

DOCUMENT SCANNER

Riyandi Aditya Fitrah
Universitas Buana Perjuangan
Karawang, Indonesia
if19.riyandifitrah@mhs.ubpkarawang.ac.id
085889697178

Aril Kholid Ismail
Universitas Buana Perjuangan
Karawang, Indonesia
if19.arilismail@mhs.ubpkarawang.ac.id
08986424521

Erlita Sofhianti
Universitas Buana Perjuangan
Karawang, Indonesia
if19.erlitasofhianti@gmail.com
082119885624

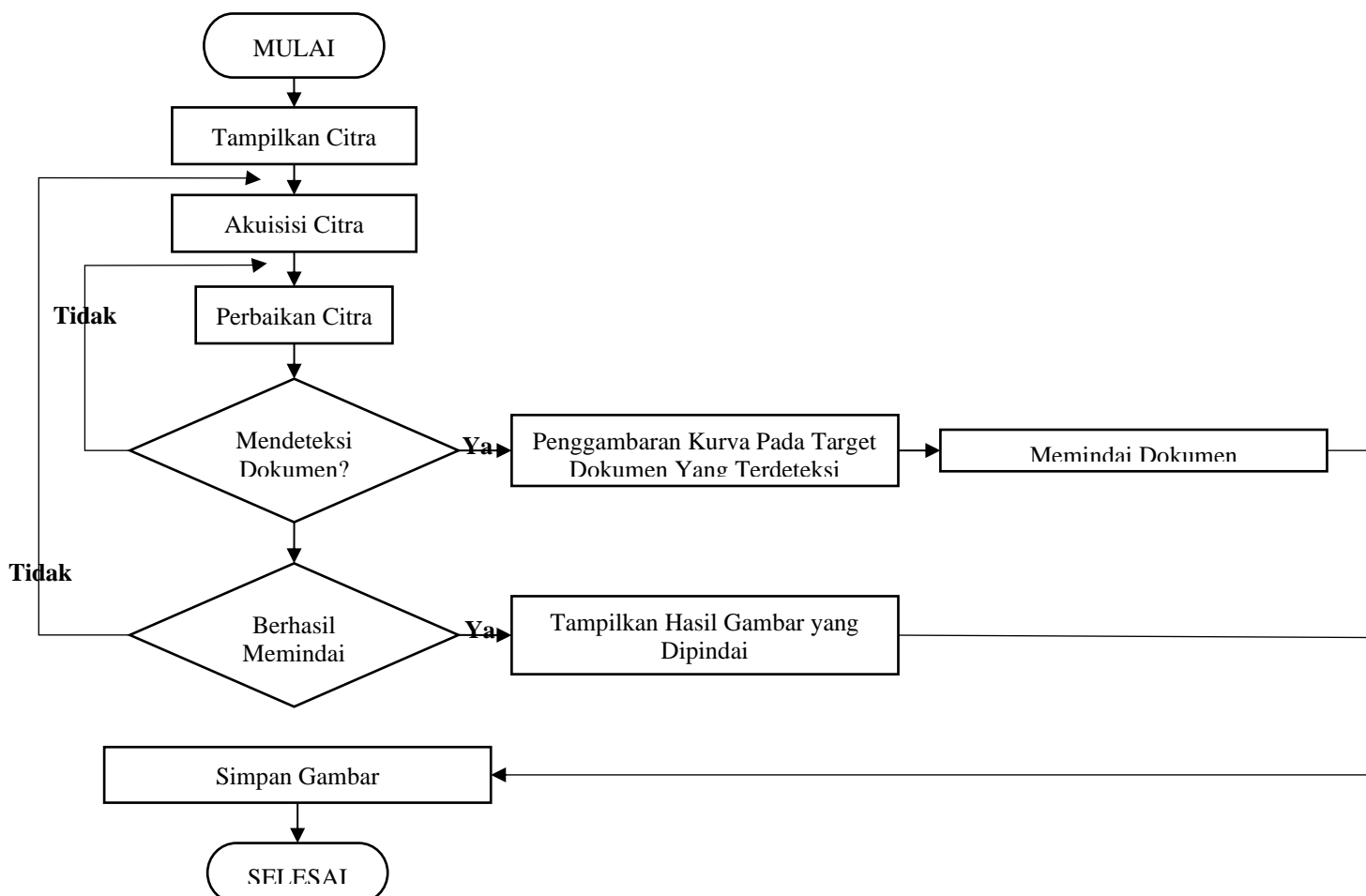
Abstract — *Document scanner* atau pemindai dokumen merupakan sebuah alat proses memindai dokumen untuk dijadikan data digital. Salah satu algoritma pengolahan citra digital yang digunakan adalah algoritma deteksi tepi (*Canny Edge Detection*). Deteksi tepi atau *Canny Edge Detection* merupakan salah satu fungsi pustaka perangkat lunak OpenCV yang digunakan untuk menentukan lokasi titik-titik tepi objek. Data yang digunakan dalam deteksi tepi berupa citra digital dari video webcam yang ditangkap. Ide yang digunakan adalah memilih bagian dari dokumen yang akan dipindai dan dapat dibuat, disalin atau disimpan dalam format gambar. Karenanya *document scanner* ini mudah untuk digunakan.

Kata kunci — *Document, scanner, OpenCV, canny edge detection.*

I. PENDAHULUAN

Document scanner atau pemindai dokumen menggunakan algoritma pendeteksi tepi yaitu *Canny Edge Detection*. Algoritma pendeteksi tepi atau *Canny Edge Detection* merupakan bagian fungsi pustaka perangkat lunak OpenCV. Ini dikembangkan oleh John F. Canny pada tahun 1986. Bahasa pemrograman yang diteliti menggunakan python, dan tool pengolahan data coding yang digunakan adalah jupyter notebook. Sistem *document scanner* atau pemindai dokumen menyediakan antarmuka yang sederhana, pengambilan, perbaikan citra, penangkapan citra, dan penyimpanan citra. Ketika sistem *document scanner* melakukan pemrosesan juga menyediakan peningkatan citra dari video webcam, analisis citra dari video webcam, dan mengubah perspektif pada video webcam. Aktivitas peningkatan citra seperti, mempertajam tepi objek dan menyesuaikan kontras. Analisis citra dari video webcam dapat mengekstrak atau mengklasifikasikan objek dalam video webcam. Mengubah perspektif seperti, memberikan poin-poin pada gambar yang ditangkap, sehingga dapat mengekstrak dari video webcam kemudian membungkus citra tersebut.

Berikut ini merupakan flowchart dari sistem *document scanner*:



II. METODE

Document scanner atau pemindai dokumen gambar dan teks menggunakan webcam dan memberikan gambar pindaian yang lebih baik sehingga dokumen yang dapat dimanfaatkan lebih lanjut sesuai kebutuhan permintaan. Sistem *document scanner* dikembangkan menggunakan python dan OpenCV sebagai perpustakaan. Metode penelitian *document scanner* lainnya menggunakan “Library Resarch”, dimana dalam penelitian ini menggunakan teori-teori yang diperoleh dari buku, jurnal, dan internet yang mendukung dan sesuai dengan laporan penelitian. Cara kerja *document scanner* atau pemindai dokumen melibatkan beberapa langkah sebagai berikut:

A. Tampilan Antarmuka

Kami menggunakan webcam NYK Nemesis A75 HEXA Full HD 1080p dengan 30FPS. Pengguna akan melihat tampilan window untuk tampilan antarmuka, dan sistem akan menangkap gambar secara auto. Komponen ini sangat berguna serta mudah digunakan.

B. Pindai Gambar Dan Teks

Langkah selanjutnya ini melibatkan 3 sub pemindaian gambar. Berikut 3 sub pemindaian yang dilakukan secara bertahap:

- Deteksi Tepi (*Canny Edge Detection*)

Untuk mendeteksi tepi citra pada video webcam digunakan algoritma *Canny Edge Detection*. Algoritma detektor tepi populer yang dikembangkan pada tahun 1986 oleh John F. Canny. Algoritma deteksi tepi *Canny* terdiri beberapa langkah:

- 1) Pengurangan Noise

Karena deteksi tepi rentan terhadap noise pada gambar, langkah pertama adalah menghilangkan noise pada gambar dengan filter sesuai yang dibutuhkan pengguna. Pengurangan noise menggunakan fungsi Gaussian Blur yang terdapat pada perpustakaan OpenCV.

- 2) Perhitungan Gradient

Perhitungan gradient dilakukan dengan mengkonversi citra menjadi grayscale, agar dapat menentukan titik tepi pada citra yang ditangkap dengan nilai intensitas yang paling besar. Gradien gambar yang telah diperhalus dengan memperkirakan gradien arah x dan y .

- 3) Suppression Non-maximum

Setelah dapatkan arah gradien dan magnitudo, pemindaian penuh dilakukan untuk menghilangkan piksel yang tidak diinginkan yang mungkin bukan merupakan bagian tepi.

- 4) Hysteresis Thresholding

Langkah ini memutuskan semua tepi benar-benar tepi dan mana yang bukan. Untuk ini kita butuh dua nilai ambang, *minimum* dan *maks*. Setiap sisi dengan intensitas gradien lebih dari *maxVal* pasti menjadi tepi dan yang dibawah *minimum* tidak menjadi bagian, sehingga dibuang. Jadi sangat penting kita harus menyesuaikan *minVal* dan *maxVal* untuk mendapatkan hasil yang benar. Tahap ini juga menghilangkan noise piksel kecil dengan asumsi bahwa tepi adalah garis panjang.

- Menemukan Kontur atau (*Finding Contours*).

Kontur adalah kurva tertutup yang menghubungkan semua titik kontinu yang memiliki beberapa warna atau intensitas, mereka mewakili bentuk benda yang ditemukan dalam citra pada video webcam. Deteksi kontur sangat berguna untuk analisis bentuk dan deteksi objek. Berikut ini mari kita ikuti agar berhasil mendeteksi kontur pada gambar atau citra:

- 1) Mengonversi citra grayscale ke *canny*, ini adalah praktik umum untuk gambar masukan menjadi citra *canny* (yang harus merupakan hasil dari gambar ambang batas atau deteksi tepi).

- 2) Menemukan kontur menggunakan fungsi `findContours()` pada library OpenCV.

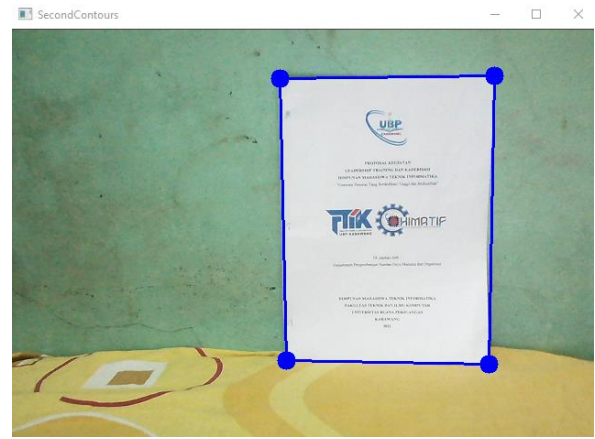
- 3) Menggambar kontur kedalam tampilan antarmuka.

- 4) Kemudian diperlukan memilih 4 titik, secara berurutan: kiri atas, kanan atas, kiri bawah, kanan bawah.

Setelah menerapkan langkah-langkah ini, Anda akan mendapatkan yang berikut ini:



Gambar 1: Kontur gambar 1



Gambar 2: Kontur gambar 2

Gambar 1 merupakan kurva yang terhubung dalam menganalisa bentuk dan deteksi objek, gambar dua merupakan kontur kedua yang dihasilkan ketika berhasil memindai sebuah objek atau dokumen yang dianalisa.



Gambar 3: Trackbar *min,max* hysteresis thresholding

Gambar ketiga merupakan trackbar yang digunakan untuk menyesuaikan nilai hysteresis *min* dan *max* agar pada saat melakukan pemindaian deteksi tepi atau ambang atas pada objek dapat disesuaikan.

C. Ekstrak Gambar Dan Teks

Langkah selanjutnya ini melibatkan 1 sub, dimana langkah ini saling terhubung dengan langkah sebelumnya yaitu pindai gambar dan teks. Berikut sub yang dilakukan:

- Transformasi Perspektif (*WarpPerspective/ Bird View*)

Transformasi Perspektif adalah operasi yang kita gunakan ketika kita ingin mengubah perspektif suatu objek. Berikut langkah melakukan transformasi perspektif. dan deteksi objek:

- 1) Pertama kita perlu memindai gambar yang ingin kita ubah
- 2) Menemukan kontur menggunakan fungsi `findContours()` pada *library* OpenCV.
- 3) Kami kemudian perlu memilih 4 titik, secara berurutan: kiri atas, kanan atas, kiri bawah, kanan bawah.
- 4) Kemudian menggambar lingkaran untuk menunjukkan titik yang tepat yang kita ambil.
- 5) Buat daftar dengan 4 poin ini, dan kami akan menggunakan daftar ini nanti untuk menerapkan transformasi.
- 6) Buat set baru 4 poin. 4 titik ini adalah ukuran jendela baru tempat kami ingin menampilkan gambar berubah.
- 7) Kemudian kami menerapkan transformasi perspektif untuk membuat matriks dan akhirnya kami dapat mengubah gambar menjadi menggunakan bingkai asli dan matriks yang baru saja dibuat.
- 8) Kami juga dapat mengatur nilai *min* dan *max* hysteresis thresholding dengan trackbar untuk penyesuaian objek dokumen.
- 9) Kemudian kita bisa menampilkannya di layar.

Setelah menerapkan langkah-langkah ini, Anda akan mendapatkan yang berikut ini:



Gambar 4: Warp Perspektif

Gambar keempat yang dihasilkan merupakan hasil dari *document scanner*, dengan menggunakan transformasi warp perspektif atau *bird view*, dapat kita lihat hasil dari warp perspektif terdapat sedikit ketidakjelasan pada gambar karena cahaya yang masuk kurang atau karena nilai *min* dan *max* pada hysteresis thresholding kurva yang terhubung kurang atau harus ditambahkan.

D. Simpan Gambar

Langkah ini merupakan langkah terakhir dari cara kerja proyek sistem *document scanner* berkaitan dengan penyimpanan gambar yang dipindai dan diekstrak agar dapat digunakan. Foto disimpan dalam format jpg, dan tersimpan secara default oleh sistem di document laptop apabila ketika sistem *document scanner* sedang berproses lalu menekan tombol 's'. Berikut hasil penerapan pada proses menyimpan gambar:


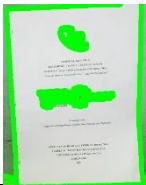



Gambar 5: Simpan Gambar

Gambar kelima merupakan proses menyimpan citra dengan menekan tombol 's' ketika sistem sedang berproses.

III. HASIL DAN PEMBAHASAN

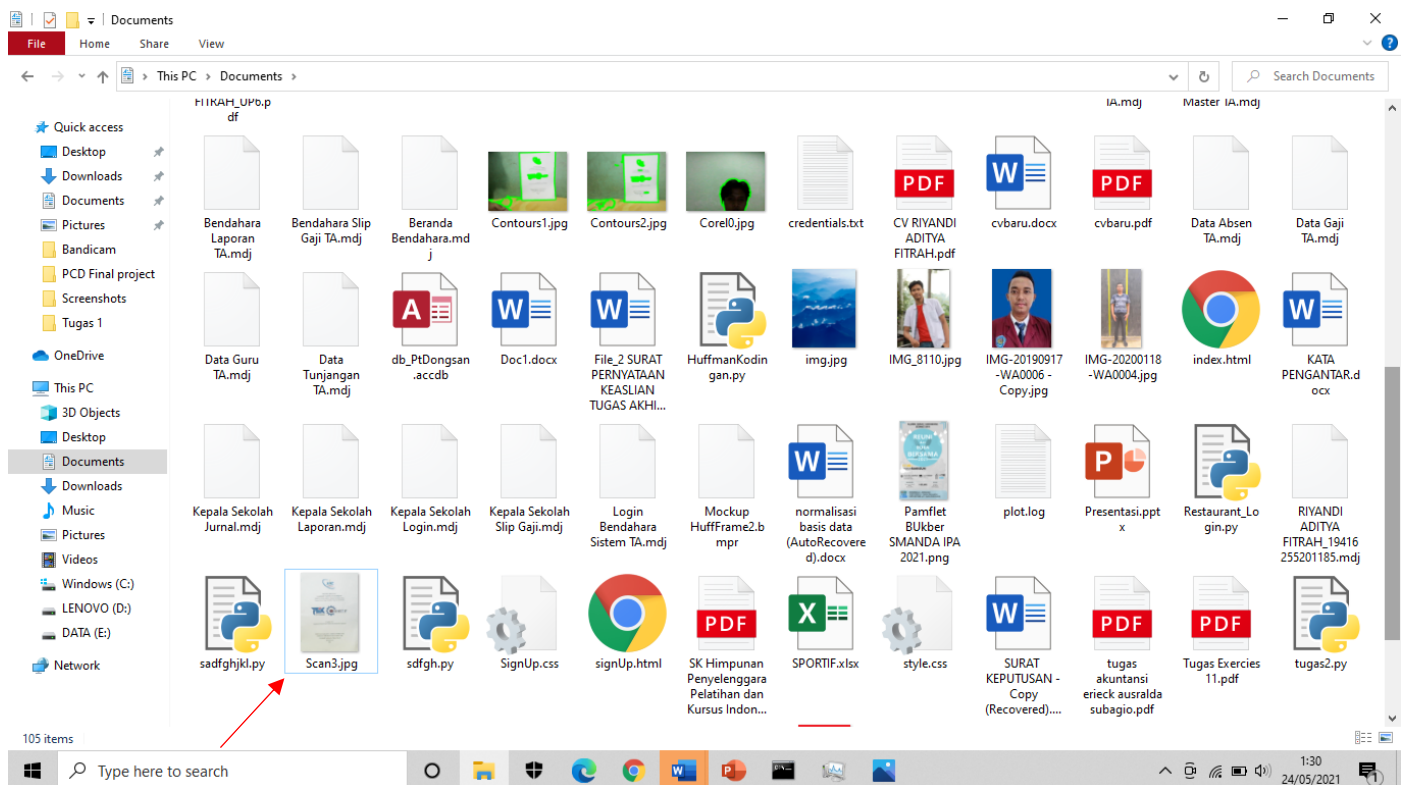
Berikut merupakan pembahasan detail mengenai citra menemukan kontur yang kami lakukan saat penelitian sistem kami dengan nilai pada *gaussian blur*, *min* dan *max* berdasarkan citra yang digunakan:

No	Citra	Nilai Pada (<i>Gaussian Blur</i>)	Nilai Minimum Dan Maximum (<i>Hysteresis Thresholding</i>)
1		$(x,y) = (7,7)$	Minimum = 4 Maximum = 255
2		$(x,y) = (7,7)$	Minimum = 20 Maximum = 220
3		$(x,y) = (7,7)$	Minimum = 70 Maximum = 200

Berdasarkan pada pembahasan penelitian, kami pun berhasil mendapatkan citra dengan hasil yang baik berdasarkan nilai-nilai tertentu. Dari cara kerja, algoritma, fungsi dan lainnya yang telah dibahas pada metode, maka dihasilkan citra yang dipindai, diekstrak, dan diimplementasikan sebagai berikut:



Gambar 6: Hasil citra yang dipindai, hasil dipindai dengan 4 titik, dan diekstrak



Gambar 7: Hasil citra yang diimpenetasikan dan disimpan

Gambar 6 merupakan tahapan pengerjaan dari *document scanner*, lalu untuk gambar 7 merupakan hasil implementasi yang disimpan secara default oleh sistem atau berdasarkan kodingan yang kami deklarasikan dengan menekan tombol keywords 's'.

IV. KESIMPULAN

Sistem *document scanner* yang diimplementasikan menggunakan python sepenuhnya dengan bantuan liblary dari OpenCV, memenuhi tujuan yang telah dikembangkannya. Pembuatan ini awalnya dimaksudkan untuk menciptakan sistem semacam aplikasi camscanner versi mobile dengan tujuan yang sama yaitu dapat menyediakan sejumlah spesifikasi yang berguna tanpa kehilangan data atau masalah lain bagi pengguna. Kini dengan adanya sistem *document scanner* yang *easy to use* dapat dijalankan di windows, kami membuat para pengguna dapat menghemat banyak waktu, biaya tambahan, dan pengoperasian dapat dilakukan secara manual. Selain itu juga, membuat tugas berbagi dan mengunggah dokumen menjadi lebih mudah dan cepat.

Saran terhadap penggunaan sistem proyek kami, ini merupakan sistem yang telah dibuat dengan banyak melakukan penelitian dan dibuat se-sederhana mungkin untuk memenuhi pengumpulan tugas proyek akhir mata kuliah Pengolahan Citra Digital. Sistem ini digunakan oleh para pengguna yang mengalami kesulitan dalam mengunggah dokumen, dan memindai dokumen. Pengakuan terhadap penelitian sistem ini, kebutuhan menyimpan citra dalam format yang tersedia hanyalah jpg dan citra yang dipindai lalu diekstrak harus memiliki masukan cahaya yang sangat banyak, selain itu juga citra yang ditangkap untuk dipindai harus tereduksi noisenya. Diharapkan kepada pengguna dapat menggunakan sistem ini dengan ramah, baik, dan benar. kami akan sebaik mungkin untuk mengembangkan sistem *document scanner* ini sehingga dapat direalisasikan.

V. DAFTAR PUSTAKA

- [1] Darma Putra. Pengolahan Citra Digital. Yogyakarta: CV. Andi Offset. 2010.
- [2] G. T. Shrivakshan and Dr. C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing", International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012.
- [3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] Bao, P., Zhang, L., Wu, X.: Canny Edge Detection Enhancement by Scale Multiplication. IEEE Trans. on PAMI 27(9), 1485–1490 (2005).
- [5] Muthukrishnan, R. and M. Radha., "Edge Detection Techniques for Image Segmentation," International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No. 6, Dec. 2011.
- [6] Zhang J. Research on the geometric distortion auto-correction algorithm for image scanned. Applied Mechanics and Materials, 2014, 3468(644): 30–44.
- [7] Nambodiri, A., Jain, A.K. (2007) "Document Structure and Layout Analysis", Digital Doc Proc: Major Dir. and Recent Adv.