Jurnal Riset Komputer (JURIKOM), Vol. 2 No. 6, Desember 2015
ISSN 2407-389X (Media Cetak)
Hal : 56-61

METODE PERANCANGAN PENGARANGKAT LUNAK MEREDUKSI NOISE CITRA DIGITAL MENGGUNAKAN CONTRAHARMONIC MEAN FILTTER

Masnun Dasopang

Mahasiswa Teknik Informatika STMIK Budi Darma
Jl. Sisingamangaraja No. 338 Simpang Limun Medan

ABSTRAK

Citra hasil rekaman kamera digital sering sekali terdapat beberapa gangguan yang mungkin terjadi, seperti lensa tidak fokus, muncul bintik-bintik yang disebabkan oleh proses capture yang tidak sempurna, pencahayaan yang tidak merata yang mengakibatkan intensitas tidak seragam, kontras citra terlalu rendah sehingga objek sulit dipisahkan dari latarbelakangnya, atau gangguan yang disebabkan oleh kotoran- kotoran yang menempel pada citra dan lain sebagainya. Meskipun sebuah citra kaya akan informasi, namun sering kali citra yang dimiliki mengalami penurunan mutu, misalnya mengandung cacat atau noise. Tentu saja citra semacam ini menjadi lebih sulit untuk diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Untuk mengatasi noise tersebut perlu dilakukan usaha untuk memperbaiki kualitas citra itu.

Kata Kunci: Reduksi Noise pada Citra, GeometricMean Filter

I. PENDAHULUAN

Citra hasil rekaman kamera digital sering sekali terdapat beberapa gangguan yang mungkin terjadi, seperti lensa tidak fokus, muncul bintik-bintik yang disebabkan oleh proses *capture* yang tidak sempurna, pencahayaan yang tidak merata yang mengakibatkan intensitas tidak seragam, kontras citra terlalu rendah sehingga objek sulit dipisahkan dari latar belakangnya, atau gangguan yang disebabkan oleh kotoran- kotoran yang menempel pada citra dan lain sebagainya. Meskipun sebuah citra kaya akan informasi, namun seringkali citra yang dimiliki mengalami penurunan mutu, misalnya mengandung cacatan *noise*. Tentu saja citra semacam ini menjadi lebih suli tuntuk diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Untuk mengatasi *noise* tersebut perlu dilakukan usaha untuk memperbaiki kualitas citra itu. Salah satunya adalah dengan *filtering* citra baik secara linear maupun secara on-linear. merupakan salah satu *filtering linear* yang berfungsi untuk memperhalus dan menghilangkan *noise* pada suatu citra yang bekerja dengan menggantikan intensitas nilai *pixel* dengan rata-rata dari nilai *pixel* tersebut dengan nilai *pixel-pixel* tetangganya.

Ada beberapa jenis *noise* yang terdapat pada citra digital antara lain adalah *noise* Gaussian, Salt-and-Pepper, Gamma, Eksponensial, Rayleigh, dan Uniform. Efek *noise-noise* di atas terhadap sebuah citra berbeda-beda karena ada yang mempengaruhi tampilan citra dana dan juga yang tidak begitu mempengaruhi. Salah satu teknik yang digunakan adalah reduksi *noise* yang melakukan restorasi citra dengan cara peningkatan kualitas. *Noise* yang akan dibahas adalah *noise* yang terjadi karena karakteristik dari derajat keabu-abuan (*gray-level*) atau karena adanya variabel acak yang terjadi karena karakteristik Fungsi Probabilitas Kepadatan (*Probability Density Functionatau PDF*). Kumpulan *PDF* tersebut sangat berguna untuk memperagakan situasi kerusakan yang diakibatkan oleh *noise*. Contohnya *noise* Gaussian muncul oleh factor elektronik, kurangnya pencahayaan atau akibat suhu yang terlalu tinggi. *Noise* Rayleigh sangat membantu untuk memodelkan fenomena derau pada saat pencitraan. *Noise* Eksponensial dan Gamma ditemukan pada pencitraan dengan menggunakan laser. Teknik reduksi *noise* yang dibahas adalah teknik filter yang melakukan teknik pemisahan *noise* dari objek-objek pada citra. Algoritma yang digunakan pada Contraharmonic Maen Filtering. Yang melakukan reduksi pada *noise* dengan cara mengganti nilai piksel dengan nilai tengah (rata-rata) intensitas piksel citra yang mengandung *noise*.

II. TEORITIS

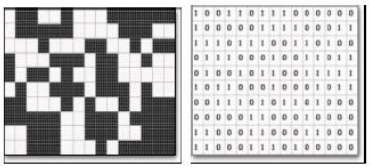
A. Citra

Citra (*image*) atau istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Meskipun sebuah citra kaya akan informasi, namun sering kali citra yang dimiliki mengalami penurunan mutu, misalnya mengandung cacat atau *noise*. Tentu saja citra semacam ini menjadi lebih sulit untuk diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang (Rachmad, 2008).

B. Matriks *bitmap*

Citra *bitmap* adalah susunan bit-bit warna untuk tiap *pixel* yang membentuk pola tertentu. Pola-pola warna ini menyajikan informasi yang dapat dipahami sesuai dengan persepsi indera penglihatan manusia. Format file ini merupakan format grafis yang fleksibel untuk *platform Windows* sehingga dapat dibaca oleh program grafis manapun. Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit sampai 24 bit. Citra bitmap didefinisikan sebagai fungsi $f(x,y)$ dengan x dan y adalah koordinat bidang. Besaran f untuk tiap koordinat (x,y) disebut intensitas atau derajat keabuan citra pada titik tersebut (Jannah, 2008). Pada Gambar

1 ditunjukkan gambar *bitmap* beserta nilai matriksnya



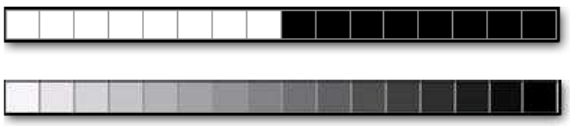
Gambar 1. Bitmap dengan nilai matriksnya
Sumber : Sutoyo, 2009

Dari definisi di atas yang diperjelas oleh Gambar 1, *bitmap* dimodelkan dalam bentuk matriks. Nilai *pixel* atau entri-entri dari matriks ini mewakili warna yang ditampilkan di mana ordo matriks merupakan dimensi panjang dan lebar dari *bitmap*. Nilai-nilai warna ditentukan berdasarkan intensitas cahaya yang masuk. Dalam komputer, derajat intensitas cahaya diwakili oleh bilangan cacah. Nilai 0 menerangkan tidak adanya cahaya sedangkan nilai yang lain menerangkan adanya cahaya dengan intensitas tertentu. Nilai-nilai ini bisa didapatkan melalui fungsi-fungsi yang disediakan oleh bahasa pemrograman berdasarkan input berupa lokasi entri-entri matriks yang hendak dicari.

C. *Pixel*

Pixel (*Picture Elements*) adalah nilai tiap-tiap entri matriks pada *bitmap*. Rentang nilai-nilai *pixel* ini dipengaruhi oleh banyaknya warna yang dapat ditampilkan. Jika suatu *bitmap* dapat menampilkan 256 warna maka nilai-nilai *pixel*nya dibatasi dari 0 hingga 255. Suatu *bitmap* dianggap mempunyai ketepatan yang tinggi jika dapat menampilkan lebih banyak warna (Putra, 2010).

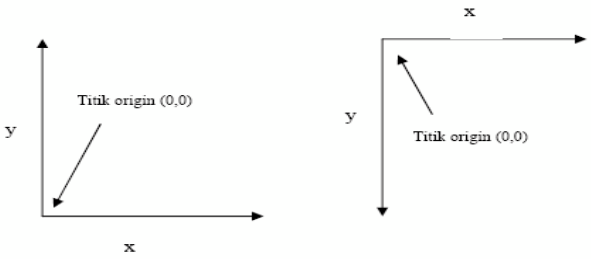
Prinsip ini dapat dilihat dari contoh pada Gambar 2 yang memberikan contoh dua buah *bitmap* dapat memiliki perbedaan dalam menangani transisi warna putih ke warna hitam.



Gambar 2. Perbedaan ketepatan warna *bitmap*
Sumber : Sutoyo, 2009

Perbedaan ketepatan warna *bitmap* pada Gambar 2.3 menjelaskan bahwa *bitmap* sebelah atas memberikan nilai untuk warna lebih sedikit daripada *bitmap* di bawahnya. Untuk *bitmap* dengan pola yang lebih kompleks dan dimensi yang lebih besar, perbedaan keakuratan dalam memberikan nilai warna akan terlihat lebih jelas. Sebuah citra adalah kumpulan *pixel-pixel* yang disusun dalm larik dua dimensi. Indeks baris dan kolom (*x,y*) dari sebuah *pixel* dinyatakan dalam bilangan bulat. *Pixel* (0,0) terletak pada sudut kiri atas pada citra, indeks *x* begerak ke kanan dan indeks *y* bergerak ke bawah.

Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer. Letak titik origin pada koordinat grafik citra dan koordinat pada grafik matematika terdapat perbedaan. Hal yang berlawanan untuk arah vertikal berlaku pada kenyataan dan juga pada sistem grafik dalam matematika yang sudah lebih dulu dikenal. Gambar berikut memperlihatkan perbedaan kedua sistem ini.



Gambar 3. Perbedaan letak titik origin pada koordinat grafik dan pada citra
Sumber : Putra, 2010

D. Citra digital

Citra digital adalah citra yang dapat diolah oleh komputer. Misalnya sebuah citra grayscale ukuran 150x150 pexel (elemen terkecil dari sebuah citra diambil sebagian (kotak kecil) bekuran 9x9 pixel. Jadi sebagian kecil itu adalah contoh bahwa citra digital disimpan dalam memori komputer dalam bentuk angka-angka yang menunjukkan besar intensitas pada masing-masing pixel tersebut. citra digital yang diambil dalam kotak itu disimpan dalam bentuk angka-angka dalam memori komputer

E. Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak dikertas foto, lukisan, pemandangan alam, hasil CT scan dan lain sebagainya. Citra analog tidak dapat dipresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog diantaranya adalah video kamera analog, kamera foto analog dan CT scan.

Suatu citra dapat didefinisikan sebagai fungsi f (*x,y*) berukuran M baris dan N kolom, dengan *x* dan *y* adalah koordinat (*x,y*) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai *x,y* dan nilai amplitudo *f* secara keseluruhan berhingga (*finite*) dan bernilai diskrit, maka dapat dikatakan bahwa citra tersebut adalah citra digital. Citra digital dapat ditulis dalam bentuk matrik sebagai berikut (Rachmad, 2008). Nilai pada suatu irisan antara baris dan kolom (pada posisi *x,y*) disebut dengan *picture element*, *image element*, *pels*, atau *pixel*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital.

F. Karnel

Dalam domain komputasi karnel adalah inti

dari sistem operasi, karnel ini dikelilingi oleh *software*, menyediakan hak akses pengguna dan memfasilitasi interaksi seperti, *shell*, *window* manager dan program aplikasi, Untuk menjalankan program real time untuk menjalankan sistem operasi yang mana telah mengalami perubahan yang khususnya pada karnel standarnya sehingga memiliki sifat yang real time. (Yuwono, 2010).

G. Konvolusi

Konvolusi (convolution) adalah sebuah proses dimana citra dimanipulasi dengan menggunakan eksternal mask / subwindows untuk menghasilkan citra yang baru. Sedangkan Filtering tanpa menggunakan ekternal mask tetapi hanya menggunakan pixel tetangga untuk mendapatkan pixel yang baru.

Konvolusi sangat banyak dipergunakan dalam pengolahan citra untuk memperhalus (smoothing), msenajamkan (crispening), mendeteksi tepi (edge detection), serta efek lainnya. (Santoso, Isman. 2013).

1. Embossing
   Embossing yaitu membuat citra seolah diukir pada permukaaan selembar nikel. Koefisien jendela konvolusi memiliki bobot tengah bernilai 0 & jumlah seluruh bobot = 0.
2. Blurring
   Blurring (Pengaburan) yaitu filter spasial low-pass yang melenyapkan detil halus dari suatu citra. Pengaburan dicapai melalui konvolusi dari seluruh koefisien mask bernilai sama. Blurring ini perataan nilai pixel-pixel tetangga, makin besar ukuran mask maka makin besar efek pengaburan
3. Sharpening
   Sharpening (Penajaman) yaitu memperjelas detil suatu citra(menambah kontras) dengan penjumlahan atas citra tepi dengan citra aslinya maka bagian tepi objek akan terlihat berbeda dengan latarnya, sehingga citra terkesan lebih tajam.
4. Edge Detection
   Deteksi tepi yaitu proses menentukan lokasi titik-titik yang merupakan tepi objek.
   Demikianlah pembahasan tentang Proses Konvolusi untuk mengolah citra.
   tag : konvolusi adalah , konvolusi citra digital , proses konvolusi , konvolusi , konvolusi pengolahan citra , pengertian blurring , pengertian sharpening , konvolusi pada pengolahan citra , pengertia konvulsi , proses konvolusi citra

H. *Contra-Harmonic Mean Filter*

Noise Salt-and-Pepper dan *noise Speckle* adalah *noise* yang sering ada dijumpai pada citra digital. Citra yang memiliki *noise* biasanya terjadi karena kesalahan tehnik pengambilan citra . Untuk mereduksi *noise* dibutuhkan suatu metode *filter* yang tepat agar citra yang dihasilkan sesuai dengan aslinya. Metode *Harmonic Mean Filter* dan *Contra-Harmonic Mean Filter* adalah salah satu metode *filter* untuk mereduksi sebuah *noise*. Dari hasil perhitungan rata-rata *Mean Square Error (MSE)* dan *Peak Signal to-Noise Ratio* (PSNR) maka dapat disimpulkan metode *Harmonic Mean Filter* lebih baik mereduksi *salt & pepper noise*, sedangkan metode *Contra-Harmonic Mean Filter* lebih baik mereduksi *speckle noise* dengan nilai *order filter*(Q) adalah nol(Q=0). (Sulistyo, 2009).

I. Model-Model Noise

Noise dalam image dapat terjadi karena faktor-faktor seperti lingkungan dan *channel* pengiriman data. Noise pada bahasan disini adalah noise yang terjadi karena karakteristik dari derajat keabu-abuan (*gray-level*) atau dikarenakan adanya variabel acak yang terjadi karena karakteristik Fungsi Probabilitas Kepadatan (*Probability Density Function* (PDF)). (Sulistyo, 2009).

Beberapa noise yang terjadi karena PDF antara lain,

1. *Gaussian Noise*
2. *Rayleigh Noise*
3. *Erlang (Gamma) Noise*
4. *Exponential Noise*
5. *Uniform Noise*
6. *Impulse (Salt-and-Pepper) Noise*

Jika terdapat kasus sebuah gambar diberi 6 tipe Noise diatas, maka yang dapat dibedakan secara kasat mata hanyalah *Impulse Noise*. Tetapi secara *histogram*, sebuah image yang telah diberi noise dapat dibedakan tipenya.

J. Mean Filters

Arithmetic Mean Filter Ini adalah metode paling mudah dari mean filters. Misalkan Sxy mewakili himpunan koordinat dalam sebuah window kotak subimage berukuran m x n, berpusat di titik (x, y). Proses Arithmetic Mean Filtering menghitung rata – rata nilai dari gambar yang rusak g(x, y) pada area yang didefinisika oleh Sxy. Nilai dari gambar yang diperbaiki pada tiap titik (x, y) adalah arithmetic mean mudah yang dihitung dengan menggunakan pixel dalam daerah yang didefinisikan oleh Sxy. Dengan kata lain :

III. ANALISA dan PEMBAHASAN

Reduksi *noise* menggunakan algoritma *Geometric Mean Filter* adalah proses *filtering* untuk mereduksi *noise* dari citra yang ber-*noise*. Tahap-tahap proses yaitu:

1. Membaca nilai piksel.
2. Menghitung nilai RGB piksel.
3. Menghitung nilai *grayscale pixel*.
4. Penambahan *noise* pada citra grayscale.
5. Reduksi *noise* dengan Geometric mean filter.
6. Perhitungan nilai MSE.

Proses awal adalah pembacaan nilai *pixel* file citra bertujuan untuk mendapatkan data *bitmap* untuk perhitungan nilai RGB dan pembentukan citra *grayscale*. Pada proses penambahan *noise*, dihasilkan

satu data citra ber-*noise* yang digunakan sebagai data *input* untuk proses pemilihan matriks *input*. Pada proses ini matriks citra *noise* yang awalnya berukuran 2 dimensi (memiliki baris dan kolom) akan dipecah dan diambil per kolom. Sehingga proses ini akan menghasilkan data berupa matriks yang berukuran 1 kolom dan *n* baris. Data ini akan digunakan sebagai *input* untuk proses selanjutnya yaitu konvolusi dengan koefisien *filter*. Hal ini akan terus-menerus dilakukan hingga proses pembentukan sinyal *output*. Data pada setiap proses akan berukuran 1 dimensi, sehingga setiap kolom disimpan pada suatu matriks *temporary* yang berguna untuk menggabungkan seluruh matriks kolom.

Membaca Nilai *Pixel*

Pada citra digital 24-bit tidak terdapat palet *RGB*, karena nilai *RGB* langsung diuraikan dalam data *bitmap*. Setiap elemen data *bitmap* panjangnya 3 *byte*, masing- masing *byte* menyatakan komponen *R*, *G*, dan *B*. Data *bitmap* pada citra yang diambil menggunakan kamera .

Pada citra *bitmap* 24-bit, tiap *pixel*nya mengandung 24-bit kandungan warna atau 8-bit untuk masing-masing warna dasar (R, G, dan B), dengan kisaran nilai kandungan antara 0 (00000000) sampai 255 (11111111) untuk tiap warna. Pada contoh format citra 24-bit di atas data pertama adalah *header* yang berisi informasi nama file, jenis format dan dimensi citra. Di bawah data bitmap terdapat *pixel* pertama mempunyai *R* = 10010000 (biner), *G* = 01011010 (biner), *B* = 011110001 (biner).

Bagian terkecil sebuah citra adalah *pixel* yang memiliki nilai intensitas hasil kuantisasi peralatan digital. Citra warna memiliki tiga komponen warna yang setiap komponen warna menggunakan penyimpanan 8 bit atau 1 *byte*. Berarti jumlah bit dalam 1 *pixel* adalah sebanyak 3 x 8 bit = 24 bit. Jadi untuk menghitung jumlah *pixel* sebuah citra dilakukan dengan membagi nilai intensitas citra dengan 24 bit dan resolusi 5 x 5 pixel. Representasi nilai *pixel* pada citra warna dapat dilihat pada Gambar 3

Gambar 4. Representasi *Pixel* pada Citra Terindikasi Noise

Nilai intensitas citra disimpan dalam data *bitmap* dalam bentuk nilai biner dengan nilai antara 0 dan 1. Jadi jumlah *pixel*nya adalah jumlah kelipatan 24 bit. Misalnya sebuah citra dengan nilai intensitas di atas adalah:

1111000011110000111111100011000011110000111
1111111100001111000
0111110001111000011110000111110011111000011
1100001111010101100
0011110000111111111111000011110000111111111
1110000111000011010
1111111000011110000110000011111000011111000
0111101100011110000 1
1110000111111111111000011110000111110000111
0000111100001111010 1
0111100001111000011110001101111110000111100
0011111111111100001 1
1100001110101111111100001111000011111111 dan
seterusnya

Nilai 111100001111000011111111 = 24 bit mewakili 1 *pixel* dari citra. Jika jumlah bit citra di atas adalah 1.800.000 bit, maka jumlah *pixel*-nya adalah 1.800.000/24 bit = 75.000 *pixel*. Pada reduksi *noise* ini dilakukan per blok citra dengan ukuran per bloknya adalah 3 x 3 *pixel* atau 9 *pixel*, maka jumlah blok adalah 1.800.000 : 9 = 200.000 blok.

Menghitung Nilai *Grayscale*

Selanjutnya nilai RGB per *pixel* pada Gambar 3. di atas di konversi ke nilai *grayscale* dengan cara mencari nilai rata-rata per *pixel*.

$$f_0\,(x,y) = \left(\frac{f^R(x,y) + f^G(x,y) + f^B(x,y)}{3}\right) \qquad .....3.4)$$

Dimana fR adalah nilai komponen red
fG adalah nilai komponen green
fB adalah nilai komponen blue

Nilai *grayscale* dihitung dengan menggunakan persamaan (3.4) sebagai berikut:

Pixel (0,0) = (255,240,240) = (255+240+240)/3 = 198
Pixel (0,1) = (225,129) = (240+225+129)/3 = 245
Pixel (0,2) = (240,255,240,192) = (255+240+192)/3 = 229
Pixel (1,0) = (240,225,185) = (240+225+185)/3 = 217
Pixel (1,1) = (243,252,249) = (243+252+249)/3 = 248
Pixel (1,2) = (114,225,217) = (114+225+217)/3 = 185
Pixel (2,0) = (231,240,80) = (231+240+80)/3 = 184
Pixel (2,1) = (252,48,187) = (252+48+187)/3 = 162
Pixel (2,2) = (156,240,187) = (156+240+187)/3 = 194

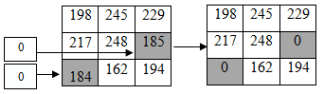
Selanjutnya nilai *grayscale* semua *pixel* pada citra *noise* blok 1 dimasukkan ke dalam matriks nilai *grayscale* seperti pada Gambar 4.

198	245	229
217	248	185
184	162	194

Gambar 5. Matriks Nilai *Grayscale* Citra Noise Blok 1

Penambahan *Noise*

Pada citra Gambar 5 dilakukan penambahan *noise salt and papper* pada citra yang ber-*noise*. *Noise* Salt and Papper merupakan *pixel* memiliki warna gelap dan terang yang nilai 0 dan 255 dan penambahan ke dalam citra dilakukan secara acak. Proses penambahan noise pada citra pada Gambar 6 dengan mengganti nilai *pixel* secara acak dengan nilai 0 dan 255 seperti pada Gambar 6.



Gambar 6 Matriks Citra Hasil Penambahan Noise

Konvolusi Matriks Citra dengan *Geometric Mean Filter*

Filter yang digunakan pada reduksi *noise* dengan *Geometric Mean Filter* berukuran 3 x 3 yang semuanya bernilai 1 dengan menggunakan rumus:

$$f(x,y) = \left[\prod_{(s,t)\in Sxy} g(s,t)\right]^{1/mn}$$

Rumus diatas berarti semua *pixel* yang mewakili nilai intensitas dari citra yang terkena jendela filter dikalikan lalu dipangkatkan dengan 1/mn. Pada Gambar 3.9 dilakukan filter dengan menggunakan rumus di atas:

198	245	229
217	248	0
0	162	194

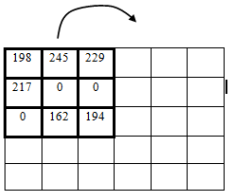
Gambar 7. Matriks Citra Noise Blok **1**

Hasil filter yang terbentuk = (198*245*229*217*248*0*0*162*194)$^{1/9}$ = 0

198	245	229
217	0	0
0	162	194

Gambar 8. Matriks Citra Noise Blok 1 Hasil Filter *Geometric Mean Filter*

Pada masing-masing filter dilakukan proses sampai semua blok citra *noise* habis di- filter dengan cara menggeser jendela filter sebanyak satu pilsel ke kanan per proses seperti pada Gambar 9



Gambar 9. Matriks Citra Noise Blok 1 Sebelum Digeser.

Dengan menggeser satu *pixel* ke kanan maka hasilnya seperti matriks Gambar 10

198	245	229	Xx		
217	0	0	Xx		
0	162	194	Xx		

Gambar 10. Matriks Citra Noise Blok 1 Sesudah Digeser

Perhitungan Nilai *Mean Squared Error* (MSE)

MSE citra hasil reduksi *noise* adalah rata-rata kuadrat nilai kesalahan antara citra asli dengan citra hasil reduksi *noise* dengan rumus sebagai berikut:

Dimana: $$MSE = \frac{1}{XY} \sum_x \sum_y [I(x,y) - I'(x,y)]^2$$

X= lebar citra dalam *pixel*
Y= tinggi citra dalam *pixel*
I= nilai *pixel* citra sebelum reduksi *noise* I' = nilai *pixel* citra sesudah reduksi *noise* Perhitungan nilai MSE pada matriks citra adalah:

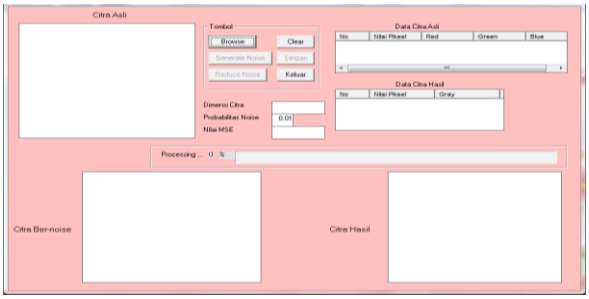
198	245	229	Xx
217	248	185	Xx
184	162	194	Xx

Gambar 11. Citra Sebelum Reduksi Noise

IV. IMPLEMENTASI

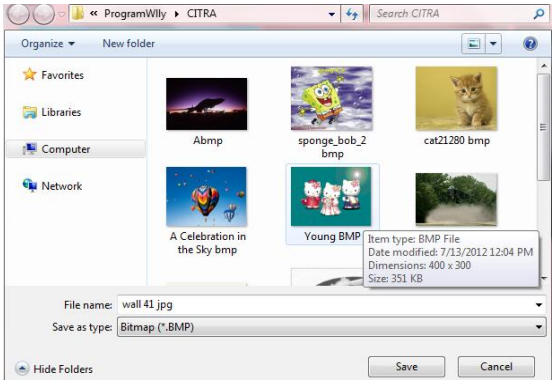
Form reduksi *noise geometric mean filter* akan ditampilkan jika *user* memilih pilihan reduksi *noise geometric mean filter* setelah memilih pilihan reduksi pada menu bar. Pada *form* citra digital yang akan di *browse* dari media penyimpanan akan di tempatkan pada image dengan mengklik tombol *browse*, kemudian *noise* akan di *generate* dengan tombol *generate noise* dan data dari citra yang sudah di *generate* akan tampil pada data citra asli dan di dapat nilai dimensi citra, kemudian *noise* akan di bersihkan dengan mengklik tombol *reduce noise*, maka diperoleh gambar hasil proses *reduice noise* pada kotak citra hasil sekaligus data citra hasil dan nilai *mean square error* (MSE). Maka gambar hasil proses *reduice noise* dapat di simpan pada media penyimpanan. Berikut gambar 12 ditampilkan

pada form menu Reduksi *Noise Geometric Mean Filter*.



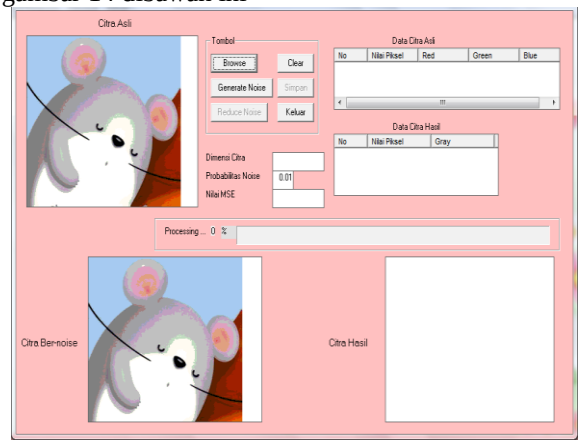
Gambar 12. *Form* Reduksi *Noise Geometric Mean Filter*

Selanjutnya untuk menyimpan hasil reduksi *noise*, pilih tombol <Save> seperti pada kotak dialog pada Gambar 13



Gambar 13. *Form* Kotak Dialog Save

Pada kotak dialog di atas, isi nama file hasil reduksi dan tekan tombol tombol <Save>. Setelah memilih tombol <Save>, maka akan tampil konfirmasi bahwa file sudah disimpan. Setelah memilih file yang akan diproses maka akan tampil tampilan seperti pada gambar 14 dibawah ini



Gambar 14 Hasil Pemilihan Citra *Geometric*

V. KESIMPULAN

Berdasarkan dari pengujian yang telah dilakukan, maka dapat disimpulkan yaitu:

1. Berdasarkan hasil pengujian reduksi noise yang terbaik adalah yang memiliki nilai rata-rata *MSE* terkecil yaitu untuk algoritma *Geometric Mean Filter*.
2. Sistem reduksi *noise* ini menggunakan dua jenis tipe gambar yaitu jpg dan bmp sehingga membutuhkan kapasitas memori yang cukup besar.

VI. DAFTAR PUSTAKA

1. Ryanto, S,2000 Rekayasa Perangkat Lunak .Penerbit Andi
2. Sutoyoso, 2009 , Komponen Komponen Pada Citra, Penerbit Bandung.
3. http://www, home.unpar.ac.id, 13, 04, 2013
4. Munir, 2002. Pengenalan Citra Digital, Penerbit Bandung.
5. Darma Putra 2010, Pengolahan Citra, Penerbit Andi Yogyakarta.
6. Jogiyanto H. M, 2005, Analisis dan Desaign, Penerbit Andi, Yogyakarta.