

IMAGE SMOOTHING MENGGUNAKAN MEAN FILTERING, MEDIAN FILTERING, MODUS FILTERING DAN GAUSSIAN FILTERING

Bambang Yuwono

Jurusan Teknik Informatika UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan 55281 Telp (0274) 485323
Email : bambangy@gmail.com

Abstract

Image smoothing aims to suppress the interference (noise) in the image. The disorder usually appears as a result of which is not good penerokan (sensor noise, photographic grain noise) or due to transmission line (the delivery of data) This research has resulted in an application program for image smoothing by four methods: mean filtering, median filtering, Gaussian filtering and filtering modes. Test images used in this study using a sample consisting of twenty 24-bit image and the image of 8 bits. The images are loaded and displayed on the program. Then the image smoothing process was done using the Gaussian method, the mean, median and mode, and displays histogramnya. Parameters measured are the result of image smoothing based on the four methods used, the signal to-noise ratio (SNR), and timing-run.

Keywords : : Image smoothing, mean filtering, median filtering, modus filtering, SNR

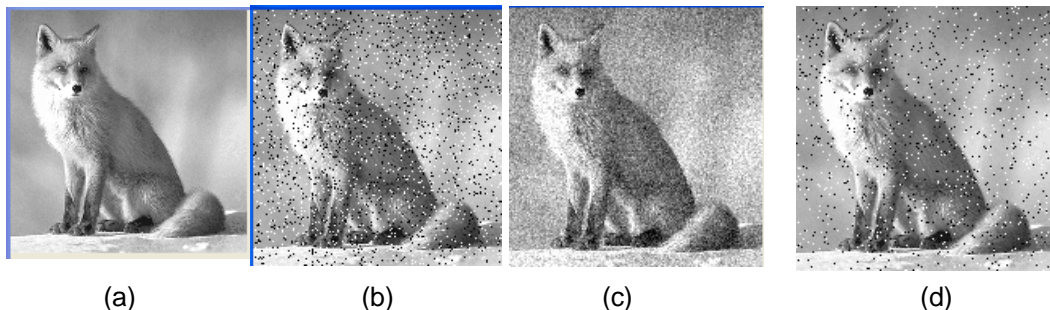
Pelembutan Citra (Image smoothing) bertujuan untuk menekan gangguan (noise) pada citra. Gangguan tersebut biasanya muncul sebagai akibat dari hasil penerokan yang tidak bagus (sensor noise, photographic grain noise) atau akibat saluran transmisi (pada pengiriman data)

Penelitian ini telah menghasilkan sebuah program aplikasi untuk image smoothing dengan empat metode yaitu mean filtering, median filtering, modus filtering dan gaussian filtering. Citra uji yang digunakan pada penelitian ini menggunakan duapuluh sampel yang terdiri dari citra 24 bit dan citra 8 bit. Citra tersebut di-load dan ditampilkan pada program. Kemudian dilakukan proses image smoothing dengan menggunakan metode gaussian, mean, median dan modus, serta menampilkan histogramnya. Parameter yang diukur adalah hasil image smoothing berdasarkan keempat metode yang digunakan, signal to-noise ratio (SNR), dan timing-run.

Kata kunci: Image smoothing, mean filtering, median filtering, modus filtering, SNR

1. PENDAHULUAN

Gangguan pada citra umumnya berupa variasi intensitas suatu pixel yang tidak berkorelasi dengan pixel-pixel tetangganya. Secara visual, gangguan mudah dilihat oleh mata karena tampak berbeda dengan pixel tetangganya. Gambar 1 adalah citra Fox yang mengalami gangguan berupa *salt and pepper* serta *gaussian* yang tampil pada gambar dalam bentuk bercak putih atau hitam seperti beras.



Gambar 1. Citra Fox yang mengalami gangguan berupa (a) Asli (b) Gaussian (c) salt and pepper (d) Speckle

Pixel yang mengalami gangguan umumnya memiliki frekuensi tinggi (berdasarkan analisis frekuensi dengan transformasi Fourier). Komponen citra yang berfrekuensi rendah umumnya mempunyai nilai pixel konstan atau berubah sangat lambat. Operasi pelembutan citra dilakukan untuk menekan komponen yang berfrekuensi tinggi dan meloloskan komponen yang berfrekuensi rendah. Ada beberapa cara atau metode pelembutan citra, diantaranya adalah mean filtering, median filtering, modus filtering dan gaussian filtering yang akan dibahas pada tulisan ini.

2. Tinjauan Pustaka

2.1 Mean Filtering

Mean adalah nilai rata-rata dari kumpulan data (Usman, 2005).

Rumusan menghitung mean adalah:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i \quad \dots\dots\dots 1)$$

Keterangan :

\bar{X} = Nilai rata-rata (*Mean*)

n = Jumlah data

x_i = Nilai ke -i

i = Nilai Awal

Mean filtering yang digunakan untuk efek *smoothing* ini merupakan jenis *spatial filtering*, yang dalam prosesnya mengikutsertakan piksel-piksel disekitarnya. Piksel yang akan diproses dimasukkan dalam sebuah matrik yang berdimensi N X N. Ukuran N ini tergantung pada kebutuhan, tetapi nilai N haruslah ganjil sehingga piksel yang diproses dapat diletakkan tepat ditengah matrik. Sebagai contoh matrik berdimensi 3 X 3 seperti gambar 2.3 di bawah ini:

1	2	3
4	T	5
6	7	8

Gambar 2.1 Matrik *mean filtering*

Nilai 1, 2, 3, 4, 5, 6, 7, dan 8 pada gambar 2.1 adalah piksel-piksel disekitar piksel T yang akan diproses. Nilai 4 didapat dari piksel sebelah kiri dari piksel T, nilai 5 didapat dari piksel di sebelah kanan dari piksel T, proses pengambilan piksel dimulai dengan mengambil piksel yang akan diproses, disimpan dalam nilai T. Kemudian diambil piksel-piksel sekiranya sehingga matrik terisi penuh. Proses selanjutnya dijumlahkan semua nilai yang terdapat pada matrik tersebut. Hasil penjumlahan tersebut dibagi dengan jumlah titik yang terdapat pada matrik tersebut. Bilangan pembagi ini dapat diperoleh dari perkalian antara N X N. Pada gambar 2.1, maka hasil pembagiannya adalah 9. Sembilan diperoleh dari hasil kali matrik 3 X 3. Hasil pembagian tersebut akan menggantikan nilai T. Nilai T yang baru akan ditampilkan pada layar monitor untuk menggantikan nilai T yang lama.

Proses diatas adalah untuk menggambar *grayscale* (hitam-putih), untuk menggambar berwarna maka masing-masing titik terlebih dahulu ditentukan nilai warna merah (*R*), hijau (*G*), dan biru (*B*). Masing-masing nilai *RGB* dijumlahkan. Hasil penjumlahan nilai *RGB* dibagi dengan jumlah titik yang diproses. Hasil dari pembagian digunakan untuk menentukan warna baru yang akan di letakkan pada titik T.

2.2 Modus Filtering

Modus adalah nilai *variable* yang memiliki *frekuensi* tertinggi (Usman, 2005). *Modus* dapat ditemukan pada data yang telah diurutkan maupun yang belum terurut. Untuk menentukan modus filter ini, pertama-tama ditentukan piksel utama yang akan diproses dari piksel-piksel disekitarnya. Pada *modus filtering* digunakan matrik berdimensi N X N. Dari matrik tersebut, kemudian data yang ada diurutkan dan dimasukkan dalam sebuah matrik berukuran 1X(N X N). Hal ini berguna untuk mempermudah menemukan *modusnya* untuk pencarian nilai yang terbanyak *frekuensinya* dari kumpulan data yang telah urut tersebut

Sebagai contoh, jika diketahui suatu piksel utama dan piksel-piksel disekitarnya adalah:

5	7	2
7	4	7
2	7	1

Gambar 2.2 Matrik untuk *modus filtering*

Maka data pada gambar 2.2 harus diurutkan terlebih dahulu sehingga menjadi:

1	2	2	4	5	7	7	7	7
---	---	---	---	---	---	---	---	---

Gambar 2.3 Matrik *modus filtering* setelah diurutkan

Sehingga dari gambar 2.3 dapat diketahui bahwa *modus* dari matrik tersebut adalah 7. Nilai 7 akan menggantikan piksel utama yang semula bernilai 4.

2.3 Median Filtering

Median adalah nilai tengah dari kumpulan data (Usman, 2005). Untuk mencari *median* dari kumpulan data yang ganjil maka:

$$x = \frac{n + 1}{2}$$

Keterangan:

- n = Jumlah data
- x = Nilai baru *median*

Untuk *median filtering* ini, data yang digunakan untuk menghitung *median* terdiri dari kumpulan data yang ganjil. Hal ini disebabkan dengan jumlah data yang ganjil maka piksel yang akan diproses dapat berada ditengah. Pada *median filtering* digunakan matrik berdimensi N X N. Dari matrik tersebut, kemudian data yang ada diurutkan dan dimasukkan dalam sebuah matrik berukuran 1X (N X N). Hal ini berguna untuk mempermudah menemukan *median* dari kumpulan data yang telah urut tersebut

Sebagai contoh jika diketahui suatu matrik berdimesi 3X3 yang berisi piksel utama dan piksel-piksel disekitarnya :

9	5	5
3	8	5
2	1	4

Gambar 2.4 matrik untuk *median filtering*

Matrik diatas harus diurutkan terlebih dahulu dan dimasukkan dalam sebuah matrik yang berukuran 1X (3X3) atau 1X9.

1	2	3	4	5	5	5	8	9
---	---	---	---	---	---	---	---	---

Gambar 2.5 Matrik untuk *median filtering* setelah diurutkan

Dari gambar 2.5 dapat dicari nilai piksel yang baru dengan menggunakan perhitungan median, maka nilai mediannya adalah x = 5. Nilai 5 ini akan menggantikan nilai 8 sehingga piksel utamanya akan memiliki warna yang berbeda dengan sebelumnya.

2.4 Gaussian Filtering

Gaussian filtering didapat dari operasi konvolusi. Operasi perkalian yang dilakukan ialah perkalian antara matriks *kernel* dengan matriks gambar asli. Matriks kernel *gauss* didapat dari fungsi komputasi dari distribusi *gaussian*, seperti pada persamaan di bawah ini:

$$G(i, j) = c.e^{-\frac{(i - u)^2 + (j - v)^2}{2\sigma^2}} \dots\dots\dots 4)$$

Keterangan :

c dan σ = konstanta

G (i,j) = elemen matriks kernel gauss pada posisi (i,j)

(u,v) = indeks tengah dari matriks kernel gauss

Berikut ini contoh matrik kernel Gauss 3 X 3 dengan $\sigma = 1.0$

1	2	1
2	3	2
1	2	1

Gambar 2.6 Matrik kernel gauss 3 X 3 dengan $\sigma = 1.0$

Perkalian antara bobot matriks gambar asli dengan bobot matrik *kernel gauss* dapat dirumuskan seperti pada di bawah ini:

$$Pixel\ B(i,j) = \frac{1}{K} \cdot \sum_{p=0}^{N-1} \left(\sum_{q=0}^{M-1} G(p,q) \cdot PixelA \left(i + p - \frac{(N-1)}{2}, j + q - \frac{(M-1)}{2} \right) \right) \dots 5$$

Keterangan :

Pixel A = gambar A (Gambar Asli)

Pixel B(i,j) = bobot hasil perkalian pada posisi (i,j)

N = jumlah kolom matriks kernel

M = jumlah baris matriks kernel

K = penjumlahan semua bobot di G

G(p,q) = elemen matriks kernel gauss pada posisi (p,q)

Gambar yang akan diproses dibagi menjadi 2 jenis piksel, yaitu piksel batas dan piksel dalam.

Piksel batas yaitu piksel yang berada dipaling luar pada gambar, selain piksel tersebut disebut piksel dalam.

Untuk piksel yang berada di dalam, perkalian dilakukan menggunakan rumus no 5, yaitu menjadikan piksel yang dicari nilai barunya sebagai piksel tengah dan bobotnya dikalikan dengan bobot pada piksel tengah matrik kernel, lalu dijumlahkan dengan hasil perkalian antara bobot piksel-piksel tetangga-tetangganya dengan bobot piksel matrik kernel. Untuk piksel yang berada disudut atau perbatasan, sebelum dilakukan perkalian, sebelumnya harus mencari bobot pada piksel-piksel luar (*dummy*). Bobot piksel-piksel ini dicari dengan menggunakan *interpolasi* yaitu dengan melihat dua piksel di dekatnya yang searah (*horizontal* atau *vertikal*). Apabila ada piksel yang memiliki bobot lebih kecil dari 0 maka bobot dijadikan 0. Apabila ada piksel yang memiliki bobot lebih besar dari 255 maka bobotnya dijadikan 255.

Contoh *konvolusi* matrik gambar asli(A) dengan matrik kernel gauss(G):

0	5	70	45	0	0	0	X	1	2	1	=	390	6585	510	325	215
0	20	55	30	10	5	0		2	3	2		440	570	523	503	483
30	35	40	15	60	45	30		1	2	1		529	660	602	579	615
0	15	50	85	23	37	51		G				644	663	556	468	568
44	52	66	16	33	44	55						627	468	415	435	649
46	47	28	36	7	51	3										
0	62	0	56	0	56	0										

A

B

$\frac{1}{15}$	X	390	6585	510	325	215	=	26	39	34	22	14
440		570	523	503	483	29		38	35	34	32	
529		660	602	579	615	35		44	40	39	47	
644		663	556	468	568	43		44	37	31	38	
627		468	415	435	649	42		31	28	29	43	

C

Keterangan:

A = Matriks gambar asal

B = Matrik hasil perkalian

G = Matrik kernel gauss C = Matrik gambar hasil

2.5 Running-time

Running-time adalah waktu dari awal proses dimulai hingga akhir proses. Waktu mulai dan waktu akhir bekerja secara bersamaan untuk menghitung total waktu yang diperlukan yang ditampilkan dalam satuan detik. Semakin kecil nilai *running-time* semakin cepat waktu yang digunakan untuk proses, dan semakin besar nilai *running-time* semakin lama waktu yang digunakan untuk proses (Nalwan,1997). *Running-time* digunakan untuk mengetahui total waktu yang diperlukan untuk sebuah proses.

2.6 SNR (signal-to-noise ratio)

SNR digunakan untuk menentukan kualitas citra setelah dilakukan operasi pengurangan derau. Citra hasil dibandingkan dengan citra asli untuk memberi perkiraan kasar kualitas citra hasil. Semakin besar nilai SNR berarti pengurangan derau dapat meningkatkan kualitas citra, sebaliknya jika nilai SNR semakin kecil maka pada citra hasil hanya sedikit juga peningkatan kualitasnya (Basuki, 2005).

Nilai SNR yang tinggi adalah lebih baik karena berarti rasio sinyal terhadap derau juga tinggi, dimana sinyal adalah citra asli. SNR biasanya diukur dengan satuan *decibels* (dB).

Rumus untuk menghitung SNR dapat dilihat dalam persamaan berikut:

$$SNR = 10 \cdot \text{Log}_{10} \frac{\sum_{m,n} I_{m,n}^2}{\sum_{m,n} (I_{m,n} - \bar{I}_{m,n})^2} \dots\dots\dots 9)$$

dimana :

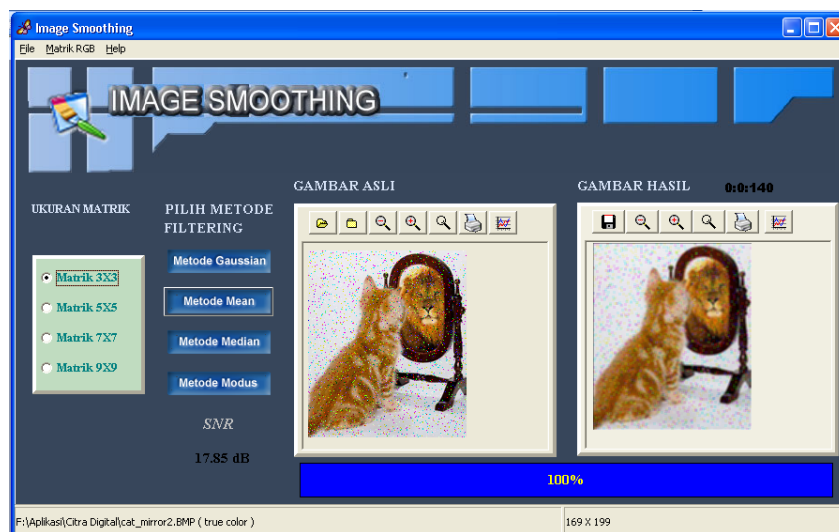
$I_{m,n}$ adalah citra asli

$\bar{I}_{m,n}$ adalah citra hasil

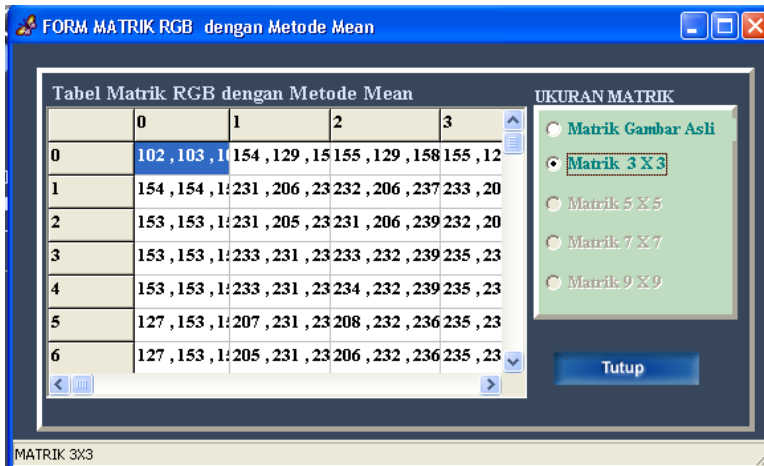
m,n adalah ukuran citra

3. Hasil Dan Pembahasan

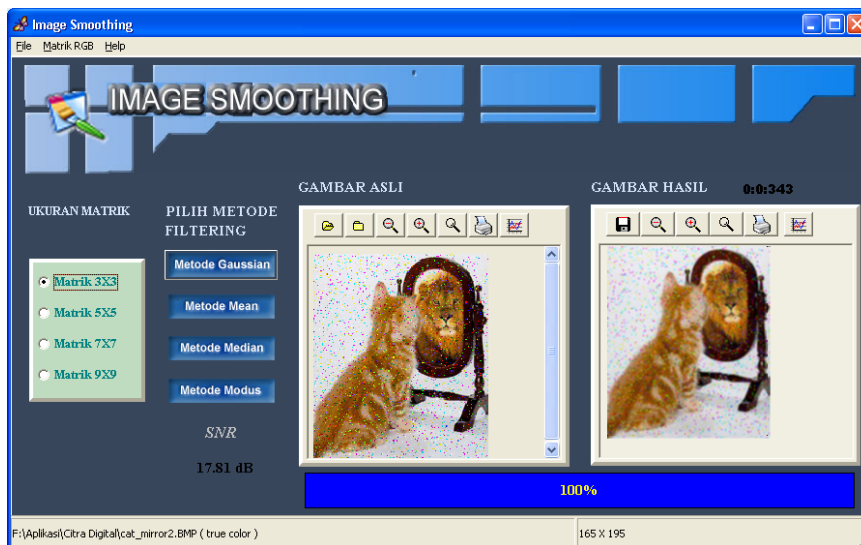
Penelitian ini telah menghasilkan sebuah aplikasi untuk *image smoothing* dengan menggunakan metode gaussian filtering, mean filtering, median filtering dan modus filtering. Ukuran matrik kernelnya terdiri 3x3, 5x5, 7x7 dan 9x9. Aplikasi ini juga dilengkapi dengan perhitungan SNR dan histogram. Berikut hasil-hasil pengujian programnya.



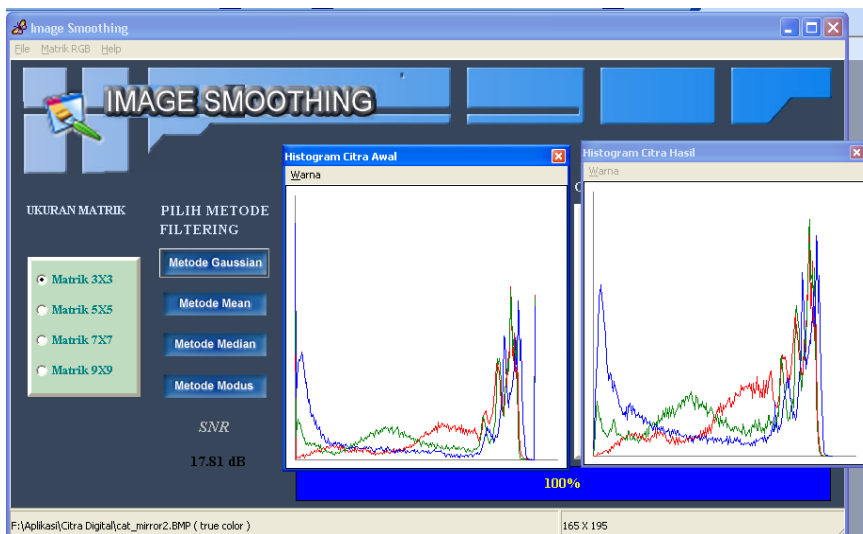
Gambar 3.1 *Image Smoothing* Menggunakan Metode Mean



Gambar 3.2 Matrik hasil metode Mean



Gambar 3.3 Smoothing Menggunakan Metode Gaussian



Gambar 3.4 Histogram sebelum dan sesudah smoothing gaussian

Analisis hasil *smoothing* berdasarkan kualitas gambar

Dari keempat metode yang digunakan, didapat perbedaan kualitas gambar dari setiap metode yang digunakan. Seperti yang terlihat pada gambar 4.14 berikut ini:



Gambar asli

Metode gaussian

Metode mean

Metode median

Metode modus

Gambar 3.5 Perbandingan Gambar Berdasarkan Metode Filtering Menggunakan Ukuran Matrik 3 X 3

Dari keempat metode yang digunakan yaitu: *Gaussian*, *Mean*, *Median*, dan *Modus* didapat gambar yang memiliki kualitas terbaik setelah dilakukan proses *smoothing* yaitu gambar yang diproses menggunakan metode *median*. Dengan menggunakan metode *median* gambar yang diproses akan mengalami penghalusan gambar secara sempurna. Dengan menggunakan metode *gaussian*, gambar hasil *smoothing* hampir sama dengan gambar asli yaitu masih memiliki derau meskipun gambar sudah mengalami penghalusan. Dengan menggunakan metode *mean* gambar hasil *smoothing* sama dengan gambar hasil metode *Gaussian* tetapi berubah dari gambar aslinya. Dengan menggunakan metode *modus* gambar yang dihasilkan memiliki titik-titik *noise* yang semakin besar.

Analisis hasil *smoothing* berdasarkan *Timing-Run*

Timing-run ini adalah lama waktu proses *smoothing* pada suatu citra, analisis ini menggunakan duapuluh sampel sehingga didapat hasil yang lebih akurat. Dari keempat metode yang digunakan, didapat selisih waktu untuk melakukan proses *smoothing*. Seperti yang terlihat pada tabel di bawah ini:

Tabel 3.1 Data Perbandingan *Timing-run* dari setiap metode *filtering* Berdasarkan ukuran matrik 3 X 3.

Nama Citra	Waktu smoothing dengan metode gaussian (detik)	Waktu smoothing dengan metode mean (detik)	Waktu smoothing dengan metode median (detik)	Waktu smoothing dengan metode modus (detik)
Butterfly.bmp 24 bit	1.883	0.591	0.861	0.982
Kuda.bmp 24 bit	1.462	0.640	0.741	0.922
Hiu.bmp 24 bit	1.663	0.611	1.2	0.931
Sun_flower.bmp 24 bit	1.502	0.661	0.951	0.881
Cat_mirror 24 bit	1.81	0.450	0.751	0.651
Cow.bmp 8 bit	1.312	0.551	0.762	0.862
Elephant.bmp 8 bit	1.923	0.620	1.101	1.82
Fox.bmp 8 bit	0.881	0.430	0.511	0.671
Nemo.bmp 8 bit	1.763	0.561	0.961	1.41
Cat_fish 8 bit	1.61	0.581	0.711	0.781
Bugline_noise.bmp8 bit	1.763	0.601	0.862	1.1
Goldhiil.bmp 8 bit	3.896	0.931	1.522	1.802
Penguin.bmp 24 bit	1.41	0.481	0.611	0.671
Saturn.bmp 24 bit	1.432	0.540	0.771	0.841
Sphinx.bmp 8 bit	1.282	0.530	0.681	0.782
Stonehenge.bmp24 bit	1.533	0.551	0.781	0.852
Freeze.bmp 24.bit	1.501	0.561	0.801	0.882
Flower.bmp 8 bit	1.482	0.561	0.751	0.851
Maria.bmp 24 bit	2.734	0.761	1.231	1.402

Butterfly.bmp 8 bit	1.723	0.611	0.842	0.961
---------------------	-------	-------	-------	-------

Dari data *timing-run* di atas, dapat dihitung nilai rata-rata dari tiap metode. Dengan menjumlahkan nilai tiap citra dari suatu metode lalu membaginya dengan jumlah sampel citra yang ada, sehingga didapat nilai rata-ratanya yaitu:

Tabel 3.2 Nilai Rata-Rata *Timing-run Smoothing*

Metode Smoothing	Waktu Proses Rata-Rata (detik)
Metode Gaussian	1.72705
Metode Mean	0.5912
Metode Median	0.87015
Metode Modus	1.0468

Berdasarkan pada tabel di atas maka diperoleh *timing-run* terbaik dari keempat metode adalah *timing-run* dengan menggunakan metode *mean filtering*.

Perbedaan waktu dari keempat proses metode *smoothing* sangat besar. Data *timing-run* menunjukkan, proses *smoothing* dengan menggunakan metode *mean* membutuhkan waktu lebih cepat dari pada ketiga metode lainnya. Hal ini terjadi karena metode *gaussian*, *median* dan *modus* bekerja dengan mendeteksi dan menghitung terlebih dahulu jumlah frekuensi pemakaian tiap nilai warna yang sama. Apabila terdapat titik yang berada pada range tersebut akan dianggap sebagai *noise*, yang menyebabkan proses bekerja dua kali lebih lama. Tentu saja hal ini membutuhkan waktu yang lebih lama dibandingkan dengan metode *mean* yang hanya mendeteksi sinyal tinggi dan sinyal rendah nya saja.

Analisis hasil *smoothing* berdasarkan SNR

Sama halnya dengan *timing-run* di atas, analisis ini juga menggunakan duapuluh sampel yang digunakan dalam uji coba program ini, agar memperoleh hasil yang lebih akurat. SNR dari masing-masing citra uji yang merupakan hasil *smoothing* dapat dilihat pada tabel di bawah ini.

Tabel 3.3 Perbandingan SNR dari setiap metode *filtering* berdasarkan ukuran matrik 3 X 3.

Nama Citra	SNR smoothing dengan metode gaussian (db)	SNR smoothing dengan metode mean (db)	SNR smoothing dengan metode median (db)	SNR. smoothing dengan metode modus (db)
Butterfly.bmp 24 bit	18.13	17.92	17.88	16.37
Kuda.bmp 24 bit	18.05	17.82	17.78	17.48
Hiu.bmp 24 bit	17.94	17.72	17.66	17.17
Sun_flower.bmp 24 bit	18.08	17.87	17.80	16.50
Cat_mirror 24 bit	18.07	17.85	17.80	17.26
Cow.bmp 8 bit	17.84	17.61	17.57	17.23
Elephant.bmp 8 bit	18.30	18.10	18.02	17.11
Fox.bmp 8 bit	18.25	18.04	17.96	17.56
Nemo.bmp 8 bit	18.21	18.00	17.94	17.48
Cat_fish 8 bit	18.15	17.92	17.88	17.23
Bugline_noise.bmp 8 bit	18.22	18.02	17.93	17.56
Goldhiil.bmp 8 bit	17.97	17.77	17.70	17.36
Penguin.bmp 24 bit	17.86	17.62	17.59	17.22
Saturn.bmp 24 bit	17.54	17.31	17.30	16.85
Sphinx.bmp 8 bit	18.31	18.08	18.01	17.63
Stonehenge.bmp 24 bit	17.99	17.75	17.70	17.43
Freeze.bmp 24 bit	17.95	17.73	17.67	17.20
Flower.bmp 8 bit	18.06	17.86	17.79	17.55
Maria.bmp 24 bit	18.30	18.09	18.02	17.16
Butterfly.bmp 8 bit	17.54	17.35	17.29	16.56

Dari data SNR di atas, dapat dihitung nilai rata-rata dari tiap metode. Dengan menjumlahkan nilai tiap citra dari suatu metode lalu membaginya dengan jumlah sampel citra yang ada, sehingga didapat nilai rata-rata nya yaitu:

Tabel 3.4 Nilai Rata-Rata SNR

Metode Smoothing	SNR Proses Rata-Rata (db)
Metode Gaussian	18.038
Metode Mean	17.8215
Metode Median	17.766
Metode Modus	17.1985

Dalam kasus ini nilai rata-rata SNR yang terbaik adalah nilai yang lebih besar, karena nilai SNR didapat dari nilai kualitas citra filter dibandingkan nilai kualitas citra asli yang berderau. Sehingga semakin besar nilai SNR maka semakin rendah ratio sinyal terhadap noise, dimana sinyal adalah citra asli yang berderau. Berdasarkan pada tabel di atas maka diperoleh SNR terbaik dari keempat metode adalah SNR dengan menggunakan metode *gaussian*. Besar nilai SNR belum tentu menentukan citra hasil terbaik menurut indra penglihatan.

Dari data yang didapat, SNR citra hasil dibandingkan dengan citra berderau terdapat perbedaan. Perbedaan SNR juga terjadi pada proses *smoothing image* menggunakan metode *gaussian*, *mean*, *median* dan *modus* berdasarkan ukuran matrik. Hal ini terjadi karena perbedaan piksel citra uji dan kompleksitas dari gambar, sedangkan keempat metode yang digunakan bekerja pada level piksel.

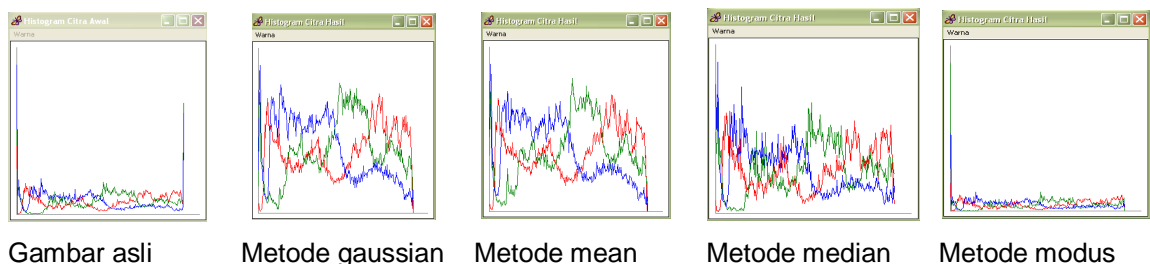
Proses *image smoothing* pada metode *gaussian*, *mean*, *median*, *modus* dengan menggunakan matrik 3 X 3 dan menghasilkan nilai SNR yang besar, hal ini berarti semakin besar peningkatan kualitas citra. Ini disebabkan karena kemungkinan banyak titik noise yang memiliki warna sama pada titik-titik gambar asli, sehingga noise dapat dengan mudah disaring.

Analisis hasil *smoothing* berdasarkan histogram

Aplikasi ini menampilkan histogram citra asli dan citra hasil. Dari keempat metode mempunyai banyak kesamaan, yaitu dari lebar histogram yang terdistribusi secara merata keseluruhan daerah, baik dari derajat keabuan maupun derajat *true-color*. Perbedaan terdapat pada daerah tumpukan histogram (gunung dan lembah). Tumpukan histogram antara citra noise sedikit lebih terdistribusi merata. Hal ini terjadi karena intensitas piksel yang dominan merata pada seluruh citra. Dari keempat metode yang digunakan, didapat perbedaan histogram dari setiap metode yang digunakan Seperti terlihat gambar 3.6.

Gambar 3.6 menunjukkan umpukan histogram citra hasil proses *smoothing* menggunakan metode *gaussian*, *mean*, *median* dan *modus* terhadap citra *noise* tidak terdapat perbedaan yang mencolok. Hal ini ditunjukkan dengan tumpukan histogram yang terdistribusi merata keseluruhan daerah derajat *true-color*, namun perbedaan terlihat pada nilai intensitas piksel pada masing-masing metode.

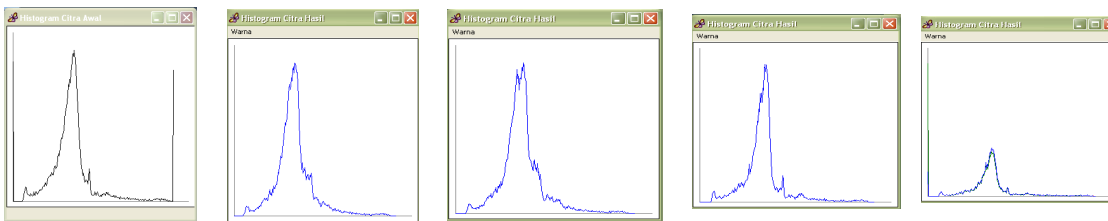
Pada metode *gaussian* dan metode *mean*, nilai intensitas piksel pada citra tinggi. Pada metode *median* nilai intensitas piksel pada citra juga tinggi tetapi memiliki warna yang merata sedangkan pada metode *modus* histogram terdistribusi secara merata keseluruhan daerah dengan distribusi yang merata pada setiap nilai intensitas piksel yang berarti tingkat kecerahan gambar tetap terjaga.



Gambar 3.6 Histogram Citra Kuda2.bmp 24 bit Berdasarkan Metode Filtering Menggunakan Ukuran Matrik 3X3

Gambar 3.7 menunjukkan tumpukan histogram citra hasil proses *smoothing* menggunakan metode *gaussian*, *mean*, *median* dan *modus* terhadap citra *noise* tidak terdapat perbedaan yang mencolok. Hal ini ditunjukkan dengan tumpukan histogram yang terdistribusi merata keseluruhan daerah derajat keabuan dengan distribusi yang merata pada setiap nilai intensitas piksel yang berarti tingkat kecerahan gambar tetap terjaga. Jadi proses *smoothing* menggunakan metode *gaussian*, *mean*, *median* dan *modus* pada citra *grayscale* tidak memberikan pengaruh pada tingkat kecerahan citra ditunjukkan dengan tumpukan histogram yang tetap terdistribusi secara merata.

Perbedaan pada puncak histogram yang nampak terjadi terdapat sewaktu menggunakan metode *modus*. Pada metode ini puncak histogram lebih rendah dari pada puncak histogram citra asli, ini mengakibatkan nilai intensitas piksel menurun yang mengakibatkan gambar tersebut buram.



Gambar asli Metode gaussian Metode mean Metode median Metode modus

Gambar 4.7 Histogram Citra Cow.bmp 8 bit Berdasarkan Metode Filtering Menggunakan Ukuran Matrik 3X3

4. Kesimpulan

Kesimpulan dari hasil penelitian adalah :

1. Secara garis besar *timing-run* untuk melakukan proses *smoothing* yang lebih cepat adalah metode *mean*. Hal ini terjadi karena metode *mean* hanya mendeteksi intensitas warna piksel pada citra yang meloloskan sinyal tinggi dan sinyal rendah nya saja.
2. Median filtering memberikan hasil yang lebih baik dibandingkan mean filtering untuk citra yang mengalami gangguan dalam bentuk spike berupa bercak-bercak putih maupun dalam bentuk salt and pepper. Bila ukuran mask diperbesar, maka derau semakin banyak dihilangkan, tetapi bersamaan dengan itu informasi detail citra juga ikut dihilangkan sehingga citra nampak kabur. Sebaliknya dengan semakin kecilnya ukuran mask, informasi detail citra dapat dipertahankan tetapi derau juga tetap dipertahankan.
3. Berdasarkan pada pengujian maka diperoleh SNR terbaik dari keempat metode adalah SNR dengan menggunakan metode *gaussian*. Besar nilai SNR belum tentu menentukan citra hasil terbaik menurut indra penglihatan

Daftar Pustaka

- Andeswari, D, 2003, Proses Peningkatan Mutu citra Menggunakan Borland Delphi 5.0, Universitas Ahmad Dahlan, Yogyakarta.
- Ansari, T, Filtering and segmentation, www.cse.ohio-state.com, 27-September-2007, 10.00 WIB.
- Balza, A dan Firdausy, K, 2005, Teknik Pengolahan Citra Digital, Ardi Publishing, Yogyakarta.
- Basuki, A, 2005, Pengolahan Citra Digital Menggunakan Visual Basic, Graha Ilmu, Yogyakarta.
- Fitri, A, Perbandingan metode low pass filter dan median filter dalam penghalusan citra untuk meningkatkan kualitas citra, www.digilab.unikom.ac.id, 11-September-2007, 14.30 WIB.
- Husni, 2004, Pemrograman Dengan Delphi, Yogyakarta, Penerbit Graha Ilmu.
- Jain, Anil K, 1989, Fundamentals of digital image processing, London, Prentice-Hall International.
- Kiswanto H, Aplikasi Perbandingan Image Smoothing menggunakan empat metode filtering pada pengolahan citra, Informatika UPN, Yogyakarta

- Munir. R, 2002, Algoritma & Pemrograman, Bandung, Penerbit Informatika.
- Munir. R, 2004, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Informatika Bandung, Bandung.
- Nalwan. A, 1997, Pengolahan Gambar Secara Digital, PT Elex Media Komputindo, Jakarta.
- Pranata. Antony, 2001, Pemrograman Borland Delphi. Andi, Yogyakarta.
- Perwitasari. E.W, 2006, Analisis Perbandingan Metode Transformasi Wavelet Dengan Metode Contour Untuk Pengurangan Derau Pada Suatu Citra, Universitas Ahmad Dahlan, Yogyakarta.
- Smoothing Operation, www.ph.tn.fudelff.nl, 11-September-2007, 15.00 WIB.