

PERBAIKAN CITRA BER-NOISE MENGGUNAKAN SWITCHING MEDIAN FILTER DAN BOUNDARY DISCRIMINATIVE NOISE DETECTION

Ahmad Saikhu, Nanik Suciati, Widhiantantri S.

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember,
E-mail: saikhu@its-sby.edu, nanik@its-sby.edu, saraswati@cs.its.ac.id

ABSTRAK – Citra yang mengalami proses editing ataupun kompresi biasanya akan terkontaminasi noise yang akan mengurangi kualitas citra. Beberapa teknik filtering telah diperkenalkan untuk penghilangan noise citra. Penelitian ini mengangkat suatu metode dalam penghilangan noise citra yaitu switching median filter yang digabung dengan sebuah metode deteksi noise yang disebut Boundary Discriminative Noise Detection (BDND). Dalam algoritma BDND untuk menentukan apakah suatu piksel termasuk noise atau bukan, maka piksel tersebut diklasifikasikan ke dalam 3 kelompok yaitu noise berintensitas rendah, piksel yang bukan noise, dan noise berintensitas tinggi. Suatu piksel dianggap bukan noise jika piksel termasuk kelompok kedua. Untuk membentuk 3 kelompok dibutuhkan 2 pembatas. Algoritma ini cukup bagus karena dengan noise density mencapai 90% bisa menghasilkan kesalahan deteksi 0. Empat model noise digunakan dalam uji coba untuk mengevaluasi ketangguhan algoritma BDND. Hasil uji coba pada citra grayscale dan warna dengan range noise density antara 10%-90% menunjukkan bahwa switching median filter yang digabung dengan algoritma BDND mempunyai kinerja yang sangat bagus dalam mengembalikan detail citra dalam range noise density antara 10% -70%.

Kata kunci : switching median filter, penghilangan noise, deteksi noise

1. PENDAHULUAN

Noise muncul dalam citra saat proses perubahan atau kompresi citra. Intensitas noise tinggi maupun rendah dapat menurunkan kualitas citra dan menyebabkan hilangnya beberapa detail informasi citra. Untuk menghilangkan noise, ada beberapa teknik filtering. Sebelumnya dikenal filter linear, yang mana filter ini menyebabkan citra menjadi kabur. Kemudian dikembangkan filter nonlinear yang bisa memperbaiki kekurangan filter linear (edge atau detail citra lebih terjaga). Salah satunya yang terkenal adalah filter standard median (SM) yang menggunakan informasi intensitas piksel dalam jendela filtering dan mengganti piksel yang berada pada pusat jendela dengan nilai median. Untuk keefektifan dalam pengurangan noise dan kemudahan dalam implementasi, modifikasi dari filter SM dikenalkan, seperti filter weighted median (WM) dan filter center weighted median (CWM).

Median filtering dilakukan dengan cara melakukan operasi median pada jendela filtering masing-masing piksel yang belum diketahui apakah piksel tersebut noise. Piksel yang bukan noise masih dijadikan subjek yang harus difilter. Hal ini menyebabkan penurunan kualitas citra. Solusi masalah ini adalah dengan menambahkan mekanisme deteksi noise untuk filtering yaitu hanya piksel yang diidentifikasi sebagai noise yang akan difilter. Penggabungan mekanisme deteksi noise dengan median filtering biasa disebut dengan switching median filter.

Pada makalah ini digunakan algoritma deteksi noise yang akurat, disebut BDND yang bisa digunakan pada citra dengan noise density mencapai 90%. Bersama dengan filter median NASM yang dimodifikasi, BDND menunjukkan ketangguhan dalam peak signal-to-noise ratio

(PSNR) [2]. Dalam penghilangan noise pada citra warna, algoritma BDND tetap menunjukkan kinerja yang bagus.

Tujuan dari Penelitian ini adalah untuk memperbaiki kualitas citra yang ber noise dengan noise density yang tinggi dengan menerapkan switching median filter dan algoritma DBND.

Permasalahan yang dihadapi dalam implementasi adalah bagaimana cara mendeteksi noise dengan algoritma BDND pada citra grayscale dan citra berwarna, bagaimana mengimplementasikan switching median filtering pada citra grayscale dan citra berwarna, dan bagaimana melakukan uji coba terhadap implementasi algoritma yang telah berhasil dibuat guna mengukur ketangguhannya.

2. MODEL NOISE

Untuk menguji ketangguhan filter, digunakan 4 buah model noise [2], yaitu:

• Model Noise I

Model noise I dimodelkan sebagai noise salt-and-paper, Piksel citra yang asli diberi nilai secara random dengan dua nilai, 0 dan 255 dengan probabilitas yang sama. Untuk setiap citra, piksel pada lokasi (i,j) dengan nilai intensitas $s_{i,j}$, korespondensi dengan citra ber-noise $x_{i,j}$ dan fungsi padat probabilitas $x_{i,j}$ adalah (1).

$$f(x) = \begin{cases} \frac{p}{2}, & \text{for } x = 0 \\ 1 - p, & \text{for } x = s_{i,j} \\ \frac{p}{2}, & \text{for } x = 225 \end{cases} \quad (1)$$

dimana p adalah noise density.

• Model Noise II

Sama dengan model noise I dengan probabilitas yang berbeda (p_1 dan p_2). Persamaan (2) adalah fungsi fungsi padat probabilitasnya

$$f(x) = \begin{cases} \frac{p_1}{2}, & \text{for } x = 0 \\ 1-p, & \text{for } x = Si, j \\ \frac{p_2}{2}, & \text{for } x = 255 \end{cases} \quad (2)$$

dimana $p = p_1 + p_2$ dan $p_1 \neq p_2$

• **Model Noise III**

Piksel citra yang asli di beri nilai secara acak dengan dua range nilai batas tepi dengan lebar masing-masing m . Bila $m = 10$ maka *noise* yang akan diisikan adalah $[0,9]$ atau $[246,255]$. Persamaan (3) adalah fungsi padat probabilitasnya

$$for(x) = \begin{cases} \frac{p}{2m}, & \text{for } 0 \leq x \leq m \\ 1-p, & \text{for } x = Si, j \\ \frac{p}{2m}, & \text{for } 255-m < x \leq 255 \end{cases} \quad (3)$$

dimana p adalah *noise density*.

• **Model Noise IV**

Sama dengan model *noise* III hanya probabilitas yang berbeda (p_1 dan p_2). Berikut adalah fungsi *probabilty density*-nya

$$f(x) = \begin{cases} \frac{p_1}{m}, & \text{for } 0 \leq x \leq m \\ 1-p, & \text{for } x = Si, j \\ \frac{p_2}{m}, & \text{for } 255-m < x \leq 255 \end{cases}$$

dimana $p = p_1 + p_2$ dan $p_1 \neq p_2$

3. ALGORITMA BDND

Algoritma BDND dikenakan pada masing-masing piksel pada citra ber-noise untuk mengidentifikasi apakah piksel tersebut suatu noise. Saat algoritma sudah dijalankan pada seluruh citra, pada akhir tahap deteksi noise dihasilkan dua dimensi *decision map* yang berisi angka "0" dan "1". Angka "0" menunjukkan bahwa piksel pada posisi tersebut bukan noise dan angka "1" menunjukkan noise.

Langkah pertama untuk menghasilkan *decision map* ini yaitu semua piksel akan digolongkan menjadi tiga kelompok. Dibutuhkan dua batas b_1 dan b_2 untuk membentuk tiga kelompok. Untuk masing-masing piksel $x_{i,j}$ jika $0 \leq x_{i,j} \leq b_1$, piksel akan dimasukkan ke dalam kelompok intensitas rendah; sebaliknya kelompok intensitas medium untuk $b_1 < x_{i,j} \leq b_2$ atau kelompok intensitas tinggi untuk $b_2 < x_{i,j} \leq 255$.

Jika piksel termasuk medium, maka tidak dianggap sebagai noise. Tetapi kondisi ini masih mungkin piksel tersebut masih berupa noise. sehingga akurasi deteksi noise bergantung pada seberapa akurat mengidentifikasi batas b_1 dan b_2 .

Untuk menentukan b_1 dan b_2 adalah sebagai berikut. Dibuat histogram dari 21×21 sub citra yang diekstrak dari citra Lena dengan noise density 80% dan model noise 3. Untuk penggambaran, sub citra yang dipilih adalah citra

netral yang isinya tidak terlalu "flat" (berisi frekuensi rendah) dan juga tidak berisi frekuensi tinggi. Dalam distribusi histogram terlihat dua gap atau lembah yang menandakan kemungkinan posisi dari dua batas b_1 dan b_2 . Gap tersebut memisahkan antara region ber-noise (pada masing-masing tepi distribusi) dan region yang tidak ber-noise (region yang lebih lebar berada di tengah). Ini digunakan untuk membagi semua piksel yang berada pada jendela ke dalam tiga kelompok. – noise dengan intensitas rendah, piksel yang tidak ber-noise, noise dengan intensitas tinggi.

Boundary discriminative proses terdiri dari dua iterasi. Iterasi pertama, menentukan jendela dengan ukuran 21×21 [2] digunakan untuk memeriksa apakah piksel yang sedang dipertimbangkan termasuk noise. Jika piksel tidak termasuk medium, pada iterasi kedua akan dilakukan pemeriksaan piksel dengan menggunakan jendela 3×3 . Berikut ini adalah langkah-langkah algoritma BDND :

1. Tentukan sebuah jendela misalnya 21×21 yang berpusat pada piksel saat ini.
2. Urutkan piksel-piksel yang terdapat dalam jendela (vo) dan cari mediannya med .
3. Hitung perbedaan intensitas pada masing-masing piksel yang berdekatan (vD).
4. Untuk piksel dengan intensitas antara 0 dan med pada vo , cari perbedaan intensitas maksimum pada VD pada range yang sama dan tandai piksel tersebut pada vo sebagai batas b_1 .
5. Batas b_2 ditentukan untuk piksel dengan intensitas antara medium dan 255, sehingga terbentuk 3 kelas.
6. Jika piksel berada pada kelas medium maka termasuk piksel yang tidak korup dan proses klasifikasi dihentikan, jika tidak maka dilanjutkan pada langkah 7.
7. Tentukan jendela dengan ukuran 3×3 yang berpusat pada piksel saat ini kemudian ulangi langkah 2 sampai langkah 5.

Jika piksel tetap berada pada kelas medium maka piksel tersebut termasuk piksel yang tidak korup.

3.1 Deteksi Noise

Algoritma ini bertujuan agar proses filtering lebih cerdas. Proses filtering hanya dilakukan terhadap piksel yang dinyatakan sebagai noise.

3.2 Deteksi Noise pada Citra Grayscale

Proses deteksi noise pada citra grayscale adalah dengan menggunakan algoritma BDND pada masing-masing piksel citra ber-noise. Proses ini dilakukan untuk mengidentifikasi apakah piksel tersebut berisi noise. Setiap piksel diperiksa apakah berisi noise. Jika piksel adalah noise maka diberi kode 1, jika tidak diberi kode 0. Kode ini disimpan dalam bentuk matriks berukuran sama dengan ukuran citra ber-noise disebut sebagai *decision map*. Saat algoritma sudah dijalankan pada seluruh citra, pada akhir tahap dihasilkan

dua dimensi decision map. Untuk menghasilkan kode angka 1 dan 0, pertama piksel dikategorikan ke dalam tiga golongan yaitu kelompok intensitas rendah, kelompok medium dan kelompok dengan intensitas tinggi. Untuk membentuk tiga golongan tersebut diperlukan dua batas pembeda misalnya b_1 dan b_2 .

3.3 Deteksi Noise pada Citra Warna

Dalam pemrosesan citra digital yang paling sering digunakan adalah space warna RGB, masing-masing piksel pada lokasi (i,j) direpresentasikan menjadi vektor warna menjadi $s_{i,j}=(sR_{i,j}, sG_{i,j}, sB_{i,j})$ dimana $sR_{i,j}$, $sG_{i,j}$, $sB_{i,j}$ adalah komponen red (R), green (G) dan blue (B). Pada proses penambahan noise pada citra berwarna, proses memasukkan noise dilakukan pada setiap komponen R, G, dan B. Jika citra berwarna diberi noise dengan noise density p , maka masing-masing komponen R, G dan B diberi noise dengan noise density p . Sehingga untuk proses deteksi noise dilakukan pada masing-masing komponen R, G dan B karena posisi noise yang berbeda pada tiap komponen. Proses setiap komponen R, G dan B sama dengan proses deteksi noise pada citra grayscale. Setiap komponen akan menghasilkan satu decision map sehingga citra berwarna akan menghasilkan tiga buah decision map. Pada akhirnya akan digabung menjadi satu decision map berdimensi tiga.

4. PENGHILANGAN NOISE

Proses penghilangan noise merupakan proses penerapan metode switching median filter. Proses penghilangan noise dengan switching median filter terdiri atas dua proses, yaitu perkiraan noise density dan perbaikan algoritma switching median filter.

4.1 Perkiraan Noise Density

Dalam menentukan ukuran jendela filtering $WF \times WF$, batas maksimum ukuran jendela $WD1 \times WD1$ harus ditentukan terlebih dahulu. Untuk itu berdasarkan pustaka [2] ditentukan ukuran jendela $WD1$ berdasarkan beda perkiraan noise density sesuai tabel 1.

Tabel 1. Ukuran $WD1$ berdasarkan Perkiraan Noise Density

Noise Density	$WD1 \times WD1$
0%	3
$< p \leq 20\%$	$\times 3$
20%	5
$< p \leq 40\%$	$\times 5$
40%	7
$>$	$\times 7$

Untuk perkiraan noise density pada pada BDND menggunakan cara yang sederhana yaitu dengan menghitung angka 1 pada decision map yang telah dihasilkan pada proses deteksi noise dan membaginya dengan jumlah piksel keseluruhan.

4.2 Perbaikan Algoritma Switching Median Filter

Switching median filter adalah filter median yang dikenakan kepada piksel yang sudah dideteksi sebagai noise. Switching median filter sudah mengalami beberapa perbaikan dari penelitian sebelumnya Noise Adaptive Soft-Switching Median Filter [1] untuk selanjutnya disebut sebagai NASM.

Dibandingkan dengan tahap filtering NASM, ada tiga perubahan yang dijelaskan sebagai berikut.

Pertama, ukuran jendela maksimum $WD1 \times WD1$ dibatasi 7×7 untuk menghindari hasil filtering yang terlalu kabur pada noise density tinggi ($p > 50\%$). Setelah itu ukuran jendela filter $WF \times WF$ didapat dengan cara yang sama pada NASM dengan beberapa modifikasi sebagai berikut.

Pada NASM, dimulai dengan $WF = 3$, jendela filtering secara iterative diperbesar sebesar satu piksel pada keempat sisi jendela, asalkan jumlah piksel yang dideteksi bukan noise (dengan notasi N_c) kurang dari setengah jumlah piksel (dengan notasi $S_{in} = (\frac{1}{2}) [WF \times WF]$) dalam jendela filtering, dimana $WF \leq WD1$. Jendela filtering (WF) juga di perbesar ketika jumlah piksel yang bukan noise sama dengan nol. Oleh karena itu, perubahan kedua untuk perbaikan yaitu pada saat ($N_c < S_{in}$ dan $WF \leq WD1$) atau $N_c = 0$, jendela $WF \times WF$ akan di perbesar sebesar satu piksel pada keempat sisi jendela.

Dengan menggunakan Standard Median (SM) filter dengan ukuran jendela $WF \times WF$ pada sebuah piksel ber-noise, maka output piksel $Y_{i,j}$ adalah

$$Y_{i,j} = \text{median} \{ X_{i-s, j-t}(s,t) \in W \},$$

dimana $W = \{ (s,t) \mid - (WF-1)/2 \leq s,t \leq (WF - 1)/2 \}$

Pada NASM, piksel saat ini masih disertakan dalam proses pencarian nilai median. Perubahan ketiga adalah tidak mengikutkan piksel saat ini yang sudah termasuk dalam piksel yang dinyatakan ber-noise dalam proses filtering. Jadi hanya piksel dalam jendela yang dinyatakan tidak ber-noise yang diikuti dalam proses filtering. Ini akan menghasilkan hasil filtering yang lebih bagus.

5. DESAIN DAN IMPLEMENTASI

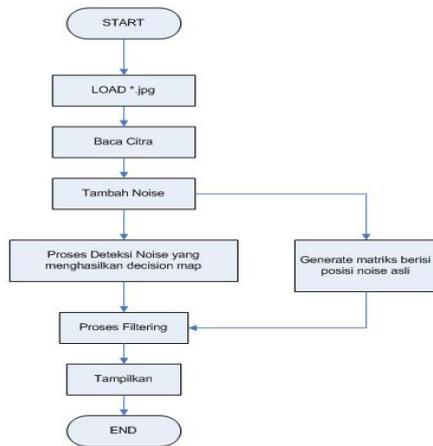
Pada tahap desain digunakan Diagram Alir Data untuk mengetahui proses-proses apa saja yang diperlukan. Perangkat lunak yang dibuat berupa aplikasi desktop yang dibangun dengan MATLAB 7.

Data masukan adalah file citra dan jenis noise yang ditambahkan. Sedangkan keluaran terdiri dari 2 citra, citra hasil filtering dan citra hasil ideal filtering.

Pengguna memilih citra kemudian sistem membacanya dan menyimpannya dalam bentuk matriks. Jika masukan adalah citra warna, akan ditanyakan oleh sistem apakah akan diubah menjadi citra grayscale.

Ada 3 tahap proses utama dalam implementasi metode penghilangan noise, yaitu penambahan noise, deteksi noise dan penghilangan noise. Proses penambahan noise adalah proses pengubahan citra input menjadi citra ber-noise. Noise yang dimasukkan adalah inputan dari pengguna. Setelah semua parameter dimasukkan, sistem akan menampilkan hasil penambahan noise tadi.

Selanjutnya sistem melakukan filtering, yaitu proses deteksi noise dan penghilangan noise. Dalam proses deteksi noise setiap piksel akan diperiksa apakah piksel tersebut noise.



Gambar 1 Deskripsi Sistem

Proses ini menggunakan algoritma BDND. Piksel diberi nilai 1 jika piksel tersebut adalah noise dan diberi nilai 0 jika piksel tersebut bukan noise. Proses ini akan menghasilkan sebuah decision map berukuran sama dengan citra ber-noise (atau citra asli) yang berisi nilai 0 atau 1. Setelah itu dilakukan proses filtering. Proses ini hanya dikenakan kepada piksel yang sudah dideteksi sebagai noise (dalam decision map mempunyai nilai 1). Proses ini menggunakan median filtering dimana yang dimasukkan ke dalam array untuk mencari median adalah piksel yang sudah dideteksi bukan noise (dalam decision map bernilai 0).

6. UJI COBA

Uji coba yang dilakukan yaitu uji coba berdasarkan model noise yang berbeda dengan variasi noise density yang berbeda-beda.

6.1 Uji Coba Model Noise I

Tabel 2 adalah hasil uji coba citra lena.jpg berukuran 512x512 dengan model noise I dan variasi noise density antara 10 sampai 90 %.

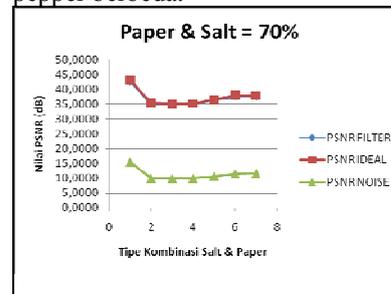
Tabel 2 Hasil Uji Noise I Citra Lena.jpg

Noise Density	Miss Detection (MD)		False Alarm (FA)	
	proposed	ASM	proposed	ASM
10	0	1	19	743
20	0	42	85	54
30	0	69	23	35
40	0	61	01	71
50	0	032	31	77
60	0	643	05	30
70	0	1231	08	2
80	0	5016	13	03
90	0			

Dari hasil ujicoba diatas terlihat bahwa BDND lebih tangguh dari pada NASM. Data NASM diperoleh dari makalah dengan judul Switching Median Filter with Boudary Discriminative Noise Detection for Extremely Corrupted Images[2]. Miss detection mencapai 0, ini berarti tidak ada sama sekali noise yang tidak terdeteksi. Tetapi dalam hal false alarm terlihat angka yang secara bergantian lebih besar atau lebih kecil dari NASM.

6.2 Uji Coba Model Noise II

Berikut ini adalah hasil uji coba citra lena.jpg berukuran 512x512 dengan model noise II dan noise density sebesar 70% dengan variasi salt and pepper berbeda.



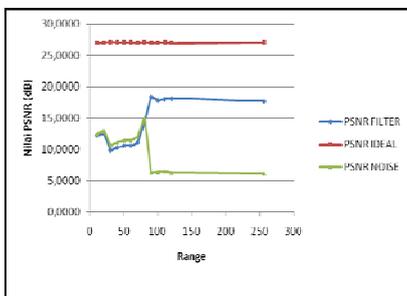
Gambar 1 Grafik Perbandingan PSNR (Noise II, Noise Density 70%)

Pada gambar 2, terlihat bahwa algoritma Boundary Discriminative Noise Detection yang digabung dengan switching median filter mempunyai kinerja yang bagus. Hal ini terlihat pada grafik yaitu posisi PSNR filter berada jauh diatas PSNR noise. PSNR noise yang berada disekitar angka 10 dibandingkan dengan PSNR filter yang berada disekitar angka 35 ke atas. Perlu diketahui bahwa nilai PSNR untuk rekonstruksi citra yang dianggap bagus berkisar antara 35 sampai 40.

Pada gambar 2 juga terlihat PSNR filter berhimpit dengan PSNR ideal filter. Ini menunjukkan deteksi noise algoritma BDND mendekati deteksi noise ideal artinya miss detection bernilai 0 dan false alarm mendekati 0.

6.3 Uji Coba Model Noise III

Berikut ini adalah hasil uji coba citra lena.jpg berukuran 512x512 dengan model noise III dan noise density sebesar 80% dengan variasi range intensitas noise yang berbeda.

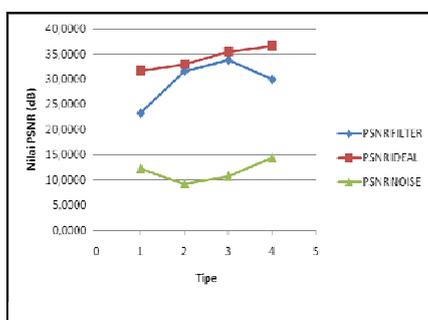


Gambar 3 Grafik Perbandingan PSNR (Noise III, Noise Density 80%)

Pada grafik di atas ini terlihat bahwa kinerja algoritma Boundary Discriminative Noise Detection mengalami penurunan dibandingkan dengan uji coba menggunakan model noise I dan model noise II. Hal ini terjadi dikarenakan intensitas noise yang dimodelkan pada model noise III sangat bervariasi berdasarkan range beda intensitas (m). Sehingga noise yang dideteksi bukan lagi bernilai 255 atau 0 tetapi sudah mulai abu-abu sesuai dengan range-nya. Jika range-nya (m) bernilai 50 maka intensitas noise yang rendah yang dimodelkan berkisar antara 0-49 dan intensitas noise yang tinggi yang dimodelkan berkisar antara 206-255. Karena variasi noise inilah yang menyebabkan kinerja algoritma Boundary Discriminative Noise Detection menurun.

6.4 Uji Coba Model Noise IV

Berikut ini adalah hasil uji coba citra lena.jpg berukuran 512x512 dengan model noise IV dan noise density sebesar 80% dengan variasi noise salt and paper yang berbeda dan range intensitas noise sebesar 30.



Gambar 4. Grafik Perbandingan PSNR (Noise IV, Noise Density 80%, m = 30)

Pada gambar 4 terlihat bahwa kinerja algoritma Boundary Discriminative Noise

Detection mengalami penurunan dibandingkan dengan uji coba model noise I dan model noise II. Tetapi hasilnya sedikit mendekati ideal. Nilai range intensitas noise yang masih kecil yaitu 30 berpengaruh dalam deteksi noise. Karena noise yang dimodelkan bervariasi dari 0-29 dan 226-255.

7. KESIMPULAN DAN SARAN

Dari aplikasi yang telah dibuat dan berdasarkan uji coba yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

- Algoritma Boundary Discriminative Noise Detection sangat tangguh dalam mendeteksi noise. Karena dari uji coba yang telah dilakukan miss detection mencapai nilai 0 untuk noise density mencapai 90% .
- Algoritma Boundary Discriminative Noise Detection yang digabung dengan switching median filter mampu memperbaiki kualitas citra yang sudah terkena noise dengan noise density yang mencapai 70%.
- Kinerja algoritma Boundary Discriminative Noise Detection bergantung pada variasi intensitas noise dan intensitas citra itu sendiri. Semakin besar variasi intensitas noise semakin menurun kinerja algoritma Boundary Discriminative Noise Detection dan semakin rendah intensitas citra atau semakin tinggi intensitas citra menyebabkan kegagalan deteksi yang semakin besar karena miripnya citra asli dengan citra noise.

Saran untuk kemungkinan pengembangan lebih lanjut dari pembuatan aplikasi penghilangan noise dengan metode boundary discriminative noise detection:

- Perubahan dalam algoritma dengan sedikit perulangan sehingga mempercepat proses penghilangan noise.
- Membuat perubahan pada algoritma sehingga mengurangi false alarm yang masih besar

PUSTAKA

- H,-L. Eng and K.-K Ma. (2001). Noise adaptive soft-switching median filter. IEEE Trans. Image Process., vol. 10, no. 2, pp. 242-251.
- P,-Eng Ng and K.-K Ma. (2006). A switching median filter with boundary discriminative noise detection for extremely corrupted images. IEEE Trans. Image Process., vol. 15, no. 6.
- Gonzales, R; Woods, R. (2002). Digital Image Processing Second Edition. Prentice Hall..