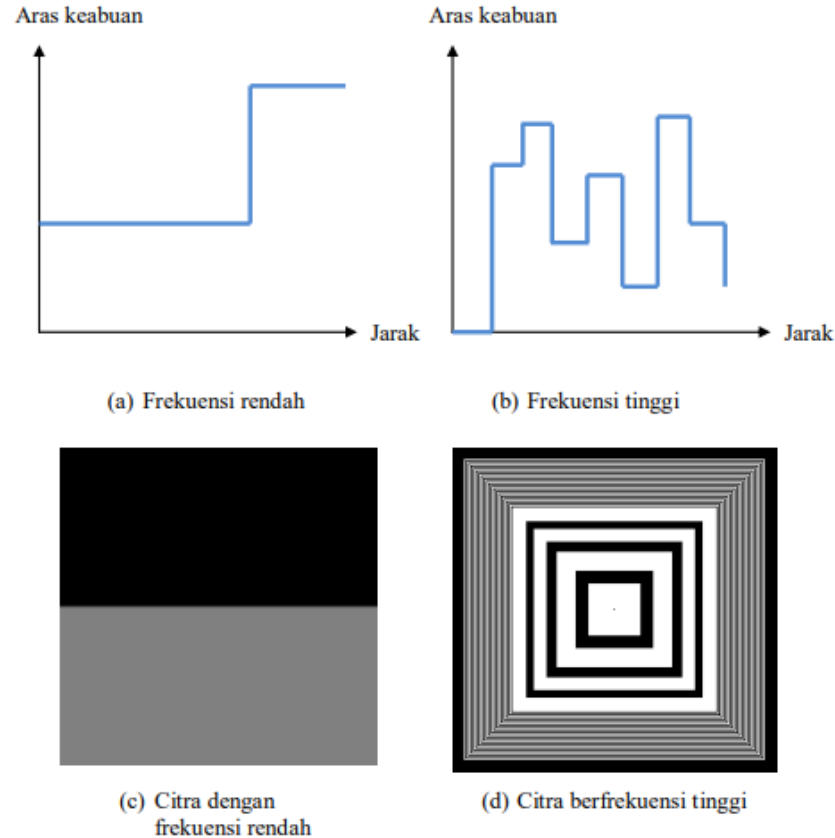


OPERASI KETETANGGAAN PIKSEL PADA DOMAIN FREKUENSI

PERTEMUAN KE 8

Perbedaan frekuensi pada citra



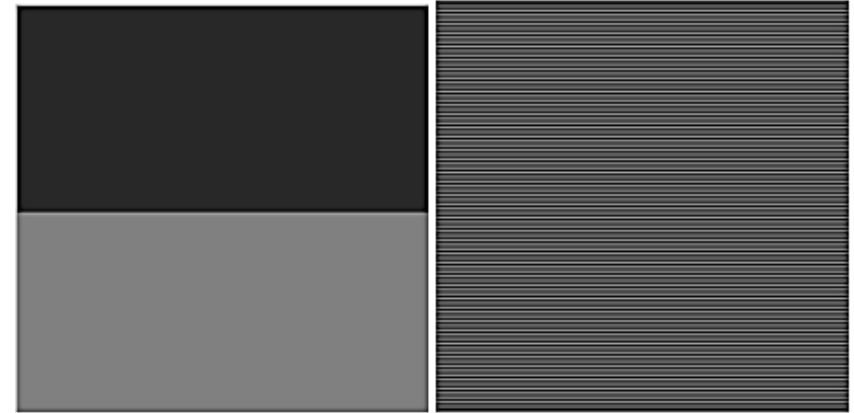
Gambar 8.1 Perbedaan frekuensi rendah dan frekuensi tinggi pada citra

intensitas dengan frekuensi rendah dan frekuensi tinggi

40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40

(a) Citra dengan frekuensi rendah

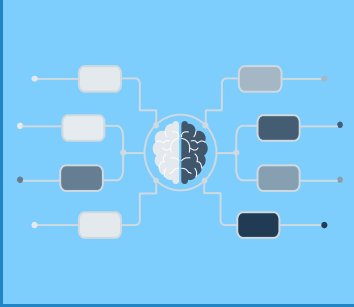
(b) Citra dengan frekuensi tinggi



(c) Citra dengan frekuensi rendah

(d) Citra dengan frekuensi tinggi

Gambar 8.2 Nilai-nilai intensitas dengan frekuensi rendah dan frekuensi tinggi



LOW PASS FILTER

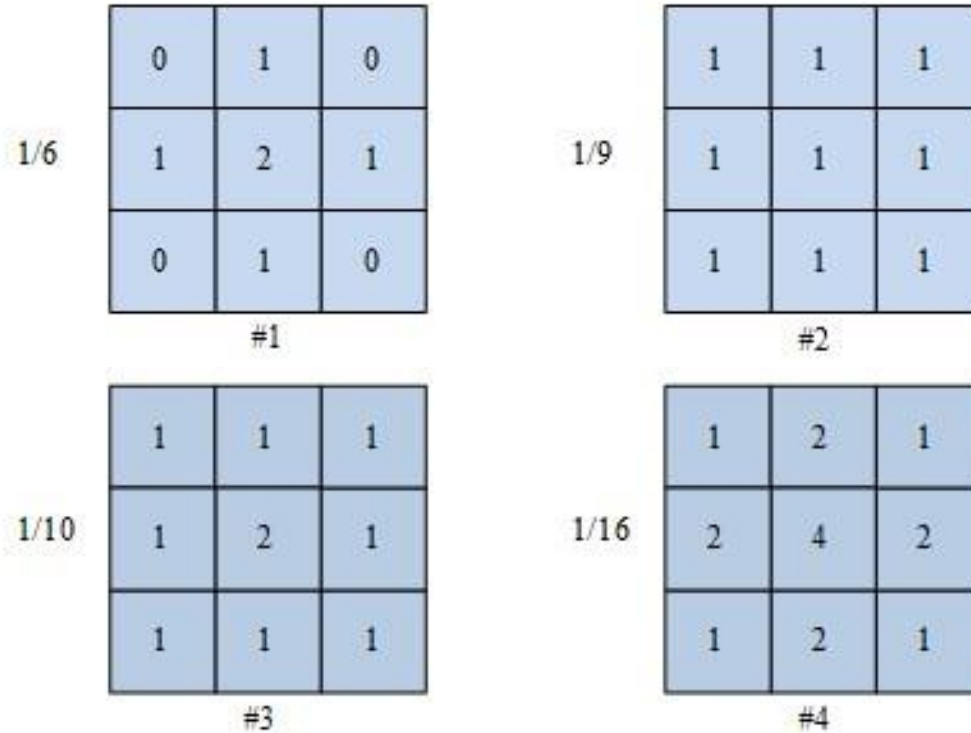
FILTER

LOLOS RENDAH



Filter lolos-bawah (*low-pass filter*) adalah filter yang mempunyai sifat dapat meloloskan yang berfrekuensi rendah dan menghilangkan yang berfrekuensi tinggi. Efek filter ini membuat perubahan aras keabuan menjadi lebih lembut. Filter ini berguna untuk menghaluskan derau atau untuk kepentingan interpolasi tepi objek dalam citra.

Contoh kernel untuk filter lolos bawah



Gambar 8.3 Contoh kernel filter lolos rendah

Kernel yang digunakan untuk pengkaburan citra, seperti berikut :

- Tinggi dan lebar kernel ganjil
- Bobot dalam kernel bersifat simetris terhadap piksel pusat
- Semua bobot bernilai positif
- Jumlah keseluruhan bobot sebesar satu



Efek pengkaburan citra dapat ditingkatkan dengan menaikkan ukuran kernel



intensitas dengan frekuensi rendah dan frekuensi tinggi

40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128

(a) Citra dengan frekuensi rendah

(b) Citra dengan frekuensi tinggi

Dengan menggunakan kernel :

$1/9$

1	1	1
1	1	1
1	1	1

Gambar 8.4 Nilai-nilai intensitas dengan frekuensi rendah dan frekuensi tinggi dengan kernel $1/9$

Penapisan dengan frekuensi rendah terhadap kedua citra

Dengan menggunakan kernel :

$1/9$

1	1	1
1	1	1
1	1	1

Maka didapat hasil seperti pada Gambar 8.5

40	40	40
40	40	40
40	40	40
40	40	40
40	40	40
40	40	40
40	40	40
69	69	69
99	99	99
128	128	128
128	128	128
128	128	128
128	128	128
128	128	128
128	128	128
128	128	128
128	128	128
128	128	128

(a) Citra dengan frekuensi rendah

99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69
99	99	99
69	69	69

(b) Citra dengan frekuensi tinggi

Gambar 8.5 Hasil penapisan dengan filter lolos rendah

Tapis Filter Lolos Rendah pada Frekuensi Rendah

[illegible]

Gambar 8.6 Hasil perhitungan penapisan frekuensi rendah, dengan filter lolos rendah menggunakan MS Excel

Tapis Filter Lolos Rendah pada Frekuensi Tinggi

<div> <div>N27</div> <div> <div>✕</div> <div>✓</div> <div>f_x</div> </div> <div> $=((B27*\\$I\\$33/9)+(C27*\\$J\\$33/9)+(D27*\\$K\\$33/9)+(B28*\\$I\\$34/9)+(C28*\\$J\\$34/9)+(D28*\\$K\\$34/9)+(B29*\\$I\\$35/9)+(C29*\\$J\\$35/9)+(D29*\\$K\\$35/9))$ </div> </div>																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
25	CITRA FREKUENSI TINGGI													CITRA HASIL					
26	0	0	0	0	0	0	0						37	56	56	56	37		
27	0	128	128	128	128	128	0						66	99	99	99	66		
28	0	40	40	40	40	40	0						46	69	69	69	46		
29	0	128	128	128	128	128	0						66	99	99	99	66		
30	0	40	40	40	40	40	0						46	69	69	69	46		
31	0	128	128	128	128	128	0						66	99	99	99	66		
32	0	40	40	40	40	40	0						46	69	69	69	46		
33	0	128	128	128	128	128	0						66	99	99	99	66		
34	0	40	40	40	40	40	0						46	69	69	69	46		
35	0	128	128	128	128	128	0						66	99	99	99	66		
36	0	40	40	40	40	40	0						46	69	69	69	46		
37	0	128	128	128	128	128	0						66	99	99	99	66		
38	0	40	40	40	40	40	0						46	69	69	69	46		
39	0	128	128	128	128	128	0						66	99	99	99	66		
40	0	40	40	40	40	40	0						46	69	69	69	46		
41	0	128	128	128	128	128	0						66	99	99	99	66		
42	0	40	40	40	40	40	0						46	69	69	69	46		
43	0	128	128	128	128	128	0						37	56	56	56	37		
44	0	40	40	40	40	40	0						9	13	13	13	9		
45	0	0	0	0	0	0	0												
46	0	0	0	0	0	0	0												

Gambar 8.7 Hasil perhitungan penapisan frekuensi tinggi, dengan filter lolos rendah menggunakan MS Excel

Hasil penapisan dengan filter lolos rendah



(a) Citra boneka yang dilengkapi derau



(b) Hasil penapisan citra boneka



(c) Citra goldhill



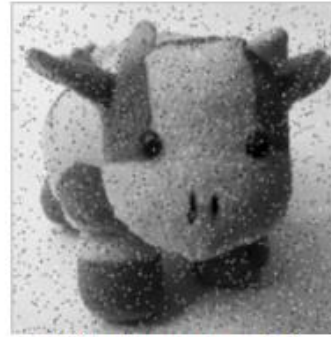
(d) Hasil penapisan citra goldhill

Gambar 8.8 Penerapan filter lolos rendah

Contoh hasil filter lolos bawah



(a) Citra boneka yang dilengkapi



(b) Hasil dengan kernel #1

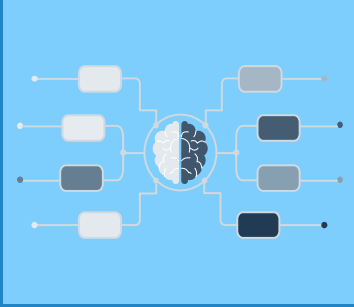


(c) Hasil dengan kernel #2



(b) Hasil dengan kernel #4

Gambar 8.9 Hasil penggunaan kernel #1, #2, #4 filter lolos bawah



HIGH PASS FILTER

FILTER

LOLOS TINGGI



Filter lolos-tinggi (*High-pass filter*) adalah filter yang mempunyai sifat dapat melewatkan frekuensi tinggi, dan menghalangi yang berfrekuensi rendah. Efek filter ini berguna untuk mendapatkan tepi objek dalam citra atau menajamkan citra.

Contoh kernel untuk filter lolos atas

0	-1	0
-1	4	-1
0	-1	0
#1		

-1	-1	-1
-1	8	-1
-1	-1	-1
#2		

1	-2	1
-2	4	-2
1	-2	1
#3		

Gambar 8.10 Kernel filter lolos tinggi

Kernel yang digunakan untuk deteksi tepi citra, seperti berikut :

- Tinggi dan lebar kernel ganjil
- Bobot dalam kernel bersifat simetris terhadap piksel pusat
- Bobot pusat kernel bernilai positif
- Bobot di sekeliling kernel bernilai negatif (dapat menggunakan 4 atau 8 ketetanggaan)
- Jumlah keseluruhan bobot lebih besar dari nol



intensitas dengan frekuensi rendah dan frekuensi tinggi

40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	40	40	40	40	40

(a) Citra dengan frekuensi rendah

(b) Citra dengan frekuensi tinggi

Dengan menggunakan kernel :

0	-1	0
-1	4	-1
0	-1	0

Gambar 8.11 Nilai-nilai intensitas dengan frekuensi rendah dan frekuensi tinggi dengan kernel HPF

Penapisan dengan frekuensi tinggi terhadap kedua citra

Dengan menggunakan kernel :

0	-1	0
-1	4	-1
0	-1	0

Maka didapat hasil seperti pada Gambar 8.12

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
88	88	88
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

(a) Citra dengan frekuensi rendah

0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176
0	0	0
176	176	176

(b) Citra dengan frekuensi tinggi

Gambar 8.12 Hasil penapisan dengan filter lolos rendah

Tapis Filter Lolos Tinggi pada Frekuensi Rendah

N3

✕

✓

f_x

=((B3*\$I\$8)+(C3*\$J\$8)+(D3*\$K\$8)+(B4*\$I\$9)+(C4*\$J\$9)+(D4*\$K\$9)+(B5*\$I\$10)+(C5*\$J\$10)+(D5*\$K\$10))

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		CITRA FREKUENSI RENDAH												CITRA HASIL				
2	0	0	0	0	0	0	0						80	40	40	40	80	
3	0	40	40	40	40	40	0						40	0	0	0	40	
4	0	40	40	40	40	40	0						40	0	0	0	40	
5	0	40	40	40	40	40	0						40	0	0	0	40	
6	0	40	40	40	40	40	0						40	0	0	0	40	
7	0	40	40	40	40	40	0						40	0	0	0	40	
8	0	40	40	40	40	40	0		0	-1	0		40	0	0	0	40	
9	0	40	40	40	40	40	0		-1	4	-1		40	0	0	0	40	
10	0	40	40	40	40	40	0		0	-1	0		-48	-88	-88	-88	-48	
11	0	40	40	40	40	40	0						216	88	88	88	216	
12	0	128	128	128	128	128	0						128	0	0	0	128	
13	0	128	128	128	128	128	0						128	0	0	0	128	
14	0	128	128	128	128	128	0						128	0	0	0	128	
15	0	128	128	128	128	128	0						128	0	0	0	128	
16	0	128	128	128	128	128	0						128	0	0	0	128	
17	0	128	128	128	128	128	0						128	0	0	0	128	
18	0	128	128	128	128	128	0						128	0	0	0	128	
19	0	128	128	128	128	128	0						256	128	128	128	256	
20	0	128	128	128	128	128	0											
21	0	0	0	0	0	0	0											
22																		
23																		

Catatan: Dalam octave atau matlab perintah **uint8**, digunakan untuk membatasi nilai dibawah 0 sampai di atas 255. Pada octave nilai di bawah 0 akan dianggap 0, sedangkan nilai diatas 255 dianggap 255. Hal ini terdapat perbedaan hasil perhitungna keluaran MS Excel, dengan keluaran pada octave.

Gambar 8.13 Hasil perhitungan penapisan frekuensi rendah, dengan filter lolos tinggi menggunakan MS Excel

Tapis Filter Lolos Tinggi pada Frekuensi Tinggi

<div> <div>N27</div> <div> <div>✕</div> <div>✓</div> <div>f_x</div> </div> <div> $=((B27*\\$I\\$32)+(C27*\\$J\\$32)+(D27*\\$K\\$32)+(B28*\\$I\\$33)+(C28*\\$J\\$33)+(D28*\\$K\\$33)+(B29*\\$I\\$34)+(C29*\\$J\\$34)+(D29*\\$K\\$34))$ </div> </div>																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
25	CITRA FREKUENSI TINGGI													CITRA HASIL				
26	0	0	0	0	0	0	0						344	216	216	216	344	
27	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
28	0	40	40	40	40	40	0						304	176	176	176	304	
29	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
30	0	40	40	40	40	40	0						304	176	176	176	304	
31	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
32	0	40	40	40	40	40	0		0	-1	0		304	176	176	176	304	
33	0	128	128	128	128	128	0		-1	4	-1		-136	-176	-176	-176	-136	
34	0	40	40	40	40	40	0		0	-1	0		304	176	176	176	304	
35	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
36	0	40	40	40	40	40	0						304	176	176	176	304	
37	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
38	0	40	40	40	40	40	0						304	176	176	176	304	
39	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
40	0	40	40	40	40	40	0						304	176	176	176	304	
41	0	128	128	128	128	128	0						-136	-176	-176	-176	-136	
42	0	40	40	40	40	40	0						304	176	176	176	304	
43	0	128	128	128	128	128	0						-8	-48	-48	-48	-8	
44	0	40	40	40	40	40	0											
45	0	0	0	0	0	0	0											
46																		
47																		

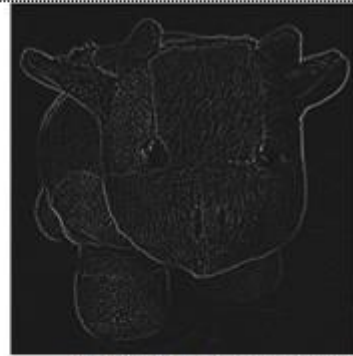
Catatan: Dalam octave atau matlab perintah **uint8**, digunakan untuk membatasi nilai dibawah 0 sampai di atas 255. Pada octave nilai di bawah 0 akan dianggap 0, sedangkan nilai diatas 255 dianggap 255. Hal ini terdapat perbedaan hasil perhitungna keluaran MS Excel, dengan keluaran pada octave.

Gambar 8.14 Hasil perhitungan penapisan frekuensi rendah, dengan filter lolos tinggi menggunakan MS Excel

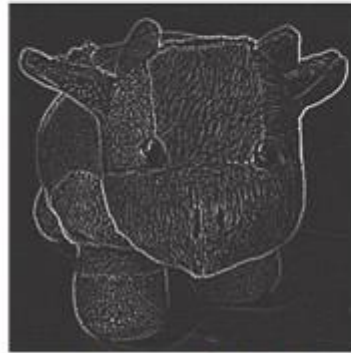
Hasil penapisan dengan filter lolos tinggi



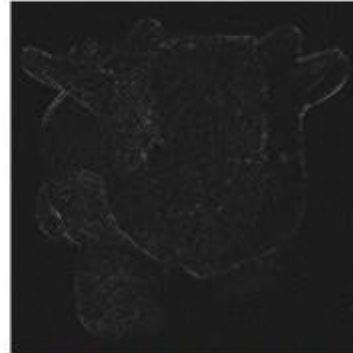
(a) Citra boneka.png



(b) Hasil dengan kernel #1

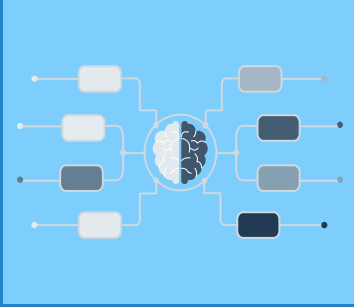


(c) Hasil dengan kernel #2



(d) Hasil dengan kernel #3

Gambar 8.15 Penerapan kernel pada filter lolos tinggi



FILTER HIGH BOOST



Penekankan komponen frekuensi tinggi (*High Boost Filtering*) adalah teknik penajaman citra melalui konvolusi. Kernel yang dapat dipakai adalah kernel filter lolos tinggi, nilai pusat diisi dengan nilai yang lebih besar dari nilai pada posisi untuk filter lolos tinggi.

Kernel yang digunakan untuk penajaman citra, seperti berikut :

- Tinggi dan lebar kernel ganjil
- Bobot dalam kernel bersifat simetris terhadap piksel pusat
- Bobot pusat kernel bernilai positif
- Bobot di sekeliling kernel bernilai negatif (dapat menggunakan 4 atau 8 ketetanggaan)
- Jumlah keseluruhan bobot lebih besar dari satu
- Bobot terbesar terletak di pusat kernel



Filter Lolos Atas Untuk Penajaman Citra

-1	-1	-1
-1	c	-1
-1	-1	-1

Nilai di pusat diisi dengan nilai yang lebih besar ($c > 8$, misalnya $c = 9$)

0	-1	0	-1	-1	-1	1	-2	1
-1	4	-1	-1	8	-1	-2	4	-2
0	-1	0	-1	-1	-1	1	-2	1
#1				#2				#3

Gambar 6.10 Kernel filter lolos tinggi

Gambar 8.16 Penerapan kernel pada filter lolos tinggi

Untuk menajamkan citra, filter high boost ini sama dengan filter lolos tinggi, namun memiliki jumlah total tidak sama dengan nol

Hasil penapisan dengan filter lolos tinggi



(a) Citra boneka



(b) Hasil untuk $c=9$

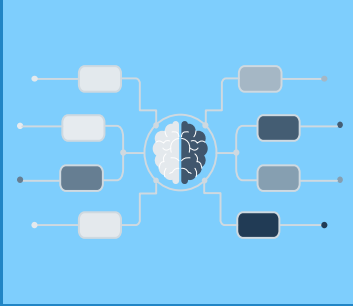


(c) Hasil untuk $c=10$



(d) Hasil untuk $c=11$

Gambar 8.17 Penerapan filter lolos tinggi, dengan nilai C menggunakan nilai 9, 10, 11



EFEK EMBOS

EFEK EMBOSS

Penebalan garis pada arah tertentu, sehingga dapat memberikan efek citra seolah diukir pada permukaan objek dalam citra, atau memberikan kesan timbul dalam citra.



(a) Berdasar citra boneka2



(b) Berdasar citra lena256

Gambar 8.18 Efek embos

Pembuatan embos terletak pada kernel konvolusi dengan sifat sebagai berikut :

- Tinggi dan lebar kernel ganjil
- Bobot dalam kernel bersifat tidak simetris terhadap piksel pusat
- Bobot pusat kernel bernilai nol
- Jumlah keseluruhan bobot bernilai nol



KERNEL YAN DIGUNAKAN UNTUK EMBOSS

-2	0	0
0	0	0
0	0	2

- Nilai negatif dan positif yang berpasangan, menentukan perubahan kecerahan yang berefek pada penggambaran garis gelap atau terang. Seperti pada Gambar 7.22.

Gambar 8.19 Kernel yang digunakan untuk embos

KERNEL YAN DIGUNAKAN UNTUK EMBOSS

-1	0	0
0	0	0
0	0	1

(a) Kernel #1



(b) Hasil untuk kernel #1

1	0	0
0	0	0
0	0	-1

(c) Kernel #2



(d) Hasil untuk kernel #2

0	0	0
-4	0	-4
0	0	0

(e) Kernel #3



(f) Hasil untuk kernel #3

Nilai negatif pada kernel embos, menentukan arah penebalan garis

Gambar 8.20 Efek embos untuk berbagai kernel

Nilai negatif pada kernel embos, menentukan arah penebalan garis

-4	-4	0
-4	1	4
0	4	4

(a) Embossing dari arah kiri atas

-6	0	6
-6	1	6
-6	0	6

(b) Embossing dari arah kiri

4	4	0
4	1	-4
0	-4	-4

(c) Embossing dari arah kanan bawah

6	0	-6
6	1	-6
6	0	-6

(d) Embossing dari arah kanan

Gambar 8.21 Berbagai kernel untuk embos



FILTER GAUSSIAN





Contoh penerapan filter *gaussian*



Gambar 8.22 Efek gaussian

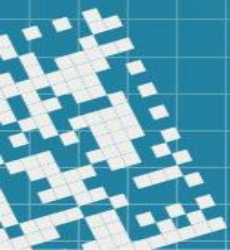
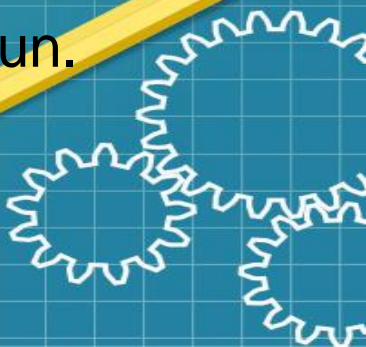

Filter *gaussian* biasanya digunakan sebagai pengolah citra agar dapat lebih halus. Gaussian filter yang banyak digunakan dalam memproses gambar. Gaussian filter bertujuan untuk menghilangkan noise pada citra dan meningkatkan kualitas detil citra.

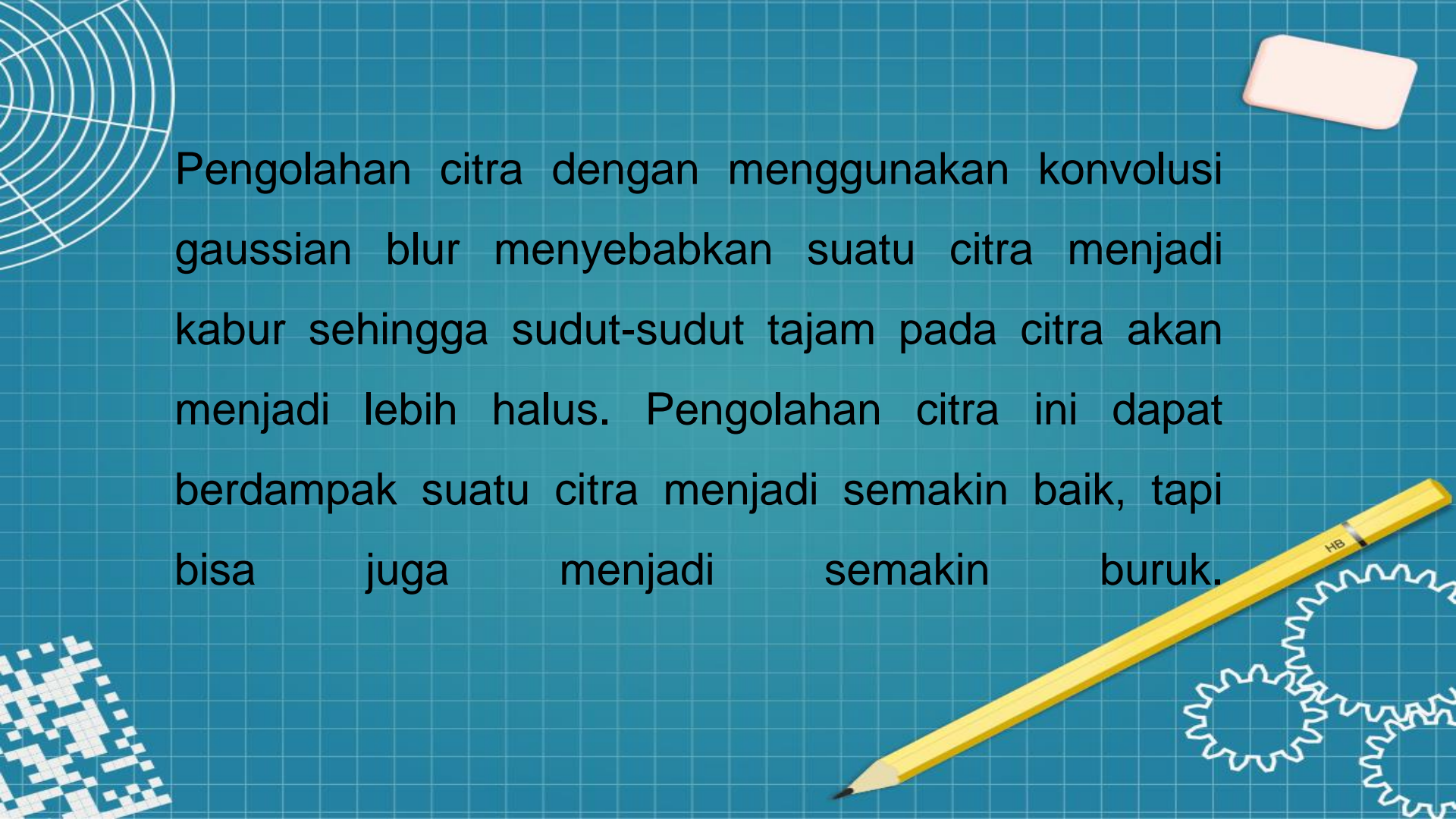
- Noise speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise. Sedangkan noise salt & pepper seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena noise.



Filter Gaussian dikategorikan sebagai filter linier, dikatakan linier karena dalam melakukan penapisan, melibatkan piksel dengan cara linier.

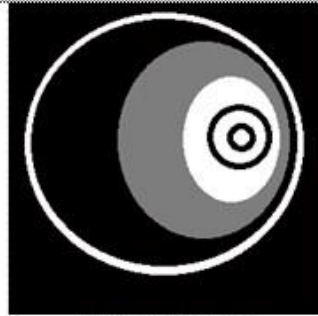
Terutama jika dipakai konvolusi citra, untuk penghilangan derau. Menyebabkan struktur pada citra meliputi titik, tepi dan garis ikut terkaburkan, menyebabkan kualitas citra keseluruhan menurun.





Pengolahan citra dengan menggunakan konvolusi gaussian blur menyebabkan suatu citra menjadi kabur sehingga sudut-sudut tajam pada citra akan menjadi lebih halus. Pengolahan citra ini dapat berdampak suatu citra menjadi semakin baik, tapi bisa juga menjadi semakin buruk.

penerapan filter *gaussian*



(a) Citra bulat.png



(b) Hasil konvolusi bulat.png

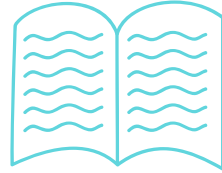


(c) Citra boneka.png



(d) Hasil konvolusi boneka.png

Gambar 8.23 Efek filter gaussian




"Filter *gaussian* tergolong sebagai filter lolos rendah"

Model dua dimensinya berupa:

$$G(y, x) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \dots\dots\dots (2)$$

Dalam hal ini σ adalah deviasi standar dan piksel pada pusat (y, x) , mendapat bobot terbesar berupa 1.

Standar deviasi adalah nilai statistik yang digunakan untuk menentukan kedekatan sampel statistik dengan rata-rata sebuah data.



Kernel filter *gaussian* berbentuk bujur sangkar, dengan mengasumsikan $\sigma^2 = 1$. Maka didapat nilai dari Kernel filter gaussian, dengan ukuran 5x5 sebagai berikut:

- $G(0,0) = e^{-0} = 1$

- $G(1,0) = G(0,1) = G(-1,0) = G(0,-1) = e^{-1/2} = 0,6065$

- $G(1,1) = G(1,-1) = G(-1,1) = G(-1,-1) = e^{-1} = 0,3679$

- $G(2,1) = G(1,2) = G(-2,1) = G(-2,-1) = e^{-5/2} = 0,0821$

- $G(2,0) = G(0,2) = G(0,-2) = G(-2,0) = e^{-2} = 0,1353$

- $G(2,2) = G(-2,-2) = G(-2,2) = G(2,-2) = e^{-4} = 0,0183$

G(1,0) =	0,6065 x 55 =	33,3575
G(1,1) =	0,3679 x 55 =	20,2345
G(2,1) =	0,0821 x 55 =	4,5155
G(2,0) =	0,1353 x 55 =	7,4415
G(2,2) =	0,0183 x 55 =	1,0065



Dengan mengatur nilai terkecil menjadi 1, maka setiap nilai dikalikan dengan 55 (nilai 55 diperoleh dari $1/0.0183$ dan kemudian hasilnya dibulatkan ke atas). Dengan mengalikan $G(y,x)$ di depan dengan 55, diperoleh hasil sebagai berikut :

1	5	7	5	1
5	20	33	20	5
7	33	55	33	7
5	20	33	20	5
1	5	7	5	1

Setelah dinormalisasi
diperoleh filter seperti
berikut:

Jml dari
keseluruhan
nilai pixel

$1/339$

$$\begin{aligned}
 G(1,0) &= 0,6065 \times 55 = 33,3575 \\
 G(1,1) &= 0,3679 \times 55 = 20,2345 \\
 G(2,1) &= 0,0821 \times 55 = 4,5155 \\
 G(2,0) &= 0,1353 \times 55 = 7,4415 \\
 G(2,2) &= 0,0183 \times 55 = 1,0065
 \end{aligned}$$

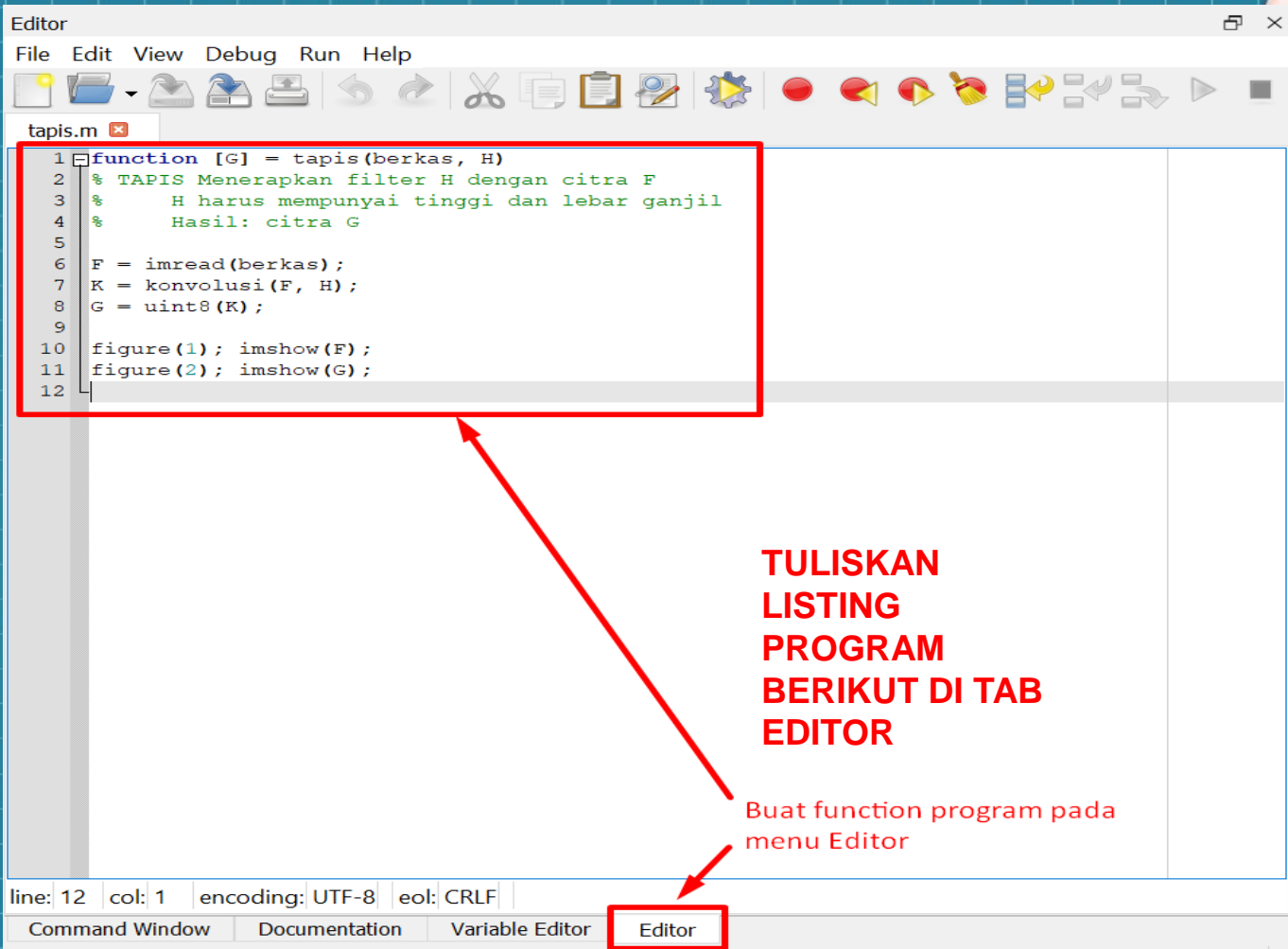
1	5	7	5	1
5	20	33	20	5
7	33	55	33	7
5	20	33	20	5
1	5	7	5	1

Gambar 8.24 Berbagai kernel untuk embosing

PRAKTIKUM



MEMBUAT FUNGSI UNTUK FILTER



**TULISKAN
LISTING
PROGRAM
BERIKUT DI TAB
EDITOR**

Buat function program pada
menu Editor

MENJALANKAN FUNGCTION UNTUK FILTER

Editor

File Edit View Debug Run Help

Jalankan Function filter

tapis.m

```
1 function [G] = tapis(berkas, H)
2 % TAPIS Menerapkan filter H dengan citra F
3 % H harus mempunyai tinggi dan lebar ganjil
4 % Hasil: citra G
5
6 F = imread(berkas);
7 K = konvolusi(F, H);
8 G = uint8(K);
9
10 figure(1); imshow(F);
11 figure(2); imshow(G);
12
```

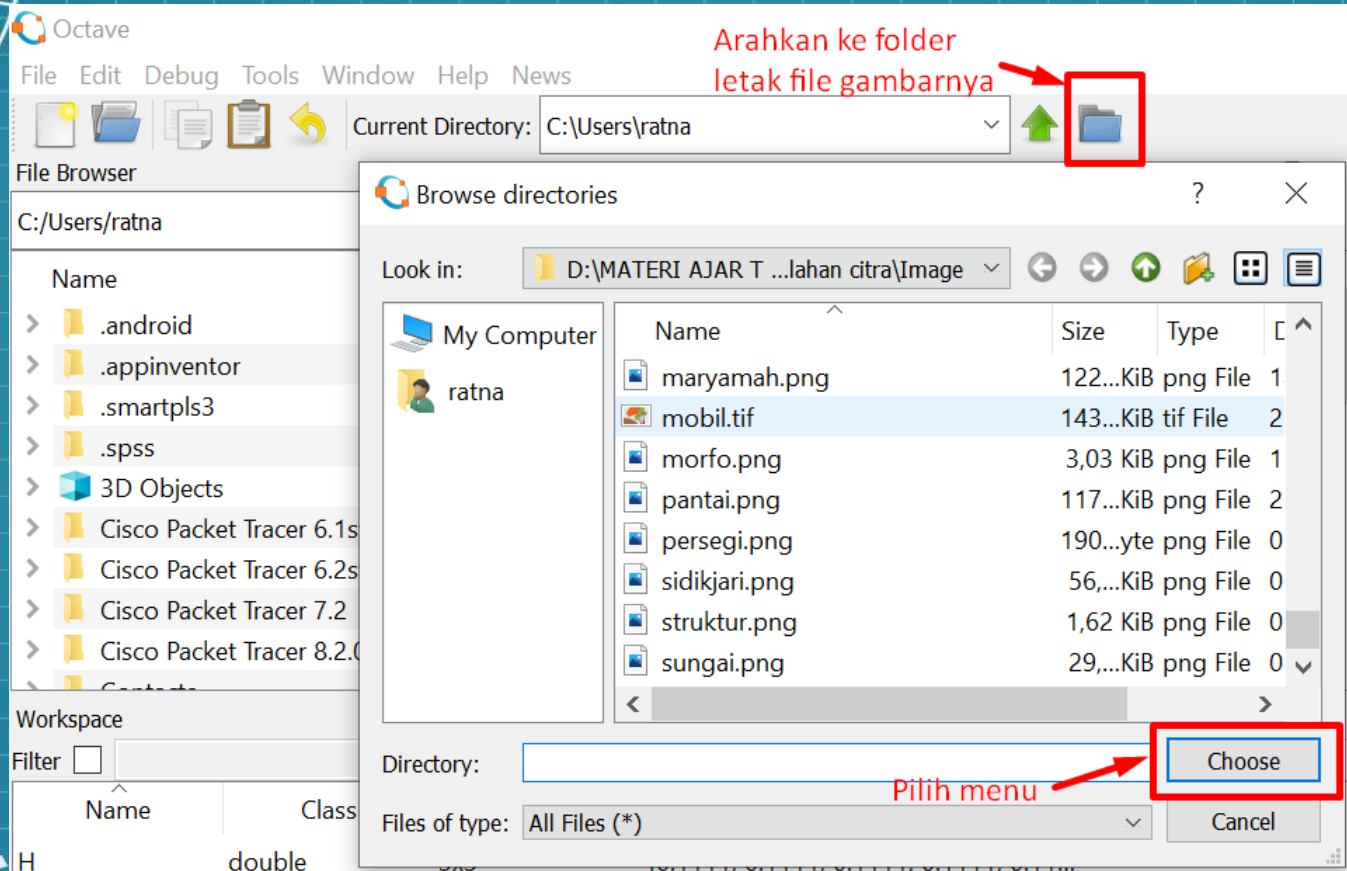
Change Directory or Add Directory to Load... ?

The file D:/MATERI AJAR T ELEKTRO/CITRA DIGITAL/cd penyerta pengolahan citra/Skrip/tapis.m does not exist in the load path. To run or debug the function you are editing, you must either change to the directory D:/MATERI AJAR T ELEKTRO/CITRA DIGITAL/cd penyerta pengolahan citra/Skrip or add that directory to the load path.

Pilih menu

Change Directory Add Directory to Load Path Cancel

MENGARAKAN KE FOLDER GAMBAR

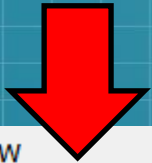


FILTER LOLOS RENDAH

TULISKAN LISTING PROGRAM BERIKUT DI COMMAND WINDOWS

```
>>% Contoh Pengujian Program diatas untuk  
>>%melakukan filter lolos rendah pada citra mobil
```

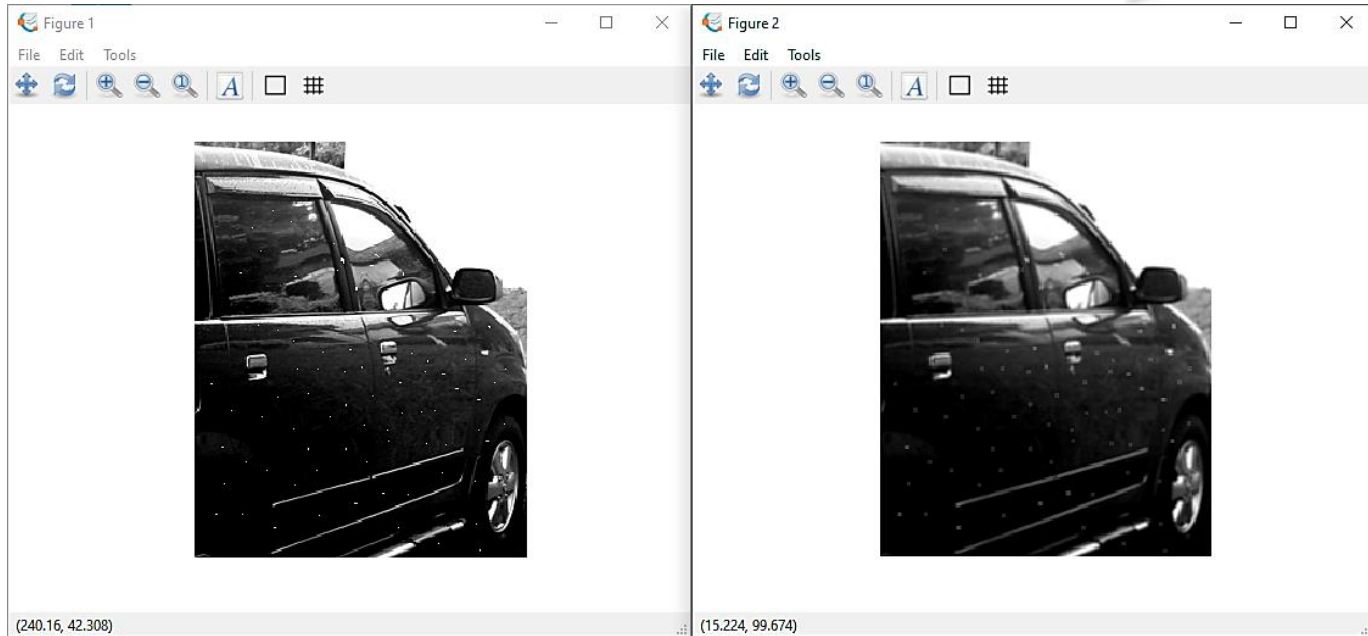
```
>> H=[1 1 1; 1 1 1; 1 1 1]/9;  
>> tapis('mobil.tif',H);
```



Command Window

```
>> H=[1 1 1;1 1 1;1 1 1]/9;  
>> tapis('mobil.tif',H);
```

HASIL FILTER LOLOS RENDAH



(a) citra mobil.tif sebelum di filter

(b) Efek filter lolos rendah pada citra mobil

Gambar 8.25 Efek filter lolos rendah pada citra mobil

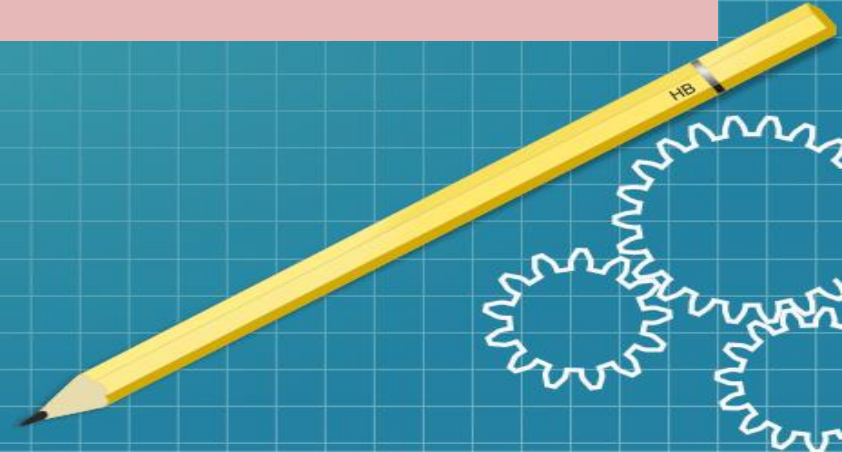


FILTER LOLOS TINGGI

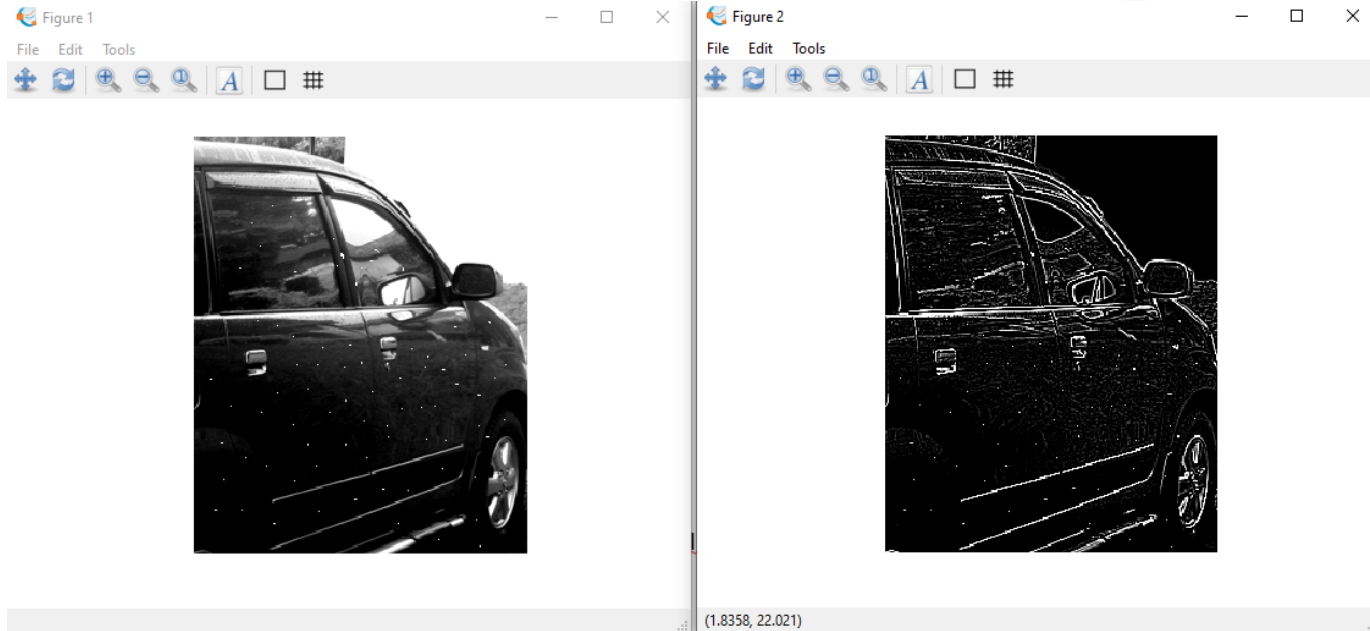


TULISKAN LISTING PROGRAM BERIKUT DI COMMAND WINDOWS

```
>>% Contoh Pengujian Program diatas untuk  
>>%melakukan filter lolos rendah pada citra mobil.tif  
  
>> H=[1 1 1; 1 8 1; 1 1 1];  
>> tapis('mobil.tif',H);
```



HASIL FILTER LOLOS TINGGI



(a) citra mobil.tif sebelum di filter

(b) Efek filter lolos tinggi pada citra mobil dengan nilai pusat 8

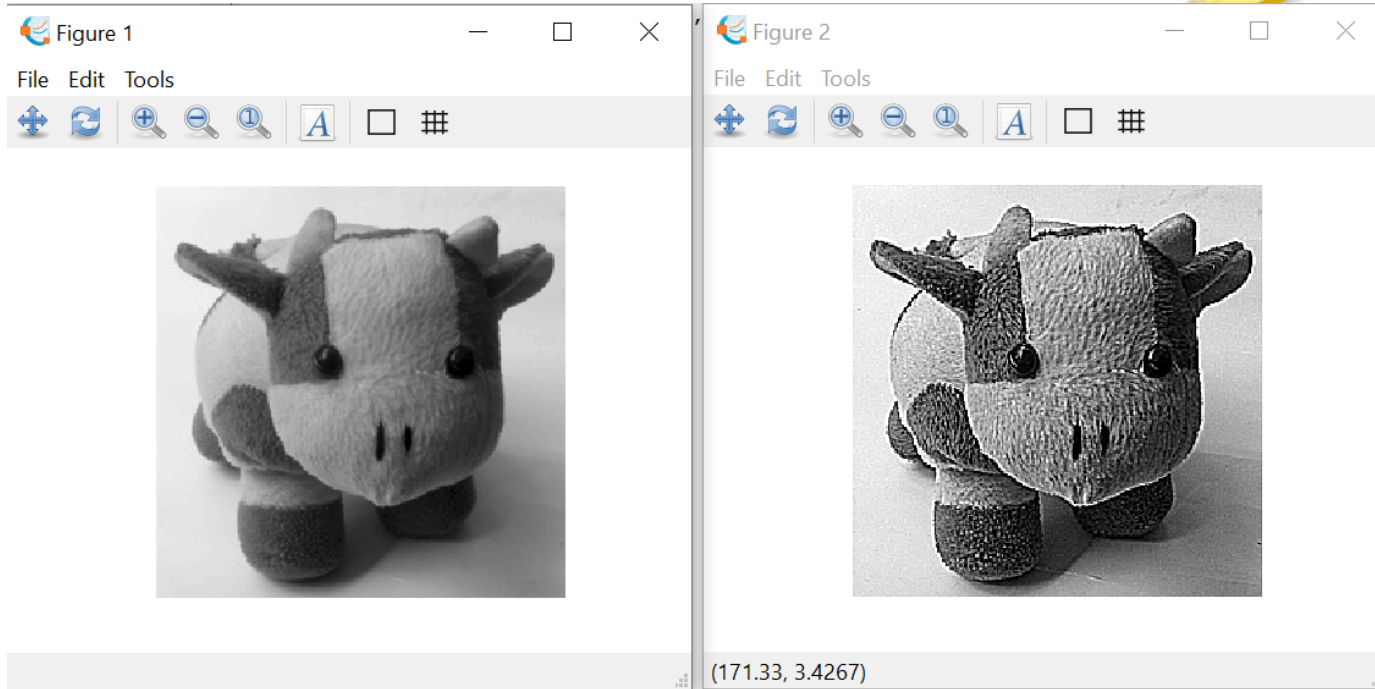
Gambar 8.26 Efek filter lolos tinggi dengan pada citra mobil

FILTER HIGH-BOOST

TULISKAN LISTING PROGRAM BERIKUT DI COMMAND WINDOWS

```
>>% Contoh Pengujian Program diatas untuk  
>>%melakukan filter high boost pada citra boneka.tiff  
  
>> H=[-1 -1 -1; -1 9 -1; -1 -1 -1];  
>> tapis('boneka.tif',H);
```

HASIL FILTER HIGH BOOST



(a) citra mobil.tif sebelum di filter

(b) Efek filter high boost pada citra boneka dengan nilai pusat 9

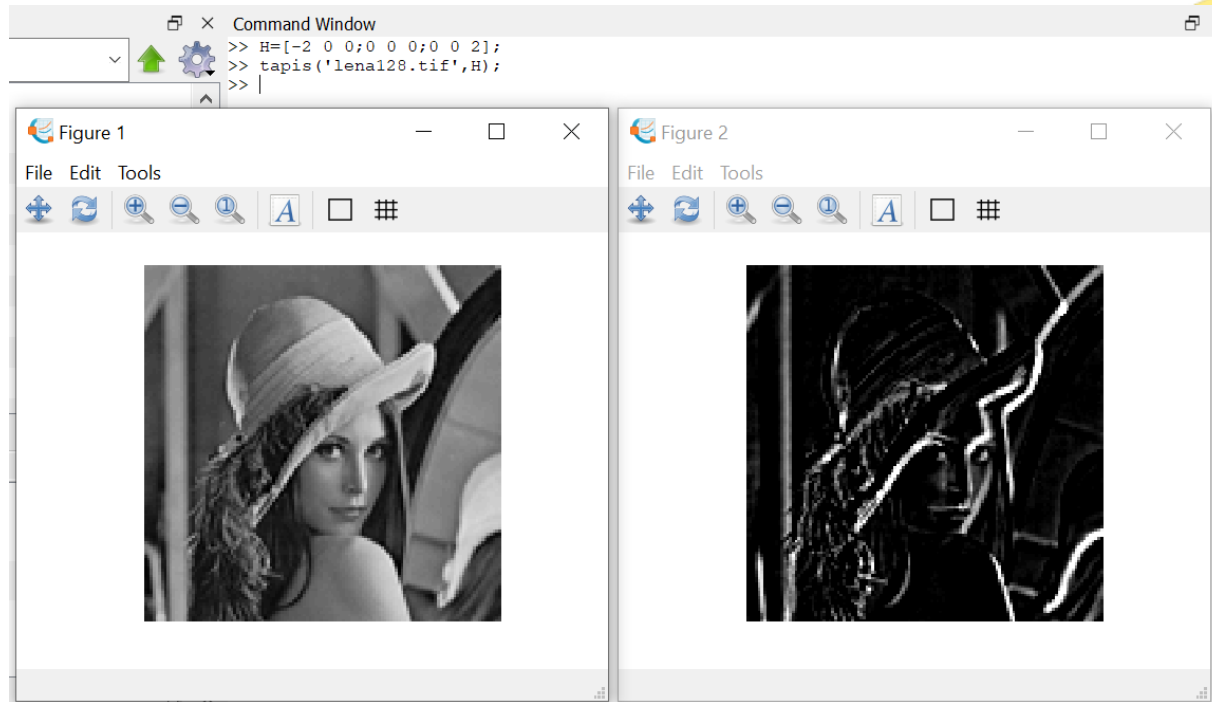
Gambar 8.27 Efek filter high boost pada citra boneka.tif

EMBOSS

TULISKAN LISTING PROGRAM BERIKUT DI COMMAND WINDOWS

```
>>% Contoh Pengujian Program diatas untuk  
>>%melakukan emboss pada citra lena128.tif  
  
>> H=[-2 0 0; 0 0 0; 0 0 -2];  
>> tapis('lena128.tif',H);
```

HASIL FILTER HIGH BOOST



(a) citra mobil.tif sebelum di filter

(b) Efek emboss pada citra lena128.tif

Gambar 8.28 Efek emboss pada lena128



....
TERIMAKASIH