<sup>2</sup>, Juan Mohammad Ishlah<sup>3</sup>

Prodi Teknik Informatika, Universitas Muhammadiyah Prof Dr Hamka.

Email : rizkiprma03@gmail.com<sup>1</sup>, erlangga@gmail.com<sup>2</sup>, Juan@gmail.com<sup>3</sup>

### ABSTRAK

Pengolahan citra digital merupakan usaha untuk melakukan suatu transformasi pada citra sehingga menjadi citra lain dengan menggunakan teknik tertentu dari teknologi komputer. Open CV adalah sebuah library yang bersifat open source untuk pengolahan citra secara real time. Mengubah citra digital menggunakan Open CV dalam pendeteksian objek dari sebuah gambar asli menjadi citra digital agar mudah dilihat dan dikelola seperti mengubah sebuah dokumen berbasis kertas menjadi dokumen tanpa kertas (Document Scanner). Tentunya dengan pengolahan citra digital dari citra asli menjadi sebuah arsip elektronik akan mudah disimpan dan lebih mudah dikirim kemanapun dari jarak jauh. Dalam membangun sistem Document Scanner ini dibutuhkan library Open CV serta metode Canny dan Contour yang digunakan untuk mendeteksi tepi pada objek dokumen. Membangun aplikasi ini dibutuhkan perangkat lunak JetBrains PyCharm sebagai kompilator lalu Python sebagai bahasa pemrograman. Berdasarkan hasil citra uji dari metode tersebut akan menghasilkan citra original dan citra hitam putih dari menggunakan Operator Canny dan Contour untuk mencari titik sudut dokumen dalam mendeteksi objek.

**Kata Kunci:** Computer Vision, OpenCV, Canny, Python, Deteksi Tepi.

### I. PENDAHULUAN

Perkembangan teknologi yang semakin maju hingga di era sekarang ini sangat lah pesat dan canggih, banyak teknologi-teknologi masa kini sudah dikembangkan dengan memadukan beberapa teknologi yang lainnya menjadi satu kesatuan hingga muncul teknologi yang berbasis digital hal itu dilakukan untuk mempermudah aktifitas serta bermanfaat untuk peggunganya. Salah

satu teknologi digital yang digunakan dalam memproses sebuah gambar ataupun citra yaitu pengolahan citra digital.

Pengolahan citra merupakan suatu persepsi visual yang termasuk didalamnya sebuah proses pengolahan dan analisis pada citra. Pada proses ini adanya data yang di input serta output yang hasilnya berbentuk sebuah citra. Meskipun citra dapat merepresentasikan objek

gambar seringkali citra mengalami penurunan intensitas mutu, misalnya mengandung noise, kontras terlalu tinggi tentu citra seperti ini akan menjadi berkurang kualitasnya. Dengan melalui citra yang diproses dalam berbentuk digital (*digital image processing*) akan membuat perbaikan citra (*image restoration*), serta peningkatan pada kualitas citra (*image enhancement*) akan menghasilkan citra yang baru [1]. Agar citra tidak mengalami penurunan mutu maka citra tersebut perlu adanya manipulasi dengan citra lain yang kualitasnya lebih baik.

Dalam sebuah proses pengolahan citra ini ialah merubah suatu objek dokumen menjadi file digital dengan input berupa citra gambar yang di proses berbagai metode pendeteksian tepi. Pendeteksi tepi merupakan objek-objek gambar yang diproses dengan menghasilkan tepi-tepi citra dan dapat mempresentasikan bentuk pada objek tersebut [2]. Lalu menentukan garis tepi serta titik sudut pada objek dokumen menggunakan contour yang sebelumnya dalam proses pendeteksian tepi menggunakan operator canny. Operator canny merupakan algoritma deteksi tepi yang meneliti pemodelan visual manusia [3] serta sebagai pendukung objek dokumen yang telah dipindai selanjutnya dilakukan transformation perspective untuk merubah sudut pandang agar objek terlihat dari atas.

Berdasarkan metode tersebut maka dilakukan sebuah penelitian untuk merancang sebuah aplikasi dokumen scanner berbasis desktop dengan menggunakan algoritma

canny dan contour dalam proses mendeteksi citra yang akan dipindai serta dari penelitian tersebut dapat memahami bahwa merancang aplikasi dokumen scanner menggunakan pustaka Open Cv dan bahasa pemrograman python berbasis desktop bisa dibangun dengan standar computer vision. Hal ini agar aplikasi dokumen scanner dapat dikembangkan lebih lanjut menggunakan metode yang sudah dipaparkan.

## II. LANDASAN TEORI

### 2.1 Pengertian Aplikasi

Secara umum, aplikasi adalah suatu program berupa perangkat lunak (*Software*) yang berjalan pada suatu sistem tertentu, yang berguna dalam mendukung berbagai aktivitas yang dilakukan oleh manusia [4] atau program yang secara langsung dapat melakukan proses-proses yang digunakan dalam komputer oleh pengguna [5].

### 2.2 Pengertian Pemindai (Scanner)

Pemindai (*Scanner*) adalah alat untuk merekam objek yang ada di depannya, pengoperasian alat pemindai hampir sama dengan mesin fotokopi yang dapat melihat hasilnya langsung di atas kertas, namun *scanner* ini dapat menghasilkan yang berbentuk file gambar atau dokumen digital [6].

### 2.3 Pengertian Aplikasi Desktop

Aplikasi desktop berguna untuk memberikan kemudahan bagi pengguna komputer atau laptop dalam mengelola data sehingga dapat menghasilkan laporan dengan cepat, akurat dan tepat [7]. Pendapat lainnya mengatakan bahwa aplikasi

desktop adalah aplikasi yang dapat berjalan secara independen dalam sistem desktop komputer atau laptop dapat melakukan berbagai macam operasi yang ditentukan oleh pengguna [8]

#### 2.4 Definisi Citra Digital

Citra digital adalah representasi yang bentuk diskrit dari fungsi intensitas cahaya pada bidang dua dimensi [9]. Sebuah citra digital diperoleh dari sekumpulan berupa gambar maupun video [10]. Dalam sebuah artikel yang ditulis oleh I. Fawwaz dan N. Dharshinni menyatakan bahwa citra yang dapat diproses oleh sebuah komputer disebut citra digital [11].

#### 2.5 Pengolahan Citra Digital

Suatu kegiatan dengan tujuan untuk mengubah informasi menjadi digital dari sebelumnya citra non-digital atau citra fisik. Menurut Sutoyo dkk, dalam sebuah artikelnya yang ditulis oleh Mas'ud Effendi, Fiitriyah dan Usman Effendi menjelaskan bahwa pengolahan citra digital adalah suatu teknik pengolahan citra yang bertujuan untuk meningkatkan kualitas suatu citra sehingga dapat dengan mudah diinterpretasikan oleh manusia atau komputer, baik itu dalam bentuk gambar maupun animasi [12].

#### 2.6 Computer Vision

Computer vision adalah proses pengolahan gambar dan video untuk mendapatkan hasil analitik [10]. Dengan kata lain, computer vision bertujuan untuk membangun sebuah mesin cerdas yang dapat “melihat” [13] artinya, computer vision memiliki fungsi yang sama dengan mata

manusia yaitu vision atau penglihatan [10].

#### 2.7 Deteksi Tepi (*Edge Detection*)

Deteksi tepi pada citra merupakan proses yang menghasilkan garis tepi dari objek pada citra [2]. Dengan pengertian lainnya, deteksi tepi merupakan prasyarat yang berguna untuk identifikasi objek dan registrasi segmentasi citra [14]. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra jika titik tersebut berbeda nilai pixel yang tinggi dengan nilai pixel tetangganya [2]. Selain itu, tujuan pendeteksian tepi adalah untuk memperbaiki detail dari citra yang kabur, karena *error* atau adanya efek dari sebuah proses akuisisi pada gambar [15].



Gambar 2.1

Contoh deteksi tepi

#### 2.8 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah perangkat lunak pustaka yang ditujukan untuk pemrosesan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez [16]. Modul pustaka OpenCV ini dibangun dengan cara yang sangat kuat dan fleksibel untuk memecahkan sebagian besar masalah computer vision yang solusinya sudah tersedia, seperti memotong citra, meningkatkan kualitas citra dengan mengubah kecerahan, ketajaman, kontras, deteksi bentuk, segmentasi gambar,

deteksi objek bergerak, mengenali objek, dan lain-lain [9].



**Gambar 2.2**  
OpenCV

### 2.9 Operator Canny

Salah satu operator deteksi tepi yang dikembangkan oleh John F. Canny [17]. Operator Canny menggunakan Kernel Gaussian Derivative untuk menyaring noise dari gambar asli untuk mendeteksi objek dengan permukaan halus[18].

Menurut I. Fawwaz dan N. Dharshinni dalam artikelnya menyatakan deteksi canny ini memiliki pendeteksian paling optimum beberapa kriteria pada algoritma Canny [11] :

- a. Mendeteksi dengan baik (kriteria deteksi)  
Kemampuan untuk menempatkan dan menandai semua tepi yang sesuai berdasarkan parameter konvolusi yang dilakukan. Sekaligus memberikan fleksibilitas tinggi dalam menentukan tingkat deteksi ketebalan tepi yang diinginkan.
- b. Melokalisasi dengan baik (kriteria lokalisasi)  
Dengan Canny dimungkinkan akan menghasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
- c. Respon yang jelas (kriteria respon)  
Memiliki hanya satu respon untuk tiap tepi. Sehingga

memudahkan mendeteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya.

### 2.10 Kontur (Contour)

Titik dalam citra digital di mana kecerahan berubah secara tiba-tiba dapat dianggap sebagai garis atau kurva yang biasa dikenal sebagai tepi [19]. Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi ke dalam suatu citra [3].

Rangkaian dari berbagai pixel-pixel tepi yang membentuk batas daerahnya (*region boundary*) disebut Kontur (*Contour*). Kontur dapat terbuka maupun tertutup, adapun bentuk kontur tertutup memiliki korelasi dengan batas tepi yang mengelilingi suatu daerah (*region*) [17].



**Gambar 2.3**

- (a) kontur tertutup, (b) kontur terbuka

### 2.11 UML

Unified Modelling Language (UML)

adalah bahasa berupa sekumpulan diagram yang digunakan untuk menggambarkan suatu sistem atau perangkat lunak berbasis objek [20]. UML muncul dari kebutuhan akan pemodelan visual untuk mendefinisikan, mendeskripsikan, membangun, dan mendokumentasikan sistem perangkat lunak [21].

### 2.12 Metode Waterfall

Waterfall adalah jenis model penelitian aplikasi yang diperkenalkan

ke dalam *classic life cycle* (siklus hidup klasik), yang berarti metode tersebut memiliki karakter yang berurutan dan sistematis [22].

### III. ANALISIS MASALAH

Pada tahap analisis ini, penulis memperoleh data berdasarkan observasi dan studi literatur sebagai adanya pengetahuan untuk memahami konsep pengolahan citra digital. Berdasarkan data yang diperoleh, tidak sedikit sering ditemukan kualitas gambar yang menurun karena noise atau kurangnya bagian detail tepi citra dan pencahayaan yang tidak konsisten. Masalah tersebut dapat diperbaiki melalui perbaikan citra (*image processing*), penapisan derau (*noise filtering*), pemugaran citra (*image restoration*), serta peningkatan pada kualitas citra (*image enhancement*) yang akan menghasilkan citra yang baru berdasarkan pemrograman pengolahan citra yang diproses menggunakan library *open source* yaitu OpenCV.

Pada analisis masalah diatas maka penulis melakukan sebuah penelitian untuk merancang sebuah aplikasi dokumen scanner berbasis desktop yang dibangun dengan standar computer vision untuk dapat melakukan proses perbaikan citra melalui pengolahan citra digital.

Terdapat juga analisis kebutuhan software dan hardware untuk proses rancang bangun document scanner application di antaranya yaitu:

1. Kebutuhan Software
  - a. JetBrains Pycharm Edition 2019. 2.4
  - b. PyQt5 Designer

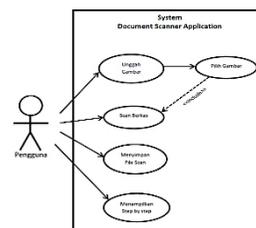
2. Kebutuhan Hardware
  - a. Laptop / Komputer
  - b. Smartphone

## IV. PERANCANGAN SISTEM

### 4.1 Desain Sistem

#### 4.1.1 Use Case Document Scanner Application

Berikut gambar *use case* diagram pada sistem document scanner application yang dibuat ini. Pengguna sebelum melakukan scanning, pengguna harus memilih file berformat JPEG/JPG untuk di upload ke aplikasi tersebut sebelum akhirnya dilakukan scan. Gambar *use case* dapat di lihat pada gambar berikut ini.

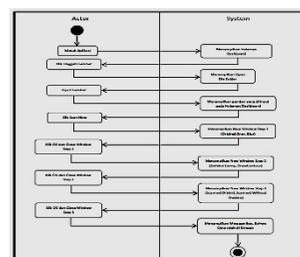


**Gambar 4.1**

Use Case Diagram Aplikasi

#### 4.1.2 Activity Diagram

Dalam Activity Diagram ini melibatkan setidaknya 2 aktor yaitu pengguna dan aplikasi. Pada diagram ini meliputi bagaimana cara user melakukan scanning document.



**Gambar 4.2**

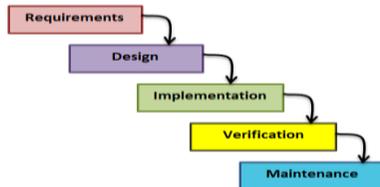
Activity Diagram Aplikasi

## 4.2 Model Pengembangan Sistem

### 4.2.1 Model Waterfall

Pada penelitian ini, dilakukan beberapa tahapan dalam merancang aplikasi document scanner dengan sesuai yang direncanakan.

Berikut ini adalah tahapan-tahapan pembangunan sistem dari gambaran *Waterfall* model.



**Gambar 4.3**

Model *Waterfall*

Adapun tahapan-tahapan dari pengembangan sistem model waterfall sebagai berikut:

a) Analisis kebutuhan

Pada tahapan pertama yaitu Analisis kebutuhan yang diperlukan dalam membangun aplikasi document scanner berbasis desktop dengan cara mencari tahu kebutuhan apa saja yang diperlukan. Dimulai dari kebutuhan library open source, bahasa pemrograman hingga kebutuhan fungsional yang mencakup kebutuhan *hardware*, *software* dan kebutuhan informasi.

b) Desain

Tahapan kedua ini dilakukan pembuatan tampilan *interface* dari aplikasi yang akan dibuat, seperti penggunaan *font*, warna dan tata letak tombol. Serta pembuatan gambaran lengkap dari sebuah sistem yang akan dibangun sehingga akan

membantu kebutuhan *hardware*, dan tampilan sistem secara utuh.

c) Pembuatan kode program

Pada tahapan ini dilakukan sebuah implementasi dari desain yang sudah dibuat sebelumnya menjadi kode-kode program dan *library* programming yang selanjutnya akan diintegrasikan menjadi sistem aplikasi yang sesuai dengan yang diharapkan.

d) Pengujian

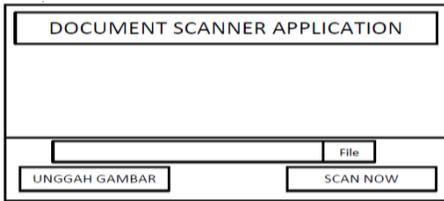
Pengujian dilakukan dengan maksud apakah sistem berjalan dengan baik sesuai yang diharapkan. Pengujian pada aplikasi yang di buat yaitu dengan menggunakan metode *Blackbox testing*, untuk memastikan jika program yang di bangun berjalan sesuai keinginan. Metode ini digunakan sebagai pengujian untuk menilai kebutuhan dan spesifikasi software serta untuk meninjau input dan output sistem tanpa harus mengetahui internal dari programnya.

e) Pemeliharaan

Pada tahapan terakhir ialah pemeliharaan yaitu dimana jika terjadi kesalahan program atau bug pada aplikasi tersebut maka dilakukan tahap perbaikan untuk menunjang sistem lebih baik lagi.

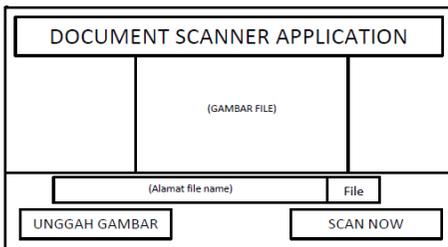
## 4.3 Desain Tampilan / User Interface

Desain tampilan dirancang dengan maksud untuk memudahkan dalam melakukan navigasi melalui sistem atau aplikasi yang dibuat. Tampilan desain pada sistem ini memiliki lima bagian, yaitu:



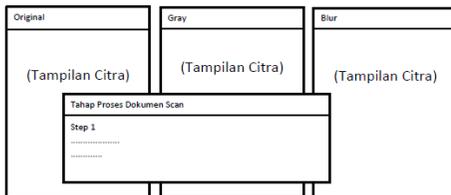
**Gambar 4.4**

Tampilan halaman dashboard awal



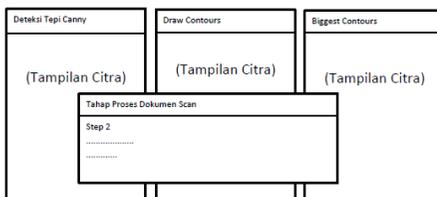
**Gambar 4.5**

Tampilan halaman dashboard setelah di inputkan file gambar



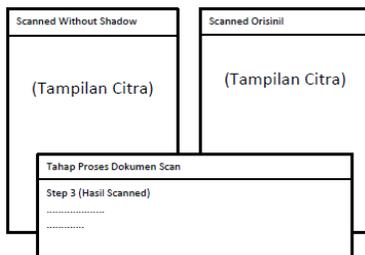
**Gambar 4.6**

Tampilan Step 1



**Gambar 4.7**

Tampilan Step 2



**Gambar 4.8**

Tampilan Step 3

## V. IMPLEMENTASI

### 5.1 Implementasi Perangkat Lunak

Setelah melalui tahapan desain, perancangan dan pengumpulan bahan kebutuhan, maka selanjutnya pada tahapan ini dilakukan implementasi pada aplikasi yang telah di bangun ini. Perangkat lunak yang digunakan adalah JetBrains PyCharm Community Edition 2019.2.4 x64.

#### 1. Tampilan Utama

Pada tampilan utama antarmuka (*Interface*) menyajikan tombol unggah gambar untuk memilih file berformat jpeg, jpg dan png yang akan di scan nantinya, tombol unggah gambar jika di klik maka akan muncul pop up untuk memilih gambar yang akan siap untuk di scan. Lalu tombol scan now ini akan aktif jika sudah ada file gambar yang telah di inputkan ke dalam aplikasi tersebut, jika file yang sudah di inputkan akan muncul di tengah-tengah aplikasi beserta alamat file itu berasal.



**Gambar 5.1**

Tampilan Dashboard

#### 2. Tampilan setelah di inputkan citra

Pada tampilan ini merupakan lanjutan dari tampilan sebelumnya, namun pada tampilan ini menunjukkan jika file gambar sudah diinputkan, maka gambar akan muncul di tengah-tengah bagian dashboard dan di bawahnya terdapat alamat beserta nama file nya.



**Gambar 5.2**

Tampilan setelah diinputkan file

### 3. Tampilan Step 1

Selanjutnya pada tampilan halaman dashboard setelah di klik scan now akan muncul window baru dimana window ini akan menampilkan layer citra original, citra grey dan citra blur hal ini ditampilkan sebagai pengetahuan langkah-langkah pengolahan document scanner dan terdapat pop up step 1 beserta penjelasannya.



**Gambar 5.3**

Tampilan step 1

### 4. Tampilan Step 2

Langkah selanjutnya pada new window tampilan step 2 menampilkan layer citra deteksi tepi canny, draw contour dan biggest contour. Pada tampilan step 2 ini adalah terusan dari step sebelumnya dalam pengolahan document scanner menggunakan library open cv.



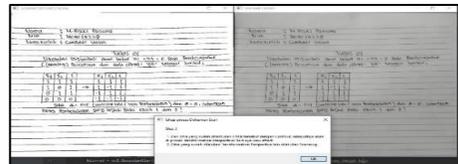
**Gambar 5.4**

Tampilan step 2

### 5. Tampilan Step 3

Pada tampilan step terakhir ini menampilkan hasil dari pengolahan

citra yang berupa citra scan, pada window ini menampilkan dua layer citra yaitu citra scanned orisinal dan citra scanned tanpa bayangan, jika sudah di step ketiga ini berarti tahap document scanner sudah selesai, jika user ingin melakukan scanner ulang maka tampilan *message box* tersebut klik "oke" dan close layer window citra tersebut.



**Gambar 5.5**

Tampilan step terakhir

## 5.2 Pengujian

Pada tahapan pengujian terhadap aplikasi yang sudah dirancang ini. Adapun pengujian yang sudah dilakukan menggunakan metode *blackbox*, hasil dari pengujian ini di dasarkan pada detail aplikasi seperti tampilan *user interface*, dan fungsi-fungsi tombol yang terdapat pada aplikasi ini.

Adapun spesifikasi system operasi dan perangkat keras yang dilakukan sebagai pengujian dengan aplikasi ini adalah sebagai berikut:

Sistem Operasi : Windows 10

Processor : Intel(R) Celeron(R)  
CPU N3060 @  
1.60GHz 1.60 GHz

RAM : 4 GB

## VI. KESIMPULAN

### 6.1. Kesimpulan

Berdasarkan hasil penelitian ini dalam pembuatan aplikasi document scanner menggunakan operator canny dan contour berbasis desktop dengan pustaka Open CV :

1. Penelitian ini menghasilkan aplikasi scan dokumen berbasis desktop yang dapat memperbaiki dan meningkatkan kualitas tampilan citra serta akan menghasilkan citra yang baru melalui pengolahan citra digital dengan standar computer vision menggunakan pustaka open cv.
2. Pembuatan aplikasi scan dokumen ini dapat dirancang dengan deteksi tepi menggunakan algoritma canny sehingga menghasilkan deteksi objek dengan halus.

## 6.2. Saran

Pembuatan aplikasi scan dokumen berbasis desktop ini merupakan contoh aplikasi yang dibangun menggunakan pustaka open source yaitu Open CV, dimana penelitian ini akan bersifat informatif dan bermanfaat bagi pembaca. Adapun saran untuk penelitian selanjutnya agar penelitian dapat lebih lanjut mengenai metode scan dokumen yang digunakan pada aplikasi ini, diharapkan penelitian selanjutnya terhadap teknologi ini akan menggunakan metode deteksi tepi (edge detection) lainnya, misalnya seperti Sobel, Prewitt, Robert atau lainnya dengan harapan penelitian lebih lanjut bahkan lebih baik.

## VII. DAFTAR PUSTAKA

- [1] E. Ardhiyanto, W. Hadikurniawati, and Z. Budiarmo, "Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video," *J. Teknol. Inf. Din.*, vol. 18, no. 2, pp. 91–100, 2013.
- [2] E. R. Kuswandi and N. Fadillah, "Perbandingan Metode Robert dan Metode Prewitt untuk Deteksi Tepi pada Citra Tanda Tangan," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 3, no. 2, pp. 155–158, 2019.
- [3] E. D. Ginting, "Deteksi Tepi Menggunakan Metode Canny Dengan Matlab Untuk Membedakan Uang Asli dan Uang Palsu," *J. Jur. Tek. Inform.*, vol. 1, no. 1, pp. 1–13, 2014.
- [4] A. E. Yanuar and M. A. Senubekti, "PERANCANGAN APLIKASI PENJUALAN ONLINE BERBASIS WEBSITE (STUDI KASUS : Bakso Emsa)," *Nuansa Inform.*, vol. 16, no. 1, pp. 19–32, 2022.
- [5] V. Sihombing and G. Juni Yanris, "Penerapan Aplikasi Dalam Mengolah Aset Desa (Studi Kasus : Kepenghuluan Sri Kayangan) 1)," *J. Mantik Penusa*, vol. 4, no. 1, pp. 12–15, 2020.
- [6] D. Susanti, M. S. Abdurrohman, P. S. Informatika, F. Teknik, and U. Majalengka, "Rancang bangun alat pemindai dokumen menggunakan raspberry pi 1,2," vol. 6, pp. 44–48, 2020.
- [7] E. Kurniawan and A. K. Syahputra, "Perancangan Aplikasi Pemesanan Dan Pembayaran Berbasis Desktop Pada Percetakan UD. AZKA GEMILANG Menggunakan Metode Prototype," *Semin. Nas. Raya*, vol. 9986, no. September, pp. 105–110, 2018.
- [8] I. Efendi and S. Annisa, "Terbit pada laman web jurnal : <http://ojsamik.amikmitragama.ac.id>," vol. 6, no. 1, pp. 122–126,

- 2022.
- [9] S. Ratna, "Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm," *Technol. J. Ilm.*, vol. 11, no. 3, p. 181, 2020.
- [10] K. M. Kaloh, V. C. Poekoel, and M. D. Putro, "Perbandingan Algoritma Background Subtraction dan Optical Flow Untuk Deteksi Manusia," *J. Tek. Inform.*, vol. 13, no. 1, pp. 1–9, 2018.
- [11] I. Fawwaz and N. P. Dharshinni, "Jurnal Sistem Informasi dan Teknologi Jaringan Perbandingan Deteksi Tepi Citra Menggunakan Operator Robert , Canny , dan Frei Chen Pada Citra Bitmap dan JPEG," vol. 2, pp. 41–45, 2021.
- [12] M. Effendi, F. Fitriyah, and U. Effendi, "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan," *J. Teknotan*, vol. 11, no. 2, p. 67, 2017.
- [13] T. K. Gautama, A. Hendrik, and R. Hendaya, "Pengenalan Objek pada Computer Vision dengan Pencocokan Fitur Menggunakan Algoritma SIFT Studi Kasus: Deteksi Penyakit Kulit Sederhana," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 3, pp. 437–450, 2016.
- [14] K. R. O. Recio and R. G. Mendoza, "Three-step approach to edge detection of texts," *Philipp. J. Sci.*, vol. 148, no. 1, pp. 193–211, 2019.
- [15] S. Reno and R. Edyal, "Analisa Perbandingan Deteksi Tepi Citra Foto Menggunakan Algoritma Robert dan Prewitt," *Multinetics*, vol. 2, no. 2, p. 11, 2016.
- [16] A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, no. 2, pp. 22–26, 2018.
- [17] D. E. Saputra and A. F. Ibadillah, "Pengolahan Citra Digital Dalam Penentuan Panen Jamur Tiram," *J. Tek. Elektro dan Komput. TRIAC*, vol. 6, no. 1, pp. 2–6, 2019.
- [18] M. Deteksi and T. Canny, "Segmentasi Pada Plat Kendaraan Dinas dengan," vol. 6, pp. 328–339, 2022.
- [19] K. Shella, "PROSES DETEKSI ORANG BERENANG MENGGUNAKAN BACKGROUND SUBTRACTION DAN CONTOUR DETECTION," 2021.
- [20] H. Sutisna and A. B. Hikmah, "Jurnal teknik informatika," *Implementasi Teknol. Augment. Real. Pada Media Promosi Perumah. Puri Melodi Mangkubumi Pt. Mustika Putri Jaya Di Kota Tasikmalaya*, vol. 5, no. 2, pp. 22–30, 2017.
- [21] E. Rahmawati and E. Abdulmanan, "Permodalan Aplikasi Dunia Islam Mengaji Berbasis Android," *J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 4, no. 2, pp. 197–20, 2019.
- [22] A. D. Saputro and B. F. Kuddi, "Pembuatan Aplikasi Try Out Cat ( Computer Assisted Test ) Penerimaan Pegawai Negeri Sipil Bidang Tes Intelegensi Umum Berbasis Desktop," vol. 6, pp. 467–475, 2022.