

Image Enhancing Menggunakan Contrast Stretch dengan Pembatas Berbasis Sobel Edge Detection

¹Muhammad Irwan Yanwari

¹Jurusan Teknik Elektro, Politeknik Negeri Semarang
E-mail : ¹irwan.yanwari@polines.ac.id

Abstrak

Tingginya atusias masyarakat terhadap mengabadikan momen hidupnya terkadang terkendala dengan kualitas kamera yang ada di tangannya. Hal inilah yang membuat banyaknya bermunculan algoritma untuk meningkatkan kualitas gambar, sehingga momen yang ditangkap memiliki kualitas yang baik. Contrast Stretch adalah salah satu metode untuk memperbaiki gambar, namun terkadang hasilnya berlebihan pada salah satu sisi gambar. Untuk menanggulangi hal tersebut, dilakukan pembatasan perubahan nilai kontras dengan menggunakan metode Sobel Edge Detection. Proses yang dilakukan pada penggabungan metode deteksi tepi sobel dengan contrast stretching terdiri dari empat tahapan. Pada tahapan pertama, dilakukan konversi gambar menjadi grayscale untuk mempermudah dalam proses penentuan tepian gambar menggunakan operator sobel yang dilakukan pada tahapan selanjutnya. Selanjutnya dilakukan pemrosesan gambar original menggunakan contrast stretching sehingga dihasilkan kontras yang lebih baik. Terakhir dilakukan penggabungan gambar dimana hasil contrast stretching dibatasi menggunakan hasil deteksi tepi dimana hasil deteksi tepi akan menentukan seberapa besar pengaruh gambar yang telah diproses menggunakan contrast stretching terhadap gambar original. Berdasarkan hasil kuesioner yang dilakukan, 50.54% dari responden lebih menyukai gambar hasil olahan menggunakan metode yang digunakan dibandingkan gambar original.

Kata kunci : Kualitas Gambar, Gambar, Contrast Stretch, Sobel Edge Detection

Abstract

The public enthusiast for capturing a moment of their life is sometimes constrained by the quality of their camera. This is the reason a lot of algorithms to improve an image quality emerge so that the captured image has good quality. Contrast Stretch is one of those methods to enhance the image quality, but sometimes the results are exaggerated on one side of an image. To handle this problem, something to constrain the change is needed. In this case, that something is the Sobel Edge Detection method. In this case, the Sobel operator is used to restricting the change of contrast values. The process carried out in combining the edge detection method with contrast stretching consists of four stages. In the first stage, the image is converted into grayscale to make it easier to determine the edges of the image using the Sobel operator which is carried out in the next stage. then, the original image processed using contrast stretching to find better contrast for the image. Finally, the image will be merged using the result of edge detection by limiting the results of contrast stretching using the results of edge detection. Based on the results of the questionnaire, 50.54% of respondents prefer the image processed using this method compared to the original image

Keywords : Image Quality, Image, Contrast Stretch, Sobel Edge Detection

I. PENDAHULUAN

(Saat ini perkembangan pengolahan gambar telah berkembang dengan pesat. Setiap tahunnya berbagai perusahaan teknologi yang berkaitan dengan kamera, mulai dari *smartphone* hingga perangkat khusus penangkap gambar seperti kamera *mirrorless* memperkenalkan produk terbarunya. Sebagai contoh adalah perusahaan Samsung yang memperkenalkan jajaran *smartphone flagship* baru setiap tahunnya. Dalam

event yang diadakan oleh pihak samsung empat tahun terakhir (*Galaxy Unpacked* 2018-2021), fitur kamera selalu ditonjolkan kepada calon pengguna *smartphone* yang ditawarkan, mulai dari penambahan kamera, hingga peningkatan hasil penangkapan gambar oleh kamera yang terpasang. Fitur kamera untuk keperluan *video call* pada *notebook* juga ikut berkembang, pada tahun 2019 dalam acara Microsoft Surface 2019, microsoft memperkenalkan pemrosesan gambar untuk *video call* dimana komputer dapat

menyesuaikan posisi mata agar interaksi *video call* terasa lebih natural karena orang-orang berinteraksi akan saling menatap mata satu sama lain selayaknya ketika mereka berbicara secara tatap muka.

Untuk menghasilkan kualitas gambar tersebut, selain kualitas kamera yang baik, tentunya algoritma pengolahan gambar ikut berperan dalam meningkatkan kualitas gambar. Mulai dari algoritma untuk penggabungan gambar [1], *zoom* [2], *sharpening* [3], *blur* [4], hingga perubahan posisi komponen gambar (fitur microsoft surface pro x). Munculnya berbagai algoritma tersebut tidak terlepas dari keinginan untuk mengabadikan gambar dalam kualitas yang terbaik.

1.1 Contrast Stretching

Kontras suatu gambar adalah distribusi pixel terang dan gelap pada sebuah gambar. Citra dikelompokkan ke dalam tiga kategori kontras: citra dengan tingkat kontras-rendah (*low contrast*), citra dengan tingkat kontras-bagus (*good contrast* atau *normal contrast*), dan citra dengan tingkat kontras-tinggi (*high contrast*) [5].

Contrast stretching adalah salah satu metode untuk meningkatkan kualitas gambar [6]. *Contrast stretching* berfokus pada peningkatan nilai kontras pada gambar dengan merenggangkan nilai intensitas gambar [7]. Kontras gambar yang rendah dapat disebabkan oleh kurangnya pencahayaan, kurangnya bidang dinamika pada sensor citra, atau kesalahan pengaturan pada pembuka lensa saat penangkapan gambar [5].

1.2 Sobel Edge Detection

Kegiatan yang ditujukan untuk mengetahui tepi dari sebuah image agar mudah diinterpretasi oleh manusia dan juga mesin dapat dilakukan dalam pengolahan citra. Pada pengolahan citra, kegiatan tersebut disebut dengan deteksi tepi. Deteksi tepi merupakan operasi yang digunakan untuk menentukan lokasi titik-titik yang merupakan tepi obyek citra [8]. *Sobel Edge Detection* merupakan salah satu metode untuk melakukan deteksi tepi. Operator Sobel melakukan pengukuran gradien spasial 2-D pada objek dan menggambarkan daerah sesuai dengan frekuensinya [9].

II. METODE PENELITIAN

Proses yang dilakukan pada penggabungan metode deteksi tepi sobel dengan contrast stretching terdiri dari empat tahapan. Pada tahapan pertama, dilakukan konversi gambar menjadi

grayscale untuk mempermudah dalam proses penentuan tepian gambar menggunakan operator sobel yang dilakukan pada tahapan selanjutnya. Selanjutnya dilakukan pemrosesan gambar original menggunakan contrast stretching sehingga dihasilkan kontras yang lebih baik. Terakhir dilakukan penggabungan gambar dimana hasil *contrast stretching* dibatasi menggunakan hasil deteksi tepi dimana hasil deteksi tepi akan menentukan seberapa besar pengaruh gambar yang telah diproses menggunakan *contrast stretching* terhadap gambar original.

2.1 Grayscale

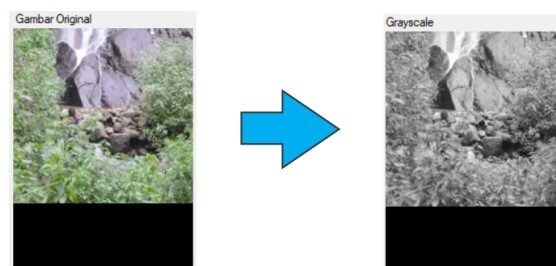
Tahapan pertama dalam pemrosesan gambar adalah membuat versi *grayscale* dari gambar yang diberikan. Adapun rumus yang digunakan dalam pembuatan *grayscale* adalah sebagai berikut :

$$Gray Level = \frac{(R + G + B)}{3}$$

R = nilai *Red* (merah), G = Nilai *Green* (hijau), B = Nilai *Blue* (biru)

Pemrosesan *grayscale* dilakukan secara berulang terhadap setiap pixel pada gambar. Berikut adalah contoh hasil perhitungan untuk 1 pixel pada gambar 1.

R	G	B	Gray
55	80	40	58



Gambar 1 Contoh hasil pemrosesan *grayscale*

Dengan nilai R = 55, G = 80, dan B = 40 didapatkan nilai Gray Level = 58.

2.2 Sobel Edge Detection

Sobel edge detection mendeteksi tepi dengan membagi proses pemeriksaan pada dua axis, yaitu horizontal (Gy) dan vertical (Gx). Adapun proses penentuan nilai dari Gx akan menggunakan matrix berikut :

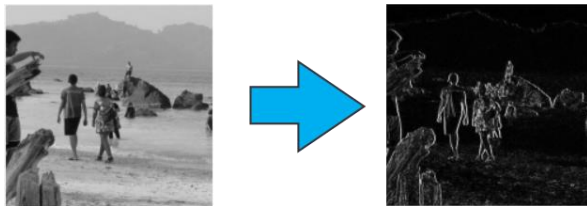
-1	0	1
-2	0	2
-1	0	1

Sebagai Contoh

50	50	100	100	-1	0	1
50	50	100	100	-2	0	2
50	50	100	100	-1	0	1
50	50	100	100			

Perhitungan untuk pixel aktif (ditandai background gray) adalah sebagai berikut :

$$G_x = (50 * -1) + (50 * -2) + (50 * -1) + (100 * 1) + (100 * 2) + (100 * 1) = 200$$



Gambar 2 Hasil pengolahan Gx

Selanjutnya proses yang sama dilakukan menggunakan matrix yang berbeda untuk melakukan perhitungan Gy. Adapun matrix yang akan digunakan untuk menghitung nilai Gy adalah sebagai berikut :

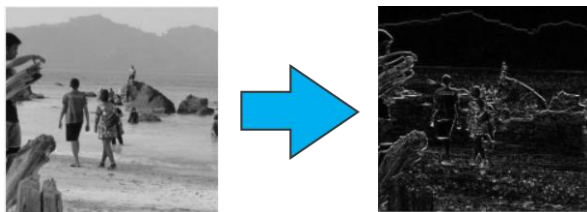
-1	-2	-1
0	0	0
1	2	1

Sebagai Contoh

50	50	100	100	-1	-2	-1
50	50	100	100	0	0	0
50	50	100	100	1	2	1
50	50	100	100			

Perhitungan untuk pixel aktif (ditandai background gray) adalah sebagai berikut :

$$G_y = (50 * -1) + (50 * -2) + (100 * -1) + (50 * 1) + (50 * 2) + (100 * 1) = 0$$



Gambar 3 Hasil Pengolahan Gy

Setelah pemeriksaan Gx dan Gy dilakukan, hasil akhir *sobel edge detection* dapat diperoleh dengan menggunakan rumus berikut :

$$G = \sqrt{G_x^2 + G_y^2}$$

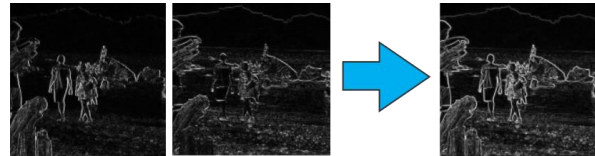
Sebagai contoh, nilai Gx dan Gy dari perhitungan sebelumnya adalah sebagai berikut :

$$G_x = 200$$

$$G_y = 0$$

Sehingga didapatkan :

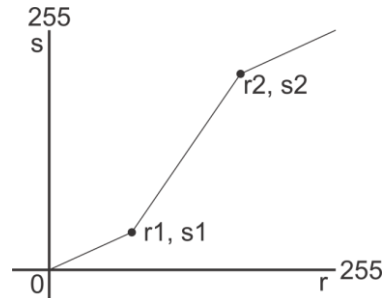
$$G = \sqrt{100^2 + 0^2} = 100$$



Gambar 4 Hasil Akhir Pengolahan *Sobel Edge Detection*

2.3 Contrast Stretching

Dalam proses *Contrast Stretching*, kondisi awal gambar dapat diasumsikan bahwa gambar memiliki range *gray level* antara 0 sampai 255. Dengan mengasumsikan bahwa r adalah citra sebelum proses dan s adalah citra setelah proses, maka transformasi citra dapat digambarkan sebagaimana kurva dibawah.



Gambar 5 Kurva transformasi contrast stretching

Secara matematis, transformasi *contrast stretching* dapat dirumuskan dengan formula berikut :

$$y - y_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} (x - x_1)$$

pada kasus $0 < r \leq r_1$, maka nilai $y = s$, $x = r$, $x_2 = r_1$, $y_2 = s_1$, $y_1 = 0$ dan $x_1 = 0$, sehingga dihasilkan rumus berikut :

$$s - 0 = \frac{(s_1 - 0)}{(r_1 - 0)} (r - 0)$$

s_1 , r_1 , dan r mengacu pada Gambar 5.

Dan dapat disederhanakan menjadi :

$$s = \frac{(s1)}{(r1)}(r)$$

s1, r1, dan r mengacu pada Gambar 5.

pada kasus $r1 < r \leq r2$, maka nilai $y = s$, $x = r$, $x2 = r2$, $y2 = s2$, $y1 = s1$ dan $x1 = r1$, sehingga dihasilkan rumus berikut :

$$s - s1 = \frac{(s2 - s1)}{(r2 - r1)}(r - r1)$$

s1, r1, s2, r2, dan r mengacu pada Gambar 5.

Dan dapat disederhanakan menjadi :

$$s = \frac{(s2 - s1)(r - r1)}{(r2 - r1)} + s1$$

s1, r1, s2, r2, dan r mengacu pada Gambar 5.

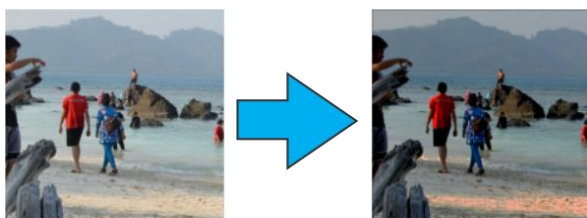
pada kasus $r2 < r \leq 255$, maka nilai $y = s$, $x = r$, $x2 = 255$, $y2 = 255$, $y1 = s2$ dan $x1 = r2$, sehingga dihasilkan rumus berikut :

$$s - s2 = \frac{(255 - s2)}{(255 - r2)}(r - r2)$$

Dan dapat disederhanakan menjadi :

$$s = \frac{(255 - s2)(r - r2)}{(255 - r2)} + s2$$

Dengan nilai r1, s1, r2, dan s2 yang dapat disesuaikan dengan konfigurasi pemilik gambar, maka dapat dihasilkan berbagai *output* gambar dengan sebaran kontras yang berbeda. Sebagai contoh, dengan $r1 = 40$, $s1 = 0$, $r2 = 220$, dan $s2 = 255$, maka hasil *output* akan menjadi sebagaimana gambar dibawah.

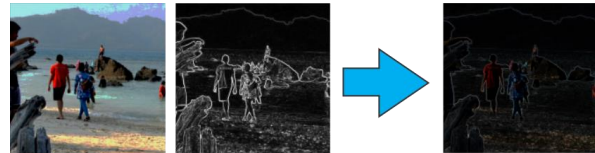


Gambar 6 Contoh Hasil Contrast Stretching

2.4 Penggabungan Sobel Edge Detection dengan Contrast Stretching

Pada penggabungan kedua metode ini, *sobel edge detection* berfungsi untuk menentukan bagian mana yang akan dipengaruhi oleh *contrast stretching* dan seberapa besar *contrast* akan

diterapkan dengan menggunakan *noise* pada hasil *edge detection*. Dengan menggabungkan kedua metode ini, *contrast* pada garis tepi setiap objek pada gambar akan semakin tinggi, sehingga objek akan semakin menonjol pada gambar yang dihasilkan.



Gambar 7 Hasil *Contrast Stretching* yang akan diimplementasikan pada gambar

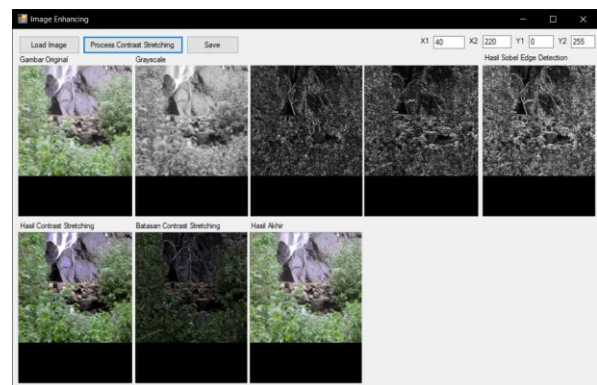
Pada tahapan akhir proses, dilakukan penggabungan antara *output* gabungan *contrast stretching* dengan gambar *input*. Adapun contoh dari hasil penggabungan dapat dilihat pada gambar dibawah.



Gambar 8 Hasil akhir gambar

III. HASIL DAN PEMBAHASAN

Untuk mempermudah proses pengujian, dilakukan implementasi algoritma kedalam bentuk program berbasis desktop.



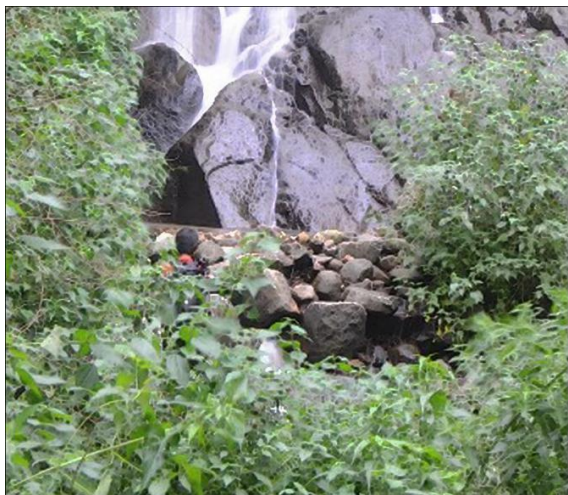
Gambar 9 Program yang digunakan dalam implementasi algoritma

Hasil implementasi pemrosesan gambar dilakukan kepada beberapa foto dengan resolusi rendah. Salah satu contoh gambar yang digunakan adalah sebagai berikut.



Gambar 10 Gambar Sebelum Diproses

Setelah dilakukan pemrosesan terdapat perubahan dimana detail dari gambar muncul.



Gambar 11 Gambar Setelah Diproses

Setelah pemrosesan selesai dilakukan, file hasil pemrosesan disimpan pada komputer yang menjalankan program.

Proses pengujian dan validasi dilakukan dengan menggunakan kuesioner yang berisikan pilihan gambar *original* dan gambar hasil olahan. Adapun konfigurasi yang digunakan dalam proses *contrast stretching* adalah *full contrast* dimana nilai $r1 = 40$, $r2 = 220$, $s1 = 0$, $s2 = 255$. Dari kuesioner yang disebar, terdapat 19 responden yang bersedia untuk mengikuti kuesioner yang disiapkan. Berdasarkan hasil pembagian kuesioner pada 19 responden, didapatkan bahwa 50.54% responden menyukai gambar hasil olahan dibandingkan gambar *original*. Hasil kuesioner dengan konfigurasi *contrast* yang diberikan tergolong bagus dikarenakan responden yang menyatakan gambar hasil olahan lebih baik dari gambar *original* melebihi 50%. Hal menarik yang

terjadi pada hasil umpan balik dari responden adalah pada gambar pantai dimana gambar olahan dapat memunculkan detail objek lebih baik dari gambar *original*, namun responden lebih cenderung menyukai gambar *original* dibandingkan gambar hasil olahan. Berdasarkan hasil tersebut terdapat *variable* yang tidak dapat dijelaskan pada penelitian ini, yaitu pada kondisi apa responden menyukai detail gambar, konfigurasi yang optimal berdasarkan preferensi responden, dan kapan responden menginginkan gambar dengan detail yang rendah.

IV. KESIMPULAN

Berdasarkan hasil implementasi, validasi, dan pembahasan yang dilakukan, maka dapat disimpulkan sebagai berikut.

1. 50.54% dari responden yang mengikuti survey menyukai gambar hasil olahan.
2. Gambar pemandangan dengan detail hasil olahan algoritma memiliki nilai respon positif tertinggi dengan nilai 68.4%.

DAFTAR PUSTAKA

- [1] M. V. S. Sakharkar, "Image Stitching Techniques-An overview," *Int. J. Comput. Sci. Appl.*, vol. 6, no. 2, pp. 324–330, 2013.
- [2] S. M. Marpaung and G. Ginting, "PERANCANGAN APLIKASI IMAGE ZOOMING DENGAN ALGORITMA K-SPACE TRANSFORMATION," *J. Pelita Inform.*, vol. 6, no. 2, pp. 175–179, 2017.
- [3] G. Mahesh, K. V. S. S. Murthy, and J. Rajanikanth, "A NOVEL APPROACH FOR SHARPENING BLUR IMAGE USING CONVOLUTIONAL NEURAL NETWORKS," *J. Crit. Rev.*, vol. 7, no. 7, pp. 139–148, 2020.
- [4] R. S. Ganesh, J. Babu, N. B. Prasad, M. Jagadeeshwar, T. Devi, and A. Sumanth, "Blurring an Image using Python," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 9, pp. 243–245, 2019, doi: 10.35940/ijitee.I1139.0789S419.
- [5] J. Teknovasi, P. Sihombing, M. Ramli, M. T. Informatika, U. S. Utara, and A. Euclidean, "ANALISIS CONTRAST STRETCHING MENGGUNAKAN ALGORITMA EUCLIDEAN UNTUK MENINGKATKAN," *J. Teknovasi*, vol. 03, no. 2013, pp. 26–38, 2016.
- [6] I. M. Dwi, P. Asana, I. M. O. Widyantara, N. Wirastuti, I. Bagus, and P. Adnyana, "Metode Contrast Stretching untuk

- Perbaikan Kualitas Citra pada Proses Segmentasi Video,” *Teknol. Elektro*, vol. 16, no. 02, pp. 1–6, 2017.
- [7] S. S. Al-amri, N. V Kalyankar, and S. D. Khamitkar, “Linear and Non-linear Contrast Enhancement Image,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 2, pp. 139–143, 2010.
- [8] I. Hastuti and P. N. Banjarmasin, “PERBANDINGAN METODE DETEKSI TEPI MENGGUNAKAN,” *Semin. Nas. Ris. Terap.*, vol. 5662, no. Nopember 2016, pp. A129–A137, 2016.
- [9] M. R. Kumar *et al.*, “IMAGE ENHANCEMENT FOR ULTRASOUND IMAGES USING SOBEL EDGE DETECTION,” *J. Crit. Rev.*, vol. 7, no. 13, pp. 390–394, 2020.