



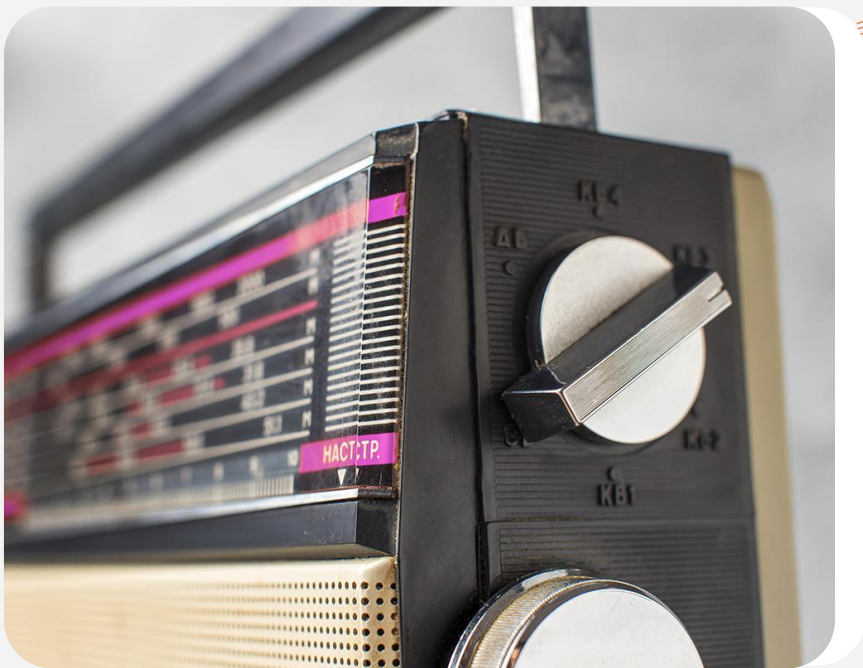
INTENSITY TRANSFORMATION & SPATIAL FILTERING

Group 3

Edgrant H. S. (2206025016)

Giovan C. S. (2206816084)

Nicholas S. (2206059396)



01 Introduction

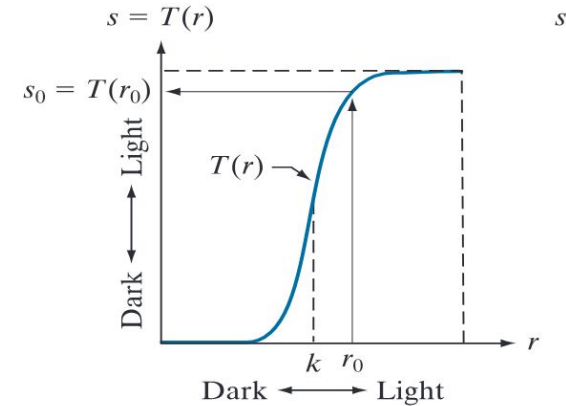
Intensity Transformation

Intensity Transformation berfokus pada perubahan tingkat kecerahan dan kontras suatu gambar dengan mengubah nilai intensitas pikselnya. Teknik ini berguna untuk meningkatkan detail dalam gambar dengan kontras rendah atau menerapkan efek khusus seperti negatif gambar dan penyesuaian gamma.

$$g(x,y) = T(f(x,y)),$$

where $f(x,y)$ is an input image, $g(x,y)$ is the output image, and T is an operator on f defined over a neighborhood (typically a rectangle) of point (x,y) .

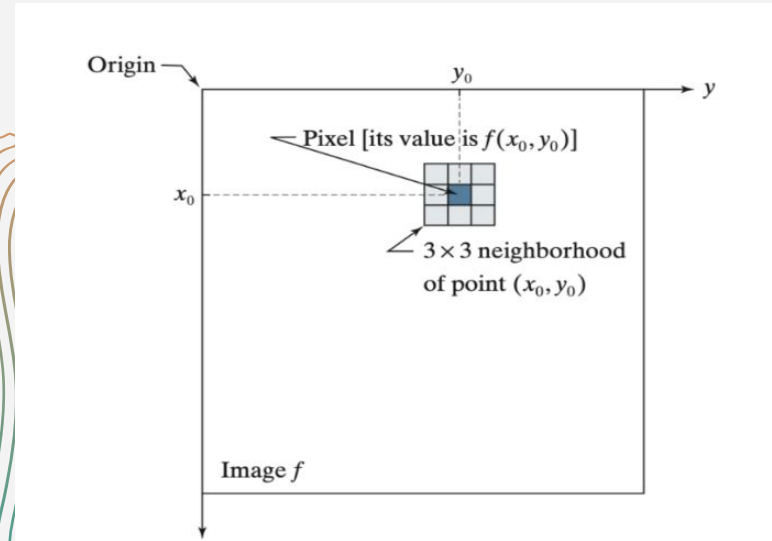
Persamaan transformasi image
 $f(x,y)$ terhadap $g(x,y)$ melalui fungsi
 $T(r)$



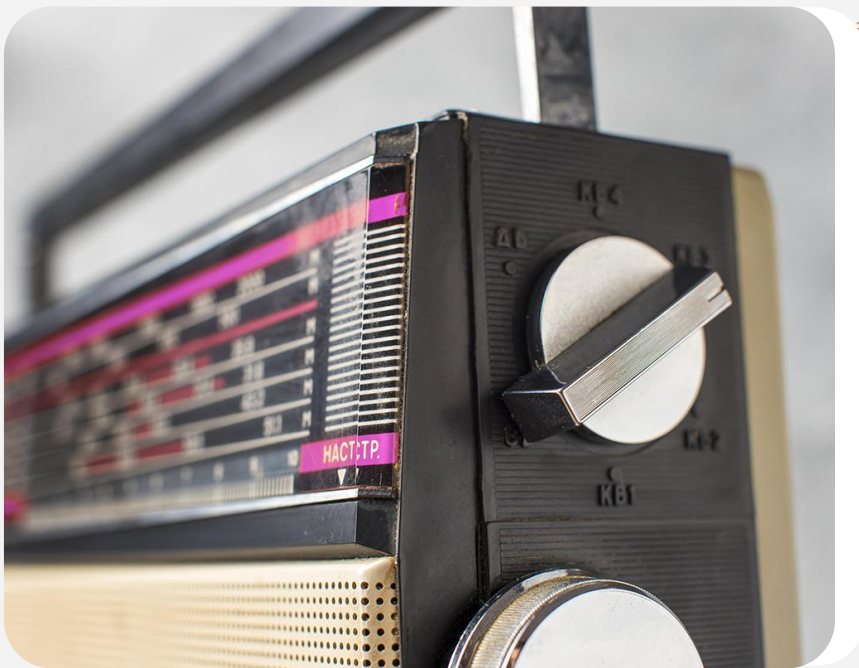
Grafik menaikan Kontras fungsi $T(r)$.
nilai r (keterangan pixel) yang
dibawah *threshold* k akan
diturunkan keterangannya dan juga
sebaliknya

Spatial Filtering

Spatial Filtering bekerja dengan menerapkan operasi matematis pada sekelompok piksel untuk menghasilkan efek tertentu, seperti penghalusan (smoothing) atau penajaman (sharpening).



pixel output dipengaruhi oleh pixel sekitarnya

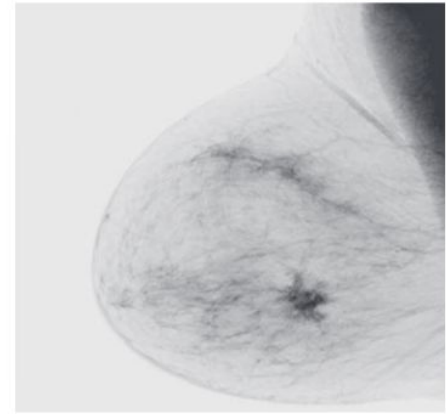
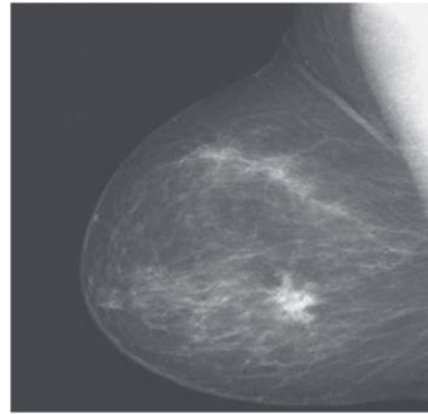
A decorative graphic consisting of several concentric, wavy lines in a light pink or orange color, positioned to the left of the section header.

02 Spatial dan Transform Domain

Spatial Domain

Spatial domain berfokus pada manipulasi langsung nilai intensitas piksel berdasarkan posisi spasialnya dalam gambar. Setiap piksel direpresentasikan oleh koordinatnya serta nilai intensitasnya yang mencerminkan tingkat kecerahan atau warna. Transformasi dalam spatial domain dinyatakan dengan persamaan: **$O(x,y)=T[I(x,y)]$**

Spatial domain banyak digunakan karena memungkinkan pemrosesan gambar yang intuitif dan langsung, misalnya dalam operasi seperti brightness adjustment, histogram equalization, dan edge detection.

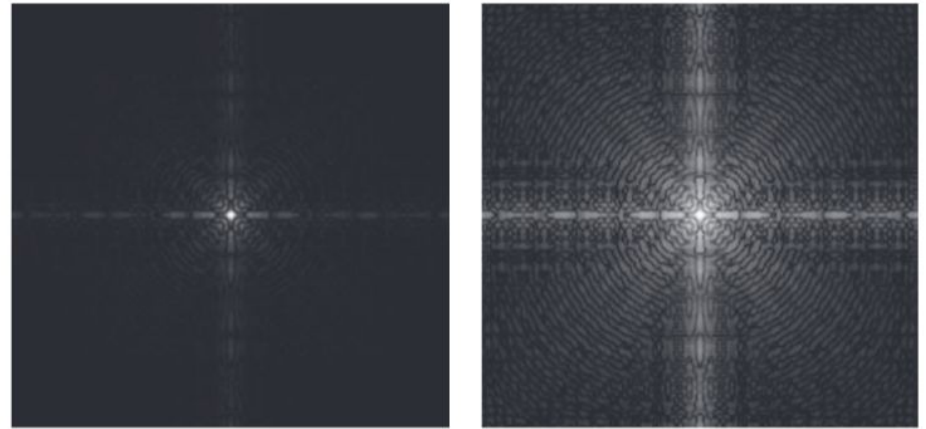


$T[r]$ merupakan fungsi inverse yang menghasilkan gambar negatif

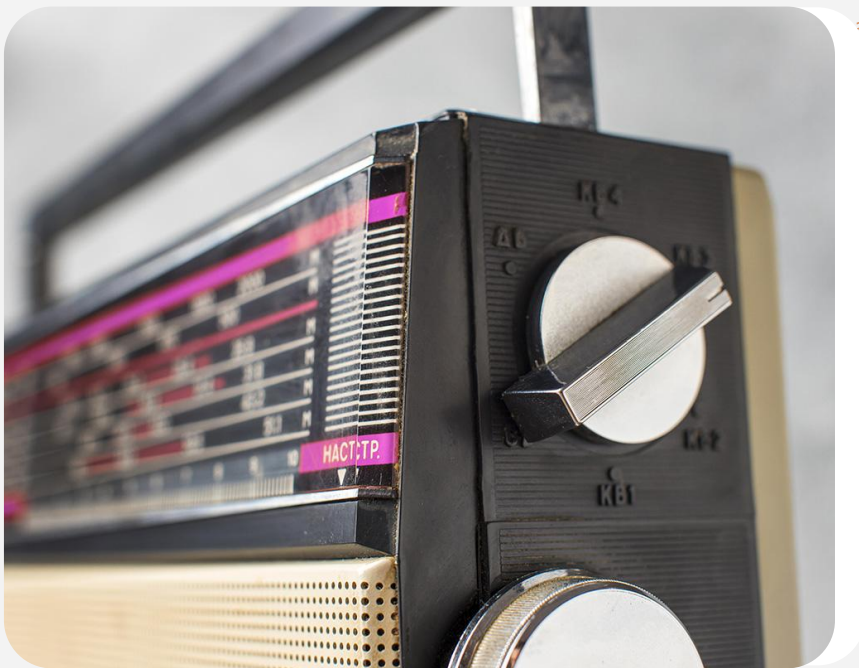
Transform Domain

Pada transform domain, gambar tidak diproses langsung dalam bentuk pikselnya, melainkan dikonversi ke representasi lain, biasanya dalam domain frekuensi. Dalam domain ini, frekuensi rendah merepresentasikan detail global atau informasi struktur utama gambar, sedangkan frekuensi tinggi mengandung detail tepi dan tekstur halus.

Teknik ini berguna dalam pengurangan noise, kompresi citra, dan deteksi fitur, di mana informasi gambar dapat dimanipulasi lebih efisien dibandingkan dengan pendekatan langsung dalam spatial domain.

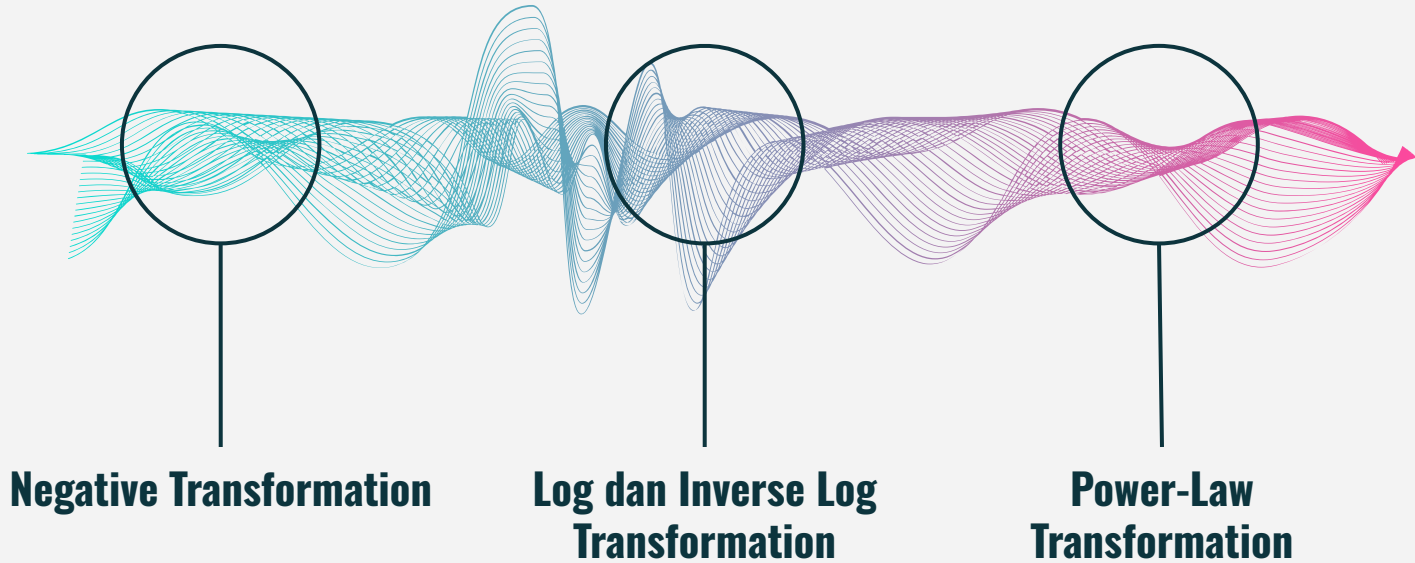


Frekuensi domain dengan Log Transformation

A decorative graphic element consisting of several concentric, wavy lines in a light pink color, positioned to the left of the section header.

03 Intensity Transformation

Metode Transformation



Negative Transformation

Negative Transformation adalah metode dasar dalam transformasi intensitas yang mengubah setiap piksel gambar dengan cara membalikkan tingkat kecerahannya.

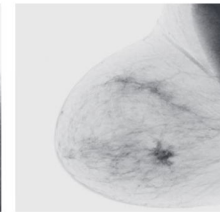
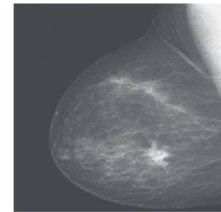
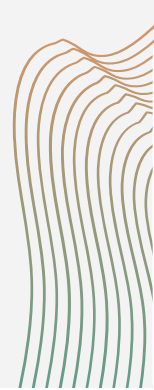
Secara matematis, transformasi ini dirumuskan sebagai: $s = L - 1 - r$

s : nilai intensitas piksel setelah transformasi,

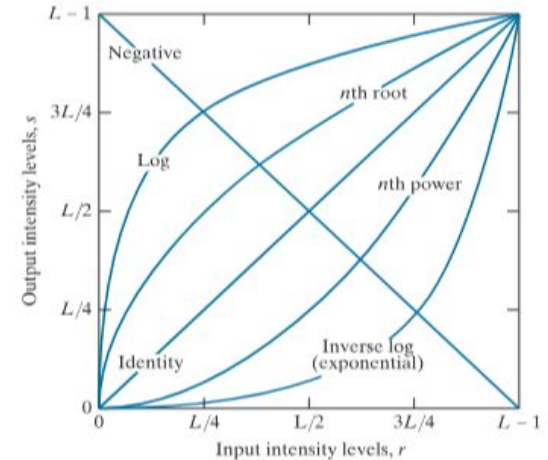
r : nilai intensitas awal piksel,

L : nilai maksimum intensitas (misalnya 255 untuk gambar 8-bit).

Transformasi ini berguna dalam meningkatkan detail gambar, terutama pada gambar dengan kontras rendah. Dalam aplikasi dunia nyata, transformasi negatif sering digunakan dalam pemrosesan gambar untuk meningkatkan visibilitas objek yang sulit terlihat.



Negative transformation

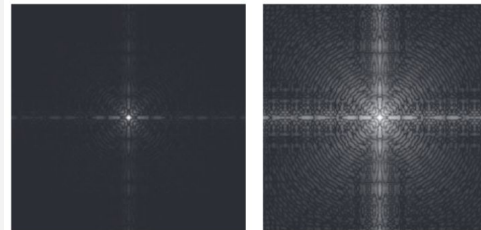
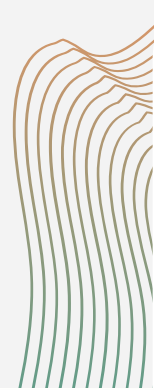


Grafik transformasi

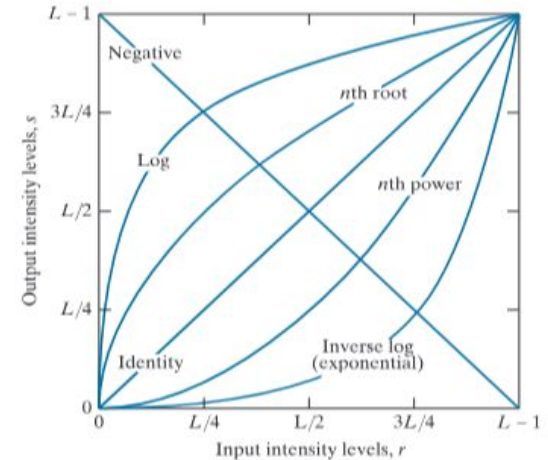
Log dan Inverse Log Transformation

Log dan inverse log transformation adalah teknik yang digunakan untuk memperjelas detail dalam gambar yang memiliki distribusi intensitas yang tidak merata.

- Log
Transformasi ini memperjelas detail pada bagian yang sangat gelap dengan meningkatkan nilai intensitas piksel rendah secara signifikan. Secara matematis :
 $s = c \log(1+r)$
- Inverse Log
Transformasi ini memperkuat detail pada bagian terang gambar dengan mengurangi intensitas piksel tinggi. Secara matematis :
 $s = e^{(r-c)} + 1$



Log transformation



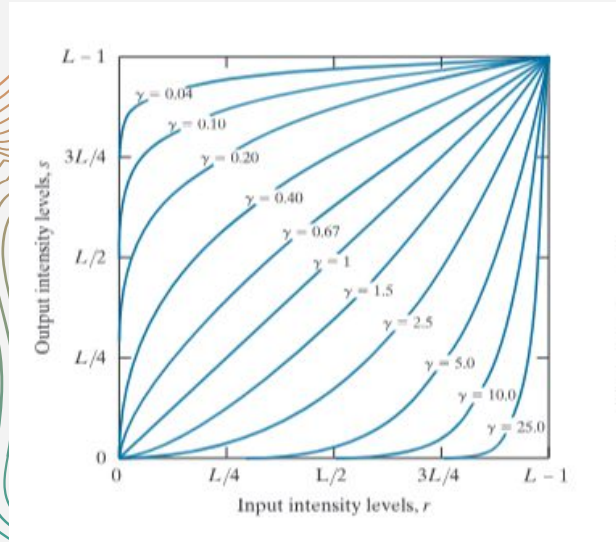
Grafik transformasi

Power-Law Transformation

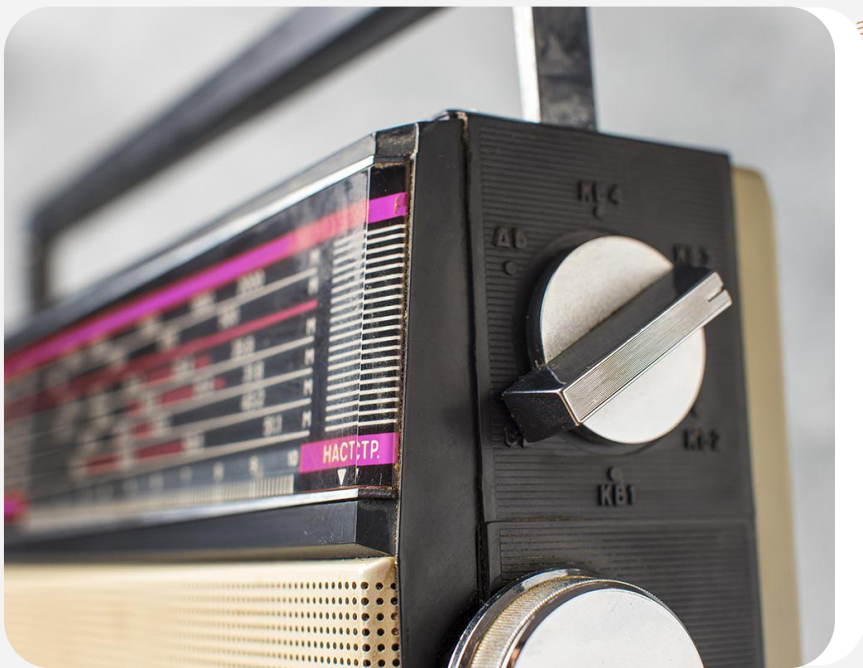
Power-Law Transformation atau Gamma Correction adalah teknik yang digunakan untuk mengontrol kecerahan dan kontras gambar. Secara matematis, transformasi ini dinyatakan sebagai: $s = c \cdot r^\gamma$

Efek dari Gamma Correction tergantung pada nilai gammanya a:

- Jika $\gamma < 1$, gambar akan menjadi lebih terang.
- Jika $\gamma > 1$, gambar akan menjadi lebih gelap.



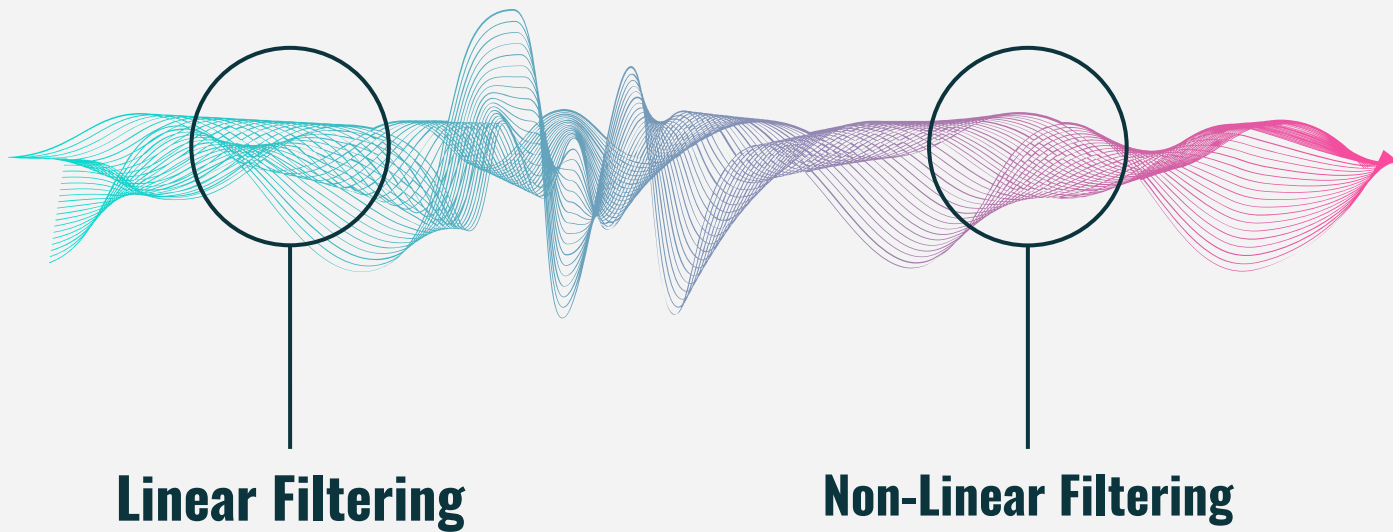
Grafik Power-Law Transformation dengan gamma yang bervariasi



04

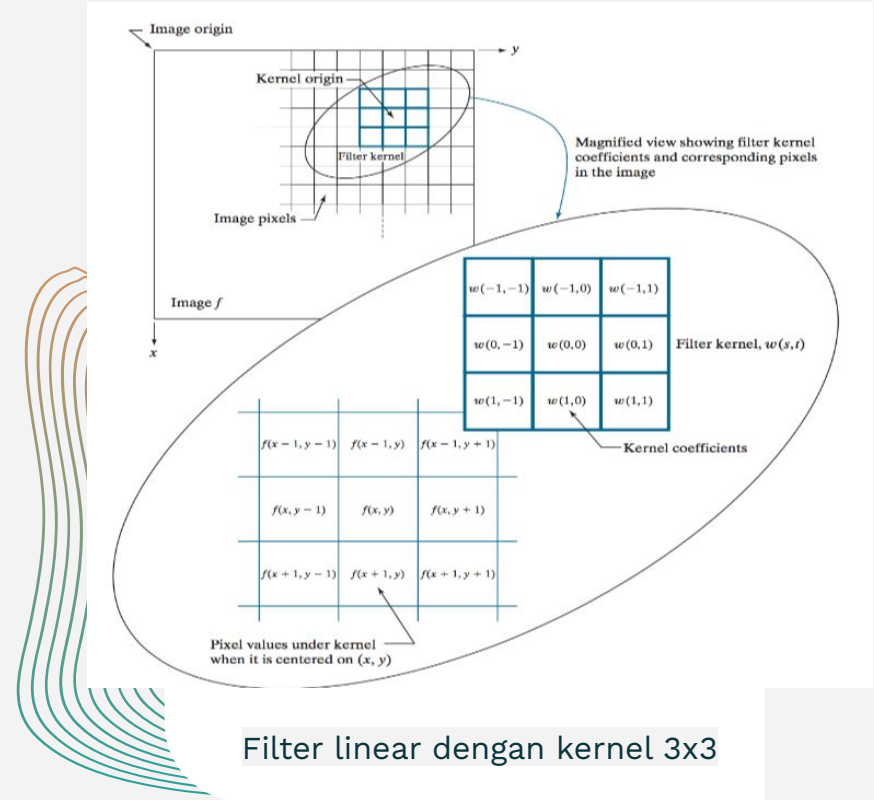
Spatial Filtering

Type Spatial Filtering



Linear Filtering

Linear filtering adalah teknik filtering dalam pemrosesan citra yang menerapkan operasi linier, seperti konvolusi, untuk menghasilkan efek tertentu pada gambar. Dua jenis utama linear filtering adalah lowpass filtering (Smoothing) dan highpass filtering (Sharpening).

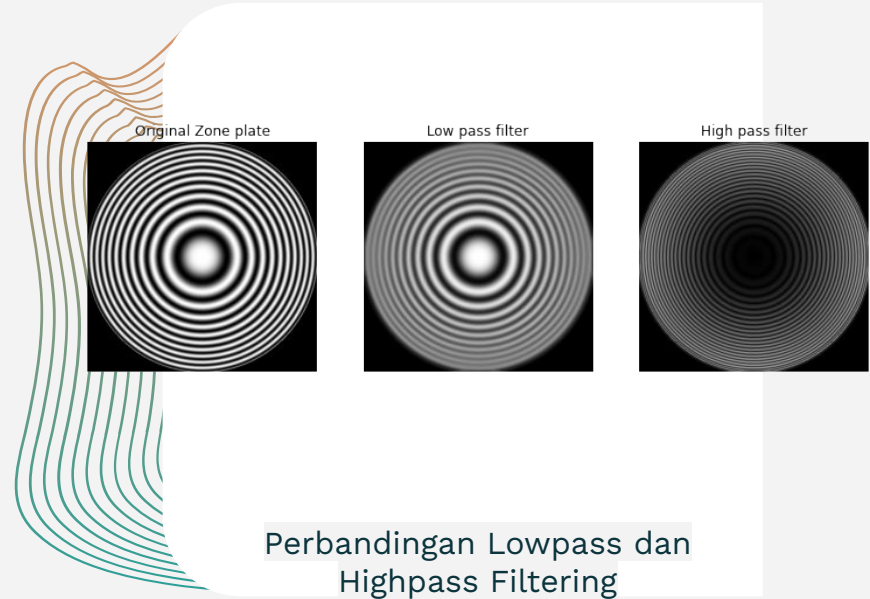


Lowpass Filtering (LPS) – Smoothing

Lowpass filtering (LPS) adalah metode filtering yang digunakan untuk menghaluskan gambar dengan cara mengurangi variasi nilai intensitas antara piksel yang berdekatan. Contoh filter LPS adalah mean filter dan gaussian filter.

- Mean Filter adalah LPS dengan menggunakan rata-rata dari piksel dalam kernel untuk menghasilkan efek penghalusan.
- Gaussian filter adalah LPS dengan menggunakan distribusi Gaussian untuk memberikan bobot lebih pada piksel pusat, sehingga menghasilkan efek smoothing yang lebih alami dibandingkan mean filter.

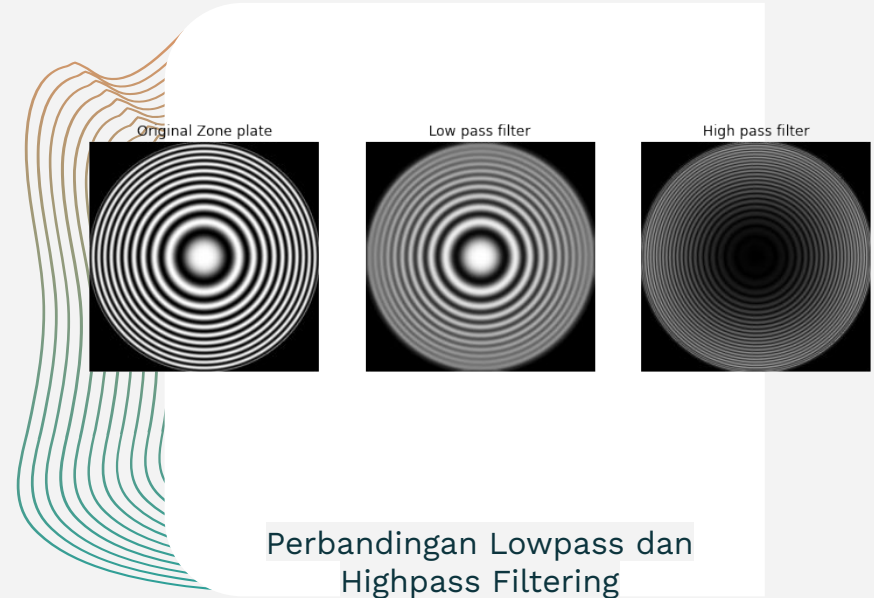
Secara matematis: $g(x, y) = f(x, y) \circ h(x, y)$



Highpass Filtering (HPS) – Sharpening

Highpass Filtering (HPS) bertujuan untuk mempertajam gambar dengan menonjolkan perubahan intensitas yang tajam, seperti tepi objek dalam gambar. HPS dilakukan dengan mendeteksi perbedaan antara piksel pusat dengan piksel sekitarnya. Contoh filter HPS adalah Laplacian Filter dan Sobel & Prewitt Filters.

- Laplacian filter menggunakan operasi diferensiasi kedua untuk mendeteksi tepi dengan melihat perubahan intensitas piksel.
- Sobel & Prewitt filter digunakan dalam deteksi tepi dengan menghitung perbedaan gradien intensitas dalam arah horizontal dan vertikal.



Secara matematis : **HPS = Original image - LPS**



05

Demo

Code Spatial Filtering

```
clc;  
clear;  
x=imread('20250119_134256.JPG');
```

```
% -----filter coeff.-----%  
f1=[1 1 1;1 1 1;1 1 1];  
f2=[1 0 -1;1 0 -1;1 0 -1];  
f3=[1 1 1;0 0 0;-1 -1 -1];
```

```
% -----filtering-----%  
x1=x(1:256,1:256);  
y1=conv2(x1,f1);  
y2=conv2(x1,f2);  
y3=conv2(x1,f3);  
subplot(2,2,1);imshow(x);title('IMAGE ')  
subplot(2,2,2);imshow(uint8((y1)));title('f1- IMAGE ')  
subplot(2,2,3);imshow(uint8((y2)));title('f2- IMAGE ')  
subplot(2,2,4);imshow(uint8((y3)));title('f3-IMAGE ')
```

Hasil Spatial Filtering

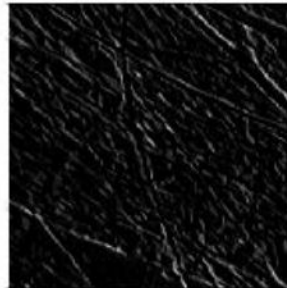
IMAGE



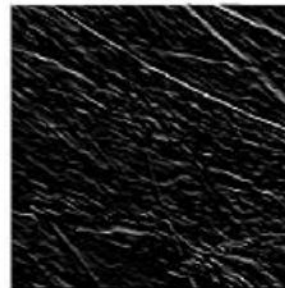
f1- IMAGE



f2- IMAGE



f3-IMAGE



Code Intensity Transformation

```
% MATLAB program to demonstrate image negative transformation
% Insert a color input image using imread function
input_image = imread('20250119_134256.JPG');
```

```
% Find the negative of the input image
negative_image = 255 - input_image;
```

```
% Display the original image and negative image
subplot(1, 2, 1), imshow(input_image), title('Original Image')
subplot(1, 2, 2), imshow(negative_image), title('Negative Image')
```

```
% Save the negative image in jpg format
imwrite(negative_image, 'negative_image.jpg');
```

```
% Log Transformation
% Convert the image to double datatype for calculations
input_image = im2double(input_image);
```

Code Intensity Transformation

```
% Constant to determine the nature of the log curve  
c = 1;
```

```
% Perform the log transformation  
log_transformed = c * log(1 + input_image);
```

```
% Display the original image and log-transformed image  
subplot(1, 2, 1), imshow(input_image), title('Original Image')  
subplot(1, 2, 2), imshow(log_transformed), title('Log-Transformed Image')
```

```
% Save the log-transformed image  
imwrite(log_transformed, 'log_transformed_image.jpg');
```

```
% Power Law Transformation  
% Set a desired gamma value for the power law transformation  
gamma = 0.4;
```

Code Intensity Transformation

```
% Perform the power law transformation of image  
power_law_image = input_image .^ gamma;
```

```
% Display the original and power law transformed images  
subplot(1, 2, 1), imshow(input_image), title('Original Image')  
subplot(1, 2, 2), imshow(power_law_image), title('Power Law Transformed Image')
```


Hasil Intensity Transformation

Original Image



Power Law Transformed Image

