

# Digitalisasi Lecturing Note

Listya Purnomo<sup>1</sup>, Liliana<sup>2</sup>, Kartika Gunadi<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: lis\_light@yahoo.com<sup>1</sup>, lilian@petra.ac.id<sup>2</sup>, kgunadi@petra.ac.id<sup>3</sup>

## ABSTRAK:

Perkembangan teknologi mempermudah manusia dalam melakukan berbagai hal. Metode pengajaran saat ini memiliki banyak variasi sehingga proses pembelajaran lebih hidup. Pembelajaran dengan penjelasan proses demi proses yang dapat terlihat sangat bermanfaat. Terkadang dalam suatu proses justru gerakan tangan memegang peranan penting untuk memahami proses secara mendalam. Penggunaan *webcam* dapat menjadi alternatif lain untuk mendigitalisasi apa yang ditulis *user*. Skripsi ini bertujuan untuk membuat digitalisasi *lecturing note* pada kertas memanfaatkan *device webcam*.

Sistem yang dibuat meliputi proses pengambilan *image*, pendeteksian kertas, *projection mapping* kertas, *skin detection* dan *skin removal*, serta pengolahan hasil goresan yang didapatkan dengan metode *threshold*. Proses pendeteksian tepi kertas menggunakan metode *canny edge detection*, dimana dari *image webcam* berubah menjadi *image* dengan *area* kertas berwarna putih. Proses *projection mapping* untuk *me-mapping image* dengan bentuk kertas yang trapesium menjadi *image* dengan bentuk kertas persegi panjang. Proses *skin detection* menggunakan ruang warna HSV dan YCbCr, dimana *image* awal yang masih menampilkan tangan berubah menjadi *image* tanpa adanya tangan. Aplikasi dibuat menggunakan bahasa pemrograman C# dengan Visual C# 2010 sebagai IDE nya.

Hasil pengujian menunjukkan warna, ketebalan tinta, ambang batas nilai *threshold*, ketinggian *webcam*, ukuran dan jarak antar goresan, serta tingkat pencahayaan akan mempengaruhi hasil akhir dari aplikasi. Selain itu, pada saat menulis dengan posisi tangan di atas kertas akan memunculkan bayangan tangan. Hasil paling bagus dari semua metode yang dicoba adalah hasil dari metode *threshold*.

**Kata Kunci:** Digitalisasi, *Lecturing*, *Webcam*, *Corner*, *Skin*, *Mapping*

## ABSTRACT:

The development of technology can make human being to do a lot of things more simple. Teaching method currently has many variations so that the learning process is more alive. Learning by explaining the process which can be seen is very beneficial. Sometimes in a process, hand gestures play an important role in understanding the whole process. The use of webcam can be another alternative to digitize what is written by users. This thesis aims to digitize lecturing notes on paper using a device webcam.

The system includes processes like image retrieval, paper detection, projection mapping, skin detection and removal, and processing results with threshold method. Paper edge detection process uses the method of canny edge detection. In paper edge detection, webcam image becomes image with white color for area of paper. Projection mapping process is used to map the image with trapezoid shape of paper into image with rectangular shape of paper. Skin detection process uses HSV and YCbCr color space. In skin detection process, hand which still appears in image becomes image without hand. The application was created using the C # programming language with Visual C # 2010 as its IDE.

The experiment results indicate that the color, ink thickness, threshold value, height of the webcam, size and distance between scratches, and lighting level will affect the final outcome of the application. In addition, the hand's position during the writing may cause a shadow on the paper. The best result of all methods which have been tried is the result of the threshold method.

**Keywords:** *Digitalization*, *Lecturing*, *Webcam*, *Corner*, *Skin*, *Mapping*

## 1. PENDAHULUAN

Perkembangan teknologi semakin mempermudah manusia dalam melakukan berbagai macam hal. Hal-hal yang dahulu dilakukan dengan banyak usaha, saat ini bisa menjadi lebih efektif. Salah satu contohnya adalah adanya internet yang memberikan individu akses menuju kekayaan informasi yang luar biasa dalam waktu singkat. Demikian pula dalam dunia edukasi. Metode pengajaran saat ini telah memiliki banyak variasi, dimana adanya teknologi LCD yang saat ini membantu proses pengajaran. Metode pengajaran yang bervariasi akan membuat proses pembelajaran lebih hidup sehingga anak-anak lebih tertarik dan lebih cepat menyerap apa yang diajarkan. Salah satu contohnya, ketika menerangkan akan terasa lebih menarik jikalau apa yang diterangkan dituangkan dalam bentuk tulisan ataupun gambar dan kemudian didigitalisasi.

Akhir-akhir ini banyak bermunculan teknologi di mana manusia bisa langsung menulis, mencoret-coret, ataupun menggambar di *tablet* atau *smartphonenya*, melalui fasilitas *stylus*. Hal itu tentu sangat praktis. Kepraktisan itu berbanding lurus dengan harga yang cukup mahal untuk mampu memperoleh teknologi tersebut, sementara tidak semua orang memiliki *budget* yang cukup untuk mampu membelinya. Selain itu, tidak semua *gadget* memiliki fasilitas *stylus*.

Untuk dapat mengatasi masalah yang telah dijabarkan di atas, dipikirkan alternatif lain yang cukup bermanfaat dalam mengefektifkan proses pengajaran. Penggunaan alat *webcam* dapat menjadi alternatif lain untuk mendigitalisasi apa yang ditulis atau digambar *user* dan menampilkannya pada layar. Digitalisasi *lecturing note* sangat menunjang proses pembelajaran. Hal ini mampu membuat proses pembelajaran terasa lebih hidup dan menarik. Output yang ditampilkan pada layar juga mendukung untuk diedit, tanpa harus mengulanginya kembali. Hal ini bermanfaat dalam menghemat waktu sekaligus memudahkan *user* sehingga proses pengajaran akan lebih efektif. Selain itu, hasil akhir juga dapat disimpan dalam bentuk file *image* sehingga tidak perlu dicatat. Hal ini juga turut meningkatkan efisiensi waktu dalam proses pengajaran.

Berawal dari permasalahan yang dijabarkan di atas, akan dibuat digitalisasi *lecturing note* pada kertas yang memanfaatkan *device webcam*.

## 2. TEORI PENUNJANG

### 2.1. Corner Detection

*Canny Edge Detector* merupakan salah satu metode yang paling umum digunakan dalam mendeteksi tepi untuk pengolahan citra [6]. Langkah-langkah untuk melakukan *canny edge detector*:

1. Pengurangan *noise*  
Pengurangan *noise* dapat dilakukan dengan menggunakan metode *gaussian filter*.
2. Menghitung panjang *gradien* dan besar sudut *gradien*

-1	0	+1
-2	0	+2
-1	0	+1

(a)

+1	+2	+1
0	0	0
-1	-2	-1

(b)

Gambar 1. (a) Kernel Dx (b) Kernel Dy

Rumus menghitung besar gradient adalah:

$$D = \sqrt{Dx(x,y)^2 + Dy(x,y)^2} \quad (1)$$

Dx adalah hasil perkalian antara piksel gambar dengan kernel pada Gambar 1(a) dibagi dengan jumlah kernel. Dy adalah hasil perkalian antara piksel gambar dengan kernel pada Gambar 1(b) dibagi dengan jumlah kernel.

Rumus untuk menghitung besar sudut *gradien* adalah:

$$\theta = \arctan\left(\frac{Dx(x,y)}{Dy(x,y)}\right) \quad (2)$$

3. *Non-Maximum Supression*  
*Non-Maximum Supression* menyatakan suatu piksel sebagai tepi (*edge*) apabila piksel tersebut memiliki besar *gradien* yang terbesar. Jika pixel (x, y) memiliki *gradien* yang paling besar dari semua piksel yang diperiksa, maka piksel tersebut merupakan tepi (*edge*).

### 4. Hysteresis Thresholding

*Hysteresis Thresholding* memiliki dua ambang batas, ambang batas bawah (*tlow*) dan ambang batas atas (*thigh*). Jika piksel (x, y) memiliki *gradien* kurang dari *tlow*, maka piksel tersebut bukan merupakan tepi (*edge*). Jika piksel (x, y) memiliki *gradien* lebih besar dari *thigh*, maka piksel tersebut merupakan tepi (*edge*). Jika piksel (x, y) memiliki *gradien* antara *tlow* dan *thigh* dan salah satu dari piksel sekeliling di daerah  $3 \times 3$  memiliki *gradien* lebih besar dari *thigh*, maka piksel tersebut merupakan tepi (*edge*).

## 2.2. Projection Mapping

*Projection mapping* adalah melakukan *mapping* geometris gambar 2D, dimana hasilnya tidak mengubah isi gambar, namun membentuk ulang *grid pixel* ke gambar yang baru. Hal ini berguna untuk memperbaiki bentuk gambar karena posisi hadap kamera, seperti yang dituturkan Li, Wong, Gong dan Chang [2].

Hal pertama yang perlu dilakukan adalah menghitung matriks *mapping* yang sesuai. Matriks *mapping* dengan ordo  $2 \times 2$  dapat diperoleh dari rumus :

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \text{map\_matrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3)$$

di mana :

$$\text{dst}(i) = (x'_i, y'_i), \text{src}(i) = (x_i, y_i)$$

*src* adalah *corner* dari *image original*, *dst* adalah *corner* dari *projection image* yang diharapkan, sedangkan *map\_matrix* adalah hasil matriks *mapping* berukuran  $2 \times 2$  yang akan digunakan pada perhitungan selanjutnya.

Setelah itu, dilakukan *mapping* pada sebuah gambar sehingga terproyeksikan pada bidang gambar lain. *Mapping* dilakukan dengan mengalikan piksel – piksel pada gambar yang akan diproyeksikan dengan matriks *mapping*, sebagai berikut:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} \\ h_{10} & h_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4)$$

$x'$ ,  $y'$  merupakan posisi piksel hasil *mapping*, matriks  $h$  merupakan matriks *mapping* yang berupa matrix  $2 \times 2$ , serta  $x$ ,  $y$  merupakan posisi piksel pada gambar *original*.

## 2.3. Skin Segmentation

*Skin segmentation* adalah proses pemisahan piksel mana dari suatu gambar yang merupakan kulit. Dua metode yang sering digunakan dalam *skin segmentation* adalah tipe warna YcbCr dan tipe warna HSV.

### 2.3.1. YCbCr

Komponen warna YcbCr terdiri dari *luminance* (pencahayaan, dihitung dari nonlinear RGB), *chrominance* (nilai Cr dan Cb) yang dihitung dengan mengurangi *luminance* dari komponen merah dan biru RGB. Warna kromatik dikenal sebagai warna murni tanpa adanya pencahayaan. Pemisahan eksplisit antara komponen *luminance* dan *chrominance* membuat ruang warna ini cocok untuk segmentasi warna kulit, seperti yang

diutarakan Sharma dan Kumar [3]. Batas nilai  $C_b$  dan  $C_r$  untuk segmentasi kulit adalah:

$$140 \leq C_r \leq 165; 140 \leq C_b \leq 195$$

### 2.3.2. HSV

HSV memberikan informasi warna *Hue* (warna dominan), *Saturation* (warna-kemurnian, seperti perbedaan antara warna merah dan *pink*) dan intensitas nilai (*value* atau warna kecerahan, seperti perbedaan antara merah gelap dan lampu merah), seperti yang diutarakan Surampalli [5].

Berdasar ruang warna ini, H dan S akan memberikan informasi yang diperlukan untuk segmentasi kulit. Batas nilai H dan S untuk segmentasi kulit adalah:

$$0 \leq H \leq 0,25; 0,15 \leq S \leq 0.9 \quad (6)$$

## 2.4. Thresholding

*Threshold* adalah metode segmentasi sederhana intensitas [1]. Untuk membedakan piksel, dilakukan perbandingan dari setiap nilai intensitas piksel dengan ambang batas yang ditentukan. Setelah itu, nilai suatu piksel bisa diberi warna hitam atau putih tergantung dari perbandingan nilai piksel itu dengan ambang batas.

### 2.4.1. Threshold Binary

Rumus dalam *threshold binary* adalah:

$$dst(x,y) = \begin{cases} 255 & \text{if } src(x,y) > \text{batas threshold} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

*src* adalah *image* awal, *dst* adalah *image* hasil *threshold*. Jadi, jika intensitas piksel lebih tinggi dari nilai batas *threshold* maka intensitas piksel baru diganti dengan *maxvalue* dan sebaliknya.

### 2.4.2. Threshold Adaptive

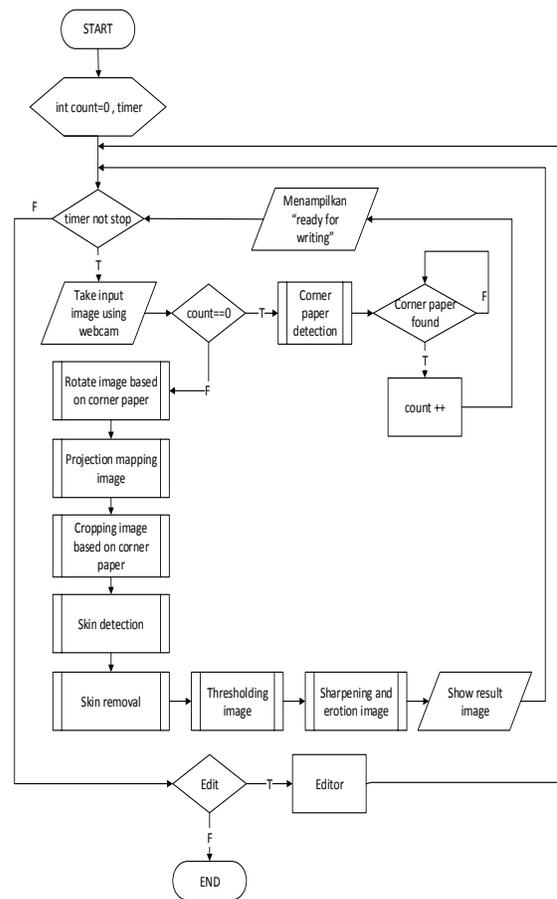
Pada *adaptive threshold*, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan *threshold* lokal atas setiap blok itu dengan nilai ambang batas yang berbeda. Nilai ambangnya ditentukan sebagai fungsi rata-rata nilai piksel di dalam daerah citra tersebut. Metode yang digunakan adalah setiap piksel akan di-*threshold* dengan ambang batas nilai yang dihitung dari nilai rata-rata piksel tetangga. Metode ini dipaparkan oleh Solaiman [4].

## 3. DESAIN SISTEM

Rancangan arsitektur dan proses kerja sistem digambarkan dalam pada Gambar 2 dan Gambar 3. Pertama *user* diminta untuk memasang *webcam* luar, menyiapkan kertas, mengatur posisi *webcam* dan kertas serta menekan tombol *start*, kemudian dilakukan proses pendeteksian tepi kertas pada program. Setiap 3 detik, akan dilakukan proses berupa pengambilan *image* dengan *webcam* kemudian dilakukan rotasi dan *projection mapping*. Setelah itu, dilakukan *cropping* dan mendeteksi tangan. Proses berikutnya adalah menajamkan (*sharpen*) dan mendilatasi *image*.

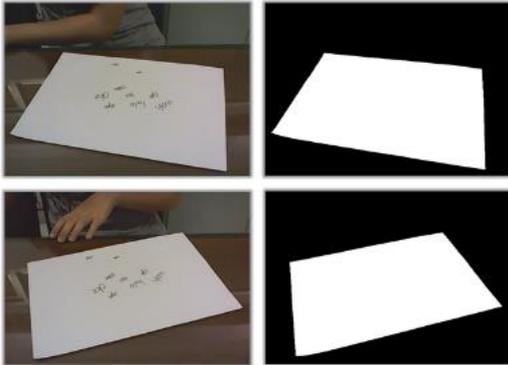


Gambar 2. Arsitektur secara garis besar



Gambar 3. Flowchart cara kerja program

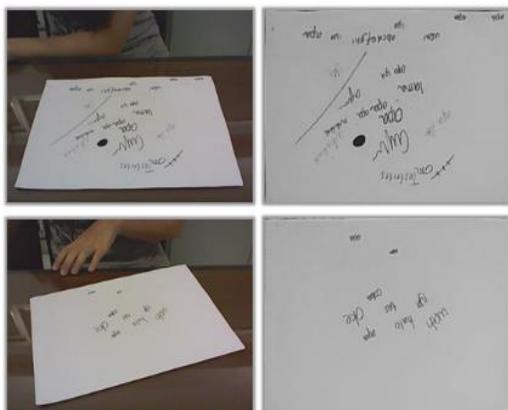
Pada proses *corner paper detection*, *image* yang diambil diolah untuk mendapatkan sudut dari kertas. Pertama-tama, akan dilakukan proses *canny edge detector*. Hasil dari fungsi ini adalah *image* di mana semua tepi dari kertas akan berwarna putih, yang selanjutnya digunakan untuk mendeteksi empat buah garis dari tepi kertas menggunakan prinsip kesamaan *gradien* dari suatu garis. Apabila *gradien* dari tepi yang terdeteksi berbeda dari *gradien* sebelumnya, maka titik sebelumnya adalah sudut kertas. Pada Gambar 4 merupakan hasil *corner paper detection*.



Gambar 4. Hasil *Corner Paper Detection*

Pada proses *rotate*, *image* yang diambil akan di-rotate agar posisi kertas lurus dan tidak miring. Dua titik sudut dengan koordinat  $y$  terbesar akan digunakan untuk menghitung besar *arctan* dari perbandingan perbedaan  $y$  dan perbedaan  $x$  untuk merotasikan *image*.

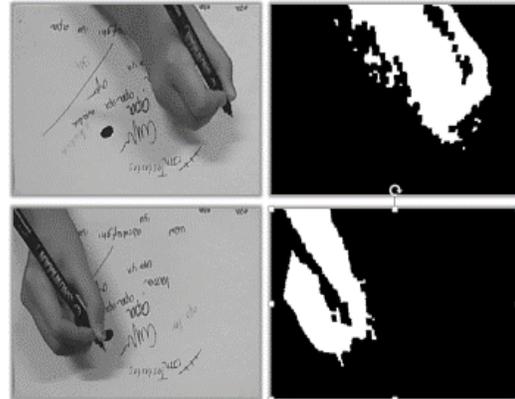
Pada proses *projection mapping*, *image* hasil dari proses rotasi sebelumnya akan diproyeksikan sehingga bentuk kertas tidak lagi trapesium tetapi menjadi berbentuk persegi panjang. Pertama-tama, akan dilakukan perhitungan matriks *mapping* yang sesuai berdasar sudut tepi asal dan sudut tepi hasil *mapping* yang diharapkan, sesuai rumus 3. Matriks *mapping* yang dihasilkan akan dikalikan dengan piksel – piksel pada *image original* untuk diperoleh piksel hasil proyeksi dan menghasilkan *image* baru hasil proyeksi, seperti pada Gambar 5.



Gambar 5. Hasil *Projection Mapping*

Pada proses *cropping*, *image* akan di-*crop* hanya pada *area* kertas saja, sehingga *area* di sekeliling kertas akan hilang. Pertama-tama, akan ditentukan *Region Of Interest* dari proses *cropping*. *Image* dengan piksel yang terletak dalam *Region Of Interest* akan disalin ke sebuah *image* yang baru.

Pada proses *skin detection*, *image* hasil *cropping* akan dideteksi bagian mana yang merupakan *skin* (dalam hal ini tangan). Metode yang digunakan adalah mengubah *image* ke dalam ruang warna YCbCr dan HSV, kemudian memeriksa setiap piksel apakah piksel tersebut berada dalam batas Cb, Cr, H, dan S yang telah ditentukan, sesuai rumus 5 dan 6, seperti pada Gambar 6.



Gambar 6. Hasil *Skin Detection*

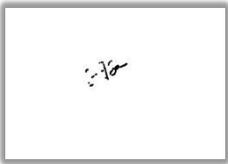
Apabila suatu piksel terdeteksi sebagai tangan, maka piksel tersebut akan diganti oleh piksel dari *image timer* sebelumnya. Selanjutnya pada proses *threshold* dilakukan perbandingan dari setiap nilai intensitas piksel dengan ambang batas yang ditentukan. Jika menggunakan *threshold binary*, apabila intensitas piksel lebih tinggi dari nilai *threshold* maka intensitas piksel baru diganti dengan 255 dan sebaliknya. Jika menggunakan *threshold adaptive*, setiap piksel akan di-*threshold* dengan ambang batas nilai yang dihitung dari nilai rata-rata piksel tetangga. Hasil proses *thresholding* tersebut adalah *monochrome image*. Selanjutnya untuk proses *sharpening* dan dilasi menggunakan konvolusi matriks atau sering disebut sebagai kernel.

#### 4. PENGUJIAN

Pengujian terhadap aplikasi dibagi menjadi beberapa bagian diantaranya adalah sebagai berikut:

- Pengujian terhadap ketebalan dan warna tinta. Pengujian dilakukan pada berbagai macam warna dan ketebalan tinta dengan nilai *threshold* 128. Pengujian ini dilakukan untuk mengetahui seberapa jauh aplikasi dapat menangani kasus/permasalahan untuk warna tinta yang tidak hitam ataupun ketebalan tinta pada nilai *threshold* tertentu.

**Tabel 1. Pengujian terhadap ketebalan dan warna tinta**

No	Gambar awal	Hasil
1. (spidol hitam)		
2. (spidol biru)		
3. (spidol merah)		
4. (bolpoin hitam)		
5. (bolpoin biru)		
6. (bolpoin merah)		

Pada Tabel 1 nomor 1 dapat dilihat bahwa warna tinta hitam akan tampak sangat jelas pada hasil. Sedangkan pada Tabel 1 nomor 2-3 dapat dilihat bahwa warna tinta biru dan merah tampak samar-samar atau bahkan tidak tampak sama sekali pada hasil. Hal ini dikarenakan warna hitam memiliki kontras warna *red*, *green*, dan *blue* yang besar dengan kertas. Demikian pula, ketebalan tinta sangat mempengaruhi kejelasan hasil goresan. Pada

Tabel 1 nomor 1-3 dapat dilihat bahwa ketebalan tinta dengan jenis spidol masih tampak pada hasil, walaupun pada nomor 2 dan 3 tampak samar. Sedangkan pada Tabel 1 nomor 4-6 dapat dilihat ketebalan tinta dengan jenis bolpoin tidak tampak sama sekali pada hasil. Hal ini dikarenakan semakin tebal tinta akan semakin tampak jelas pada *image* (tidak kabur).

Melalui pengujian ini dapat disimpulkan warna dan ketebalan tinta mempengaruhi hasil akhir dari aplikasi. Semakin hitam dan tebal suatu tinta akan semakin jelas pula hasilnya. Tinta warna hitam dengan ketebalan spidol akan tampak jelas pada hasil dengan batas nilai *threshold* 128. Tinta warna selain hitam dengan ketebalan spidol akan tampak samar-samar pada hasil dengan batas nilai *threshold* 128.

- Pengujian pengaruh *threshold* terhadap bayangan. Posisi tangan *user* yang sedang menggores di atas kertas akan menimbulkan bayangan jika terkena cahaya lampu. Pengujian dilakukan pada berbagai nilai *threshold*, yaitu nilai 112, 128, dan 144. Pengujian ini dilakukan untuk mengetahui pengaruh nilai *threshold* dalam mengatasi masalah bayangan tangan.

**Tabel 2. Pengujian pengaruh *threshold* terhadap bayangan**

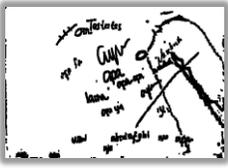
No	Gambar awal	Hasil
1. ( <i>threshold</i> 112)		
2. ( <i>threshold</i> 128)		
3. ( <i>threshold</i> 144)		

Pada Tabel 2 nomor 1 dengan nilai *threshold* 112 dapat dilihat bahwa bayangan tangan tampak beberapa bagian pada hasil. Pada Tabel 2 nomor 2 dengan nilai *threshold* 128 dapat dilihat bahwa bayangan tangan tampak semakin besar pada hasil. Pada Tabel 2 nomor 3 dengan nilai *threshold* 144 dapat dilihat bahwa bayangan tangan tampak sangat besar pada hasil dan hal itu sangat

mengganggu. Hal ini dikarenakan semakin tinggi batas nilai *threshold* semakin besar nilai toleransi suatu piksel diklasifikasikan sebagai goresan. Dari pengujian ini dapat disimpulkan bahwa batas nilai *threshold* 112 sudah cukup bagus untuk mengatasi masalah bayangan tangan, sedangkan batas nilai *threshold* 144 semakin tidak mampu dalam mengatasi bayangan tangan. Batas nilai *threshold* yang sejauh ini baik untuk kejelasan goresan sekaligus mengatasi faktor bayangan adalah nilai 128.

- Pengujian perbandingan hasil *threshold* biasa dan *threshold adaptive*.  
*Threshold* biasa adalah *threshold* dengan ambang batas nilai yang berlaku untuk semua piksel dalam satu *image* secara keseluruhan. Sementara, *threshold adaptive* menghitung ambang batas untuk setiap bagian daerah dari gambar. Pengujian dilakukan untuk membandingkan kualitas hasil dari metode *threshold* biasa dengan hasil dari metode *threshold adaptive*.

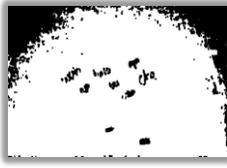
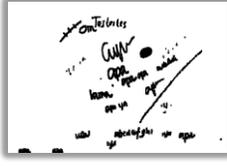
**Tabel 3. Pengujian perbandingan hasil *threshold* biasa dan *threshold adaptive***

No	<i>Threshold Binary</i>	<i>Threshold Adaptive</i>
1.		
2.		

Pada Tabel 3 merupakan hasil *threshold* biasa dan *threshold adaptive*. Hasil dengan *threshold adaptive* lebih bagus daripada hasil dengan *threshold* biasa, seperti pada gambar nomor 2. Sedangkan, hasil dengan *threshold* biasa lebih bagus daripada hasil dengan *threshold adaptive*, seperti pada gambar nomor 1. Dari Tabel 3 dapat dilihat bahwa bayangan yang tampak pada *threshold* biasa lebih mengisi, sedangkan bayangan yang tampak pada *threshold adaptive* berupa tepian dari bayangan. Dari pengujian ini dapat disimpulkan bahwa terkadang hasil dari *threshold* biasa lebih bagus daripada hasil dari *threshold adaptive* dan sebaliknya pula.

- Pengujian terhadap pengaruh pencahayaan.  
 Pengujian dilakukan pada ruangan kelas dan juga ruangan laboratorium dikarenakan tingkat pencahayaan masing-masing ruang berbeda. Pada tiap ruangan yang digunakan dalam pengujian, banyaknya lampu yang dinyalakan juga diuji mulai dari semua lampu dalam kondisi hidup hingga semua lampu dalam kondisi mati.

**Tabel 4. Pengujian terhadap pengaruh pencahayaan**

No	Gambar awal	Hasil
1. (kelas, 2 buah saklar hidup)		
2. (kelas, 1 buah saklar hidup)		
3. (kelas, semua lampu mati)		
4. (lab, 2 buah saklar hidup)		
5. (lab, 1 buah saklar hidup)		
6. (lab, semua lampu mati)		

Pada Tabel 4 dapat dilihat bahwa hasil aplikasi pada kondisi satu buah saklar menyala memiliki bayangan yang tampak lebih besar daripada hasil aplikasi pada kondisi dua buah saklar menyala. Sedangkan apabila semua saklar lampu dimatikan, hasil aplikasi menjadi banyak bagian yang berwarna hitam (dapat dilihat pada gambar nomor 5-6). Pada Tabel 4 dapat dilihat bahwa hasil aplikasi pada kondisi satu buah saklar menyala (nomor 3-4) tampak lebih putus-putus dan tidak jelas daripada hasil aplikasi pada kondisi dua buah saklar menyala (nomor 1-2). Sedangkan apabila semua saklar lampu dimatikan, sebagian besar dari goresan justru tidak tampak pada hasil (dapat dilihat pada gambar nomor 5-6).

Dari pengujian ini dapat disimpulkan bahwa tingkat pencahayaan turut mempengaruhi hasil aplikasi. Pada ruang kelas dengan kondisi 14x50 watt lampu menyala atau pada ruang laboratorium dengan kondisi 12x50 watt lampu menyala, hasil aplikasi cukup baik.

## 5. KESIMPULAN

Berdasarkan hasil pengujian didapatkan beberapa kesimpulan yaitu:

- Proses *corner detection paper* berhasil apabila semua *area* kertas masuk dalam *image* serta ada perbedaan warna yang kontras antara kertas dengan alas dengan posisi kertas condong ke arah manapun.
- Warna dan ketebalan tinta beserta ambang batas nilai *threshold* akan mempengaruhi hasil akhir dari aplikasi. Semakin hitam dan tebal suatu tinta serta semakin tinggi batas nilai *threshold* akan semakin jelas pula hasilnya. Tinta yang terbaik untuk hasil aplikasi adalah tinta warna hitam dengan ketebalan spidol.
- Pada saat menulis dengan posisi tangan di atas kertas akan memunculkan bayangan tangan. Semakin tinggi batas *threshold* yang digunakan, semakin besar bayangan tangan yang muncul. Batas nilai *threshold* yang sejauh ini baik untuk kejelasan goresan sekaligus faktor bayangan adalah 128.
- Hasil dari *threshold binary* biasa menampilkan bayangan sebagian bayangan tangan yang lebih bersifat *blocky*, sedangkan hasil dari *threshold adaptive* menampilkan semua tepian dari bayangan tangan.

- Tingkat pencahayaan turut mempengaruhi hasil aplikasi. Semakin terang suatu ruangan akan semakin baik hasilnya, sedangkan semakin gelap ruangan semakin tidak jelas hasilnya.

## 6. REFERENSI

- [1] Kusumanto, R.D. dan Tompunu, A.N. 2011. *Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB*. Semantik 2011.
- [2] Li, Z., Wong, K.H., Gong, Y. dan Chang M.Y. 2011. *A Method for Movable Projector Keystone Correction*. IEEE Transaction on Multimedia, Vol. 13, No. 1, February 2011.
- [3] Sharma, L. dan Kumar, M. 2013. *A Morphology based Approach for Human Skin Detection in Color Image*. Journal of Pure and Applied Science & Technology, Vol. 3, pp. 44-49, July 2013.
- [4] Soelaiman, R. 2010. *Segmentasi Citra Menggunakan Metode Kontur Aktif Dengan Segmentasi Lokal Atau Global Secara Selektif*.
- [5] Surampalli, G.P. 2012. *An Analysis of Skin Pixel Detection using Diffrenet Skin Color Extraction Techniques*. International Journal of Computer Applications, Vol. 54, No. 17, September 2012.
- [6] Zhou, P., Ye, W., Xia Y., & Wang, Q. 2011. *An Improved Canny Algorithm for Edge Detection*. Journal of Computational Information Systems, 7(5): 1516-1523, May 2011.