

Perbaikan Citra Luar Ruangan pada Cuaca Hujan dengan *Fast Fourier Transform*

Kevin John Wesley Hutabarat - 13521042
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13521042@std.stei.itb.ac.id

Abstract— Perkembangan teknologi saat ini memberikan banyak dampak kepada kehidupan manusia. Salah satu teknologi yang sedang berkembang dengan pesat adalah kendaraan yang dapat bergerak tanpa kemudi dari manusia, atau yang disebut juga kendaraan otonom. Salah satu aspek yang penting dari kendaraan otonom adalah sensor. Salah satu sensor yang terdapat dalam kendaraan dalam kendaraan otonom adalah kamera. Cuaca yang dihadapi oleh kendaraan otonom tidak selalu baik. Terdapat kendala saat cuaca hujan, di mana penglihatan dari kamera menjadi terbatas. Eksperimen ini mencoba untuk menyelesaikan masalah ini dengan beberapa metode dalam pemrosesan citra konvensional.

Keywords—*image deraining, pemrosesan citra, fast fourier transform, reduksi derau*

I. PENDAHULUAN

Perkembangan teknologi saat ini membuat banyaknya kemunculan kendaraan-kendaraan yang bekerja secara otomatis. Kendaraan ini dapat beroperasi secara otomatis tanpa kendali dari manusia secara manual. Saat ini, banyak perusahaan berlomba-lomba untuk menciptakan kendaraan otonom yang paling aman dan mampu memberikan pelayanan yang paling baik.

Untuk melakukan fungsinya, kendaraan otonom dilengkapi dengan banyak komponen, seperti sensor dan aktuator. Aktuator adalah komponen yang memungkinkan kendaraan otonom untuk bergerak dan melakukan aksinya. Sensor digunakan untuk menangkap hal-hal di sekitar kendaraan. Sensor memungkinkan kendaraan untuk menentukan aksi yang akan dilakukan, seperti berjalan, berhenti, ataupun menaikkan kecepatan.

Dalam penerapannya, kamera dapat bekerja dengan baik untuk “melihat” lingkungan sekitar dalam kondisi yang ideal (cuaca yang cerah dan tidak hujan). Namun dalam praktiknya, ada masa di mana kendaraan otonom ini harus bekerja dalam cuaca yang kurang baik seperti hujan. Tetesan air hujan akan menghambat penglihatan dari kamera dan menghambat kinerja dari kendaraan otonom. Oleh karena itu, dibutuhkan pemrosesan terhadap gambar yang ditangkap pada saat cuaca hujan.

Sudah banyak teknologi yang dapat melakukan penghapusan derau hujan yang terdapat pada citra. Rintik hujan

dapat dianggap sebagai derau yang muncul secara berkala. Oleh karena itu, salah satu pendekatan sederhana yang dapat dilakukan adalah dengan memprosesnya dalam ranah frekuensi. Penelitian ini akan menunjukkan bagaimana transformasi citra ke dalam domain frekuensi dapat diolah untuk menghilangkan derau hujan yang terkandung di dalamnya.

II. LANDASAN TEORI

A. *Image Enhancement*

Image enhancement adalah suatu upaya dalam pengolahan citra untuk memperbaiki kualitas citra agar tampak lebih baik. Tujuan dari dilakukannya proses ini adalah memperoleh citra yang lebih sesuai untuk digunakan pada aplikasi lebih lanjut, misalkan untuk mengenali objek di dalam citra. Proses ini penting karena menjadi satu proses awal dalam visi komputer.

Terdapat beberapa alasan diperlukannya *image enhancement*, antara lain

- Citra kerap kali mengandung derau
- Citra terlihat terlalu gelap/terang, kurang tajam, atau kabur
- Terdapat kecacatan saat akuisisi citra, yang disebabkan oleh
 - o Lensa
 - o Objek bergerak
- Distorsi geometrik yang disebabkan oleh lensa atau sudut pengambilan gambar
- Dll.

B. *Image Enhancement dalam Ranah Frekuensi*

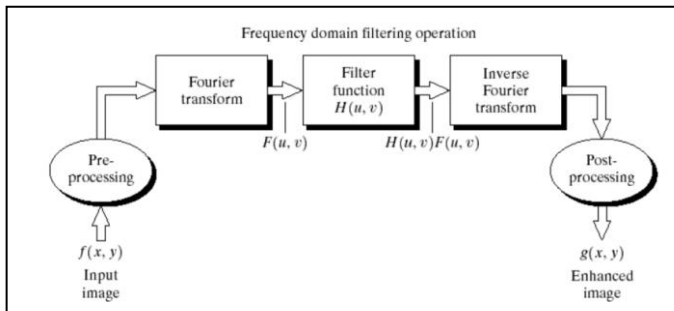
Operasi perbaikan citra yang umum dan mudah dilakukan adalah dalam ranah spasial. Operasi pada ranah spasial berarti operasi dilakukan langsung pada pixel-pixel yang terkandung di dalam citra. Namun, ada beberapa kasus untuk citra di mana operasi pada ranah spasial kurang mampu memberikan hasil yang baik dalam perbaikan citra. Oleh karena itu, terdapat metode yang dapat mengubah citra ke dalam ranah frekuensi.

Hubungan antara operasi penapisan dalam ranah spasial dan ranah frekuensi ditunjukkan dengan persamaan sebagai berikut

$$g(x,y) = h(x,y)*f(x,y) \leftrightarrow G(u,v) = H(u,v)F(u,v)$$

$$g(x,y) = h(x,y)f(x,y) \leftrightarrow G(u,v) = H(u,v) * F(u,v)$$

dengan $h(x,y)$, $f(x,y)$, dan $g(x,y)$ secara berturut-turut adalah penapis, masukan, dan hasil citra dalam ranah spasial, dan



$H(u,v)$, $F(u,v)$, dan $G(u,v)$ secara berturut-turut adalah penapis, masukan, dan hasil citra yang ditransformasi dalam ranah frekuensi.

Gambar II.1 Proses image enhancement dalam ranah frekuensi

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/12-Penapisan-Citra-dalam-Ranah-Frekuensi-2024.pdf>

Langkah-langkah pemrosesan citra dalam ranah frekuensi ditampilkan pada Gambar II.1. Citra masukan yang akan diproses akan dilakukan prapemrosesan terlebih dahulu dengan metode-metode lain, seperti pencerahan, perataan histogram, dan lain-lain. Citra yang telah dilakukan prapemrosesan akan ditransformasi ke dalam ranah frekuensi dengan transformasi fourier. Proses ini akan menghasilkan citra dalam bentuk frekuensi. Frekuensi ini kemudian dimasukkan ke dalam fungsi penapis tertentu sesuai dengan kebutuhan. Hasil dari fungsi ini adalah frekuensi-frekuensi yang lolos dari penapisan yang belum dapat dilihat hasilnya karena masih berada di ranah frekuensi. Untuk dapat dilihat oleh manusia, hasil dari penapisan ini harus dikembalikan ke dalam ranah spasial.

C. Low-pass filter

Low-pass filter atau penapis lolos-rendah adalah penapis yang digunakan untuk pelembutan citra. Penapis ini akan menghasilkan efek blur dan untuk mengurangi derau pada citra. Derau pada citra umumnya berupa variasi intensitas suatu pixel yang tidak berkorelasi dengan *pixel-pixel* tetangganya. *Pixel-pixel* derau umumnya memiliki frekuensi tinggi. Dengan penapis lolos-rendah, pixel-pixel dengan frekuensi rendah akan diloloskan, dan yang berfrekuensi tinggi akan ditekan.

Terdapat berbagai metode untuk melakukan penapisan lolos-rendah, seperti penapis rerata (mean filter/averaging filter), penapis median, minimum, dan maksimum. Penapis lolos-rendah rerata memiliki aturan sebagai berikut:

- Semua bobot di dalam penapis harus positif
- Jumlah semua bobot sama dengan 1
- Jika jumlah semua bobot lebih besar dari 1, maka konvolusi menghasilkan penguatan pada derau
- Jika jumlah semua bobot kurang dari 1, maka yang dihasilkan adalah penurunan, dan nilai setiap pixel di

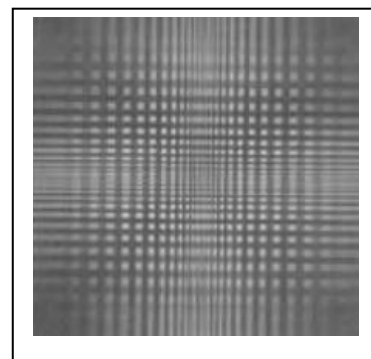
seluruh bagian citra berkurang. Akibatnya, citra hasil pelembutan tampak lebih gelap.

Salah satu contoh penapis lolos-rendah adalah *gaussian filter*. Gaussian filter adalah suatu penapis yang memberikan efek blur secara halus dan perlahan saat menuju ke tepi penapis.

D. Transformasi Fourier

Transformasi Fourier adalah metode untuk mengubah fungsi dari ranah waktu atau spasial ke dalam ranah frekuensi. Contoh dari ranah waktu adalah gelombang bunyi (satu dimensi), dan contoh dari ranah spasial adalah citra (dua dimensi). Inti dari Transformasi Fourier adalah menguraikan sinyal atau gelombang menjadi sejumlah sinusoida dari berbagai frekuensi, yang jumlahnya ekuivalen dengan gelombang asal.

Di dalam pemrosesan citra, umumnya citra disajikan dalam ranah spasial $f(x,y)$. Meskipun begitu, citra juga dapat disajikan dalam ranah frekuensi $F(u,v)$, yaitu ruang di mana setiap nilai pada posisi citra F menyatakan besaran bahwa nilai intensitas berubah pada jarak tertentu. Contohnya adalah jika frekuensi pada nilai *pixel* 20 adalah 0.1, artinya 1 periode setiap 10 pixel. Intensitas *pixel* berubah dari gelap ke terang dan kembali ke gelap setelah 10 *pixel*.



Gambar II.2 Contoh citra dalam ranah frekuensi

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/11-Transformasi-Citra-2024.pdf>

Gambar II.2 menunjukkan citra dalam ranah frekuensi. Frekuensi tinggi berada pada sekitar pusat, sedangkan frekuensi rendah berada di sekitar titik sudut. Frekuensi rendah bersesuaian dengan intensitas *pixel* yang bervariasi secara perlahan (contohnya adalah permukaan kontinu). Sebaliknya, frekuensi tinggi bersesuaian dengan intensitas pixel yang bervariasi dengan cepat (contohnya adalah *pixel* tepi).

E. Bilateral Filtering

Bilateral filter adalah suatu metode yang digunakan untuk melembutkan citra dan mereduksi derau, tetapi tetap mempertahankan tepi pada citra. Bilateral filtering dapat diformulasikan dengan rumus pada Gambar II.3 berikut.

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$

Gambar II.3 Rumus *bilateral filtering*

Rumus ini memuat *normalization factor*, *space weight* dan *range weight*, yang mampu mereduksi derau dengan mempertahankan tepi.

III. IMPLEMENTASI

Rintik hujan dapat dipandang sebagai unsur citra yang memiliki frekuensi tinggi. Pada derau hujan, terdapat perubahan pixel secara mendadak dari citra asli ke citra hujan. Hal ini akan terdeteksi sebagai bagian yang memiliki frekuensi yang tinggi.

Implementasi dari sistem ini dilakukan dengan bahasa pemrograman MATLAB karena MATLAB banyak digunakan untuk program-program pemrosesan citra.

Program dibagi menjadi tiga bagian, yaitu

1. Gaussian blur
2. Bilateral filtering
3. Penapisan dengan penapis lolos rendah pada ranah frekuensi

Berikut adalah hasil implementasi lengkap kode untuk program penghilangan derau hujan pada citra

```
function derain_image(image_path)
% Membaca citra
img = imread(image_path);

% Inisialisasi citra hasil
img_derained = zeros(size(img), 'like', img);

% Visualisasi FFT
figure;
colormap gray;

% Pemrosesan untuk tiap kanal
for channel = 1:3
% Ekstraksi setiap kanal warna
img_channel = img(:, :, channel);

% Gaussian blur
h = fspecial('gaussian', [7 7], 2);
img_gaussian = imfilter(img_channel, h, 'replicate');

% Bilateral Filtering (Edge-preserving smoothing)
img_bilateral = imbilatfilt(img_gaussian, 75, 15);

% FFT untuk menghapus komponen dengan frekuensi
tinggi
img_fft = double(img_bilateral);
fft_img = fft2(img_fft); % Perform FFT
```

```
fft_shift = fftshift(fft_img); % Shift zero frequency
to the center

fft_magnitude_before = log(abs(fft_shift) + 1); % Log
scale for better visualization

% Low-Pass Filter Mask
[rows, cols] = size(img_fft);
crow = round(rows / 2); % Center row
ccol = round(cols / 2); % Center column
radius = 200; % Radius of low-pass filter
mask = zeros(rows, cols);
[x, y] = meshgrid(1:cols, 1:rows);
mask(((x - ccol).^2 + (y - crow).^2) <= radius^2) = 1;

fft_filtered = fft_shift .* mask;

fft_magnitude_after = log(abs(fft_filtered) + 1); %
Log scale for better visualization

% FFT balikan
fft_ishift = ifftshift(fft_filtered);
img_reconstructed = abs(ifft2(fft_ishift));

% Menyimpan hasil ke setiap kanal pada citra
img_derained(:, :, channel) =
uint8(img_reconstructed);

subplot(3, 6, (channel - 1) * 6 + 1);
imshow(img_channel); title(['Channel ', num2str(channel), '
Original']);

subplot(3, 6, (channel - 1) * 6 + 2);
imshow(img_gaussian); title(['Gaussian ', num2str(channel)]);

subplot(3, 6, (channel - 1) * 6 + 3);
imshow(img_bilateral); title(['Bilateral ', num2str(channel)]);

subplot(3, 6, (channel - 1) * 6 + 4);
imshow(fft_magnitude_before, []); title(['FFT Before (Ch ',
num2str(channel), ')']);

subplot(3, 6, (channel - 1) * 6 + 5);
imshow(fft_magnitude_after, []); title(['FFT After (Ch ',
num2str(channel), ')']);

subplot(3, 6, (channel - 1) * 6 + 6);
imshow(uint8(img_reconstructed)); title(['Channel ',
num2str(channel), ' Processed']); end

% Display the original and derained images
figure;
subplot(1, 2, 1); imshow(img); title('Original Image');
subplot(1, 2, 2); imshow(img_derained); title('Derained
Image');
end
```

Fungsi *derain_image* adalah fungsi yang dapat menghilangkan derau hujan pada citra masukan dengan pendekatan penapisan lolos-rendah pada ranah frekuensi. Fungsi ini menerima parameter berupa *image_path*, yang

digunakan untuk membaca *file* gambar masukan pengguna. Mekanisme dari fungsi tersebut adalah sebagai berikut

1. Membagi per kanal

Citra masukan akan diproses untuk setiap kanalnya. Jika citra merupakan citra *grayscale*, kanal yang akan diproses hanyalah satu. Namun, jika citra merupakan citra RGB atau berwarna, akan ada tiga kanal yang akan diproses. Oleh karena itu, diperlukan adanya *traversal* untuk setiap kanal.

```
for channel = 1:3
    % Extract the channel
    img_channel = img(:, :, channel);
```

Gambar III.1 Membagi citra untuk setiap kanal warna

Sumber: Dokumentasi pribadi

2. Gaussian Blurring

Citra yang sudah dibagi per kanal akan diproses dengan gaussian blurring. *Gaussian blurring* ini dilakukan untuk menghilangkan derau pada citra. Proses ini dilakukan dengan membuat matriks *gaussian* berukuran 7x7 dengan standar deviasi 2. Pada citra kemudian akan dilakukan konvolusi dengan matriks yang sudah dibangkitkan tersebut.

```
% Gaussian blur
h = fspecial('gaussian', [7 7], 2);
img_gaussian = imfilter(img_channel, h, 'replicate');
```

Gambar III.2 Proses gaussian blur pada citra

Sumber: Dokumentasi pribadi

3. Bilateral Filtering

Setelah dilakukan *blurring*, akan dilakukan bilateral filtering pada citra. Proses ini dilakukan untuk menghilangkan derau lebih banyak lagi sembari mempertahankan tepi-tepi yang ada pada citra. Fungsi yang digunakan adalah *imblatfilt*, yang sudah tersedia dari MATLAB.

```
% Bilateral Filtering (Edge-preserving smoothing)
img_bilateral = imblatfilt(img_gaussian, 75, 15);
```

Gambar III.3 Proses bilateral filtering pada citra

Sumber: Dokumentasi pribadi

4. Low-pass filtering dalam ranah frekuensi

Setelah dilakukan prapemrosesan dengan *gaussian blurring* dan *bilateral filtering*, pada citra akan dilakukan penghilangan derau dalam ranah frekuensi. Proses ini dilakukan dengan mentransformasi citra dengan Fast Fourier Transform (FFT) menggunakan fungsi *fft2* dari MATLAB. Komponen dengan frekuensi rendah akan digeser ke tengah dan komponen dengan frekuensi tinggi akan digeser ke pinggir dengan fungsi *fftshift*.

```
% FFT for High-Frequency Noise Removal
img_fft = double(img_bilateral);
fft_img = fft2(img_fft); % Perform FFT
fft_shift = fftshift(fft_img); % Shift zero frequency to the center

% Store FFT (magnitude) for visualization
fft_magnitude_before = log(abs(fft_shift) + 1); % Log scale for better visualization

% Create a Low-Pass Filter Mask
[rows, cols] = size(img_fft);
crow = round(rows / 2); % Center row
ccol = round(cols / 2); % Center column
radius = 200; % Radius of low-pass filter
mask = zeros(rows, cols);
[x, y] = meshgrid(1:cols, 1:rows);
mask(((x - ccol).^2 + (y - crow).^2) <= radius^2) = 1;

% Apply the mask
fft_filtered = fft_shift .* mask;

% Store FFT (magnitude) after filtering for visualization
fft_magnitude_after = log(abs(fft_filtered) + 1); % Log scale for better visualization

% Inverse FFT to reconstruct the channel
fft_ishift = ifftshift(fft_filtered);
img_reconstructed = abs(ifft2(fft_ishift));
```

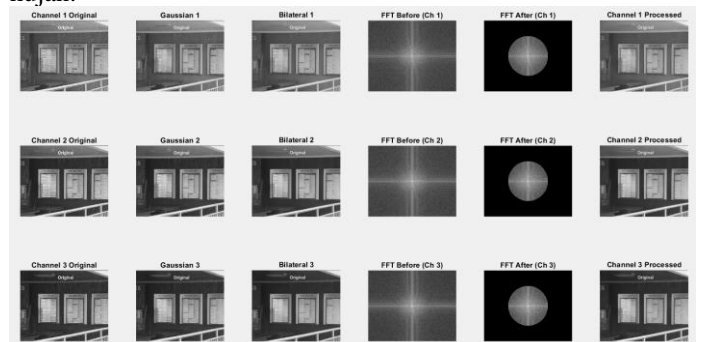
Gambar III.3 Proses penapisan pada ranah frekuensi

Sumber: Dokumentasi pribadi

Setelah memperoleh frekuensi yang telah digeser, akan dibuat mask untuk *low-pass filter*. Mask ini digunakan untuk menapis komponen-komponen dengan frekuensi tertentu. Mask yang digunakan pada program ini adalah mask berbentuk lingkaran, yang akan meloloskan komponen-komponen dengan frekuensi yang ada dalam lingkaran, dan menekan komponen-komponen dengan frekuensi yang ada di luar lingkaran. Mask ini dibuat dengan menetapkan suatu radius, kemudian di-generate dengan *meshgrid* dari MATLAB. Komponen mask di dalam lingkaran diberikan nilai 1, sedangkan di luar lingkaran diberikan nilai 0. Citra dalam ranah frekuensi kemudian dikalikan dengan mask tersebut dengan melakukan operasi pada komponen dengan letak yang bersesuaian. Setelah dilakukan operasi, citra dalam ranah frekuensi tersebut akan dikembalikan ke posisi semula dengan fungsi *fft_ishift* dari MATLAB. Untuk mengembalikannya ke dalam ranah spasial, pada citra akan dilakukan proses pengembalian dengan fungsi *ifft2*.

IV. HASIL EKSPERIMEN DAN ANALISIS

Program yang telah diimplementasi kemudian diuji dengan beberapa citra uji. Citra uji yang digunakan adalah citra yang mengandung rintik hujan, mulai dari hujan ringan, sedang, maupun berat. Berikut adalah hasil dari pemrosesan citra hujan.



Gambar IV.1 Proses penghilangan derau hujan pada setiap kanal warna

Sumber: Dokumentasi pribadi

Gambar IV.1 menunjukkan tahapan penghilangan derau hujan untuk setiap kanal warna yang ada pada citra. Pada gambar Gaussian, ditampilkan hasil operasi bluring pada citra input. Operasi ini menghasilkan citra yang kabur untuk menghilangkan derau. Penghilangan derau lebih lanjut dihasilkan oleh operasi *bilateral filtering*. Dengan operasi ini, tepi dari objek pada citra tetap dapat dipertahankan, meskipun sedang menghilangkan komponen dengan frekuensi tinggi. Pada ranah frekuensi, citra dipotong menjadi radius tertentu untuk mengambil frekuensi yang rendah saja. Hasil pemrosesan untuk setiap kanal pada akhirnya akan digabungkan.



Gambar IV.2 Citra dengan hujan ringan

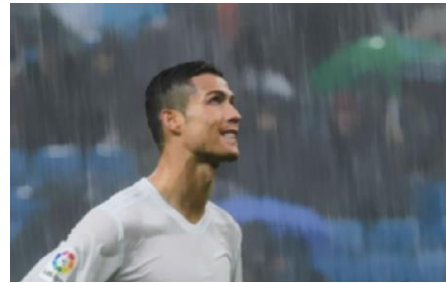


Gambar IV.3 Hasil penghilangan rintik hujan pada citra hujan ringan
Sumber: Dokumentasi pribadi

Hasil menunjukkan bahwa penghilangan rintik hujan pada curah hujan ringan sudah menghasilkan citra yang cukup baik. Rintik hujan yang memiliki ketebalan tipis berhasil dihilangkan dengan operasi yang dilakukan dengan program. Namun, terdapat beberapa rintik hujan yang terlalu tebal yang tidak dapat dihilangkan. Hasil penghilangan hujan juga tidak terlalu bagus karena objek yang terdapat pada citra mengalami pengaburan.



Gambar IV.4 Citra asli dengan hujan medium



Gambar IV.5 Citra hasil penghilangan rintik hujan
Sumber: Dokumentasi pribadi

Pemrosesan pada citra hujan dengan curah hujan medium memberikan hasil yang cukup baik. Rintik hujan pada citra sudah tidak terlihat. Pemrosesan memberikan citra yang sedikit buram namun masih bisa dilihat dengan jelas.



Gambar IV.6 Citra asli hujan lebat



Gambar IV.7 Citra hasil penghilangan rintik hujan
Sumber: Dokumentasi pribadi

Pemrosesan pada citra hujan lebat menunjukkan hasil yang tidak begitu baik. Citra hasil terlihat seperti hanya mengalami pengaburan. Hal ini terjadi karena FFT hanya mampu mendeteksi perubahan frekuensi mendadak yang tinggi, sehingga jika goresan hujan terlalu tebal, program tidak terlalu mampu menghilangkan goresan hujan tersebut.

V. KESIMPULAN

Berdasarkan eksperimen yang dilakukan, diperoleh bahwa citra dengan derau hujan dapat dihilangkan dengan beberapa metode dalam pemrosesan citra. Metode yang dapat digunakan adalah gaussian blur, bilateral filtering, dan low-pass filter dalam ranah frekuensi. Ketika tiga metode tersebut digabungkan, program mampu menghilangkan efek rintik hujan yang ada dalam citra. Eksperimen menunjukkan program mampu menghilangkan derau hujan yang tipis, namun kurang mampu menghilangkan derau hujan yang tebal. Kelemahan lain dari metode ini adalah membuat citra menjadi lebih blur dan objek menjadi tidak jelas. Hal ini terjadi karena metode yang dipakai adalah metode-metode konvensional dalam pemrosesan citra, sehingga memberikan hasil yang kurang baik. Penggunaan metode yang lebih maju seperti machine learning atau deep learning bisa saja lebih mampu dalam menangani citra dengan derau hujan yang lebih banyak dan tebal.

UCAPAN TERIMA KASIH

Dengan selesainya tugas makalah ini, penulis mengucapkan rasa syukur kepada Tuhan Yang Maha Esa, atas berkat dan karunia-Nya penulis dapat menyelesaikan makalah ini dengan baik. Penulis juga berterima kasih kepada orang tua dan teman-teman yang telah membantu penulis dalam memberikan dukungan dan doa sehingga memungkinkan penulis dalam menyelesaikan makalah ini. Penulis juga berterima kasih kepada Bapak Dr. Ir.. Rinaldi Munir, M.T. selaku dosen yang telah mengajar penulis selama perkuliahan mata kuliah IF4073 Pemrosesan Citra Digital. Semoga penulisan makalah ini memberikan manfaat kepada pembaca, serta memberikan kontribusi kepada penelitian dalam lingkup pemrosesan citra digital secara umum.

PRANALA GITHUB PROGRAM

Kode program dapat diakses pada pranala berikut:

<https://github.com/kevinjohn01/IF4073-Makalah>


REFERENSI

- [1] Munir, Rinaldi. 2024 'Image Enhancement'. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/09-Image-Enhancement-Bagian2-2024.pdf>, diakses pada 15 Januari 2025
- [2] Munir, Rinaldi. 2024 'Transformasi Citra'. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/11-Transformasi-Citra-2024.pdf>, diakses pada 15 Januari 2025
- [3] Munir, Rinaldi. 2024 'Penapisan Citra dalam Ranah Frekuensi'. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/12-Penapisan-Citra-dalam-Ranah-Frekuensi-2024.pdf>, diakses tanggal 15 Januari 2025
- [4] <https://www.geeksforgeeks.org/python-bilateral-filtering/>, diakses pada 15 Januari 2025

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Januari 2024



Kevin John Wesley Hutabarat
13521042