



# Road Network Detection from Satellite Image

---

Kelompok B6

- 2006533181 - Ahmad Naufan Wicaksonoputra
- 2106638173 - Ardian
- 2106637580 - Ikra Bhaktiananda
- 2106752344 - Glan Harith Teguh
- 1806186875 - Carissa Syieva Q

# Pendahuluan

# Pendahuluan

**Proyek ini meneliti berbagai macam pendekatan untuk melakukan segmentasi jalan dari citra/gambar satelit.**

Proses segmentasi melibatkan pengelompokan objek untuk memisahkan jalanan dari latar belakang gambar, namun kompleksitasnya meningkat karena variasi lingkungan, infrastruktur, resolusi rendah, pencahayaan tidak ideal, dan gangguan lainnya.

Tujuan penelitian adalah menjelajahi metode segmentasi jalanan yang efektif, dengan menerapkan pendekatan konvensional dan deep learning.

Terdapat batasan pada penelitian ini, termasuk penggunaan citra satelit terbatas pada area perkotaan, keterbatasan sumber daya komputasi, dan penggunaan sebagian dataset pada metode konvensional karena kendala waktu, sumber daya, dan implementasi.

# Studi Literatur

# Filter Gabor

Gabor merupakan metode untuk mengekstrak tekstur suatu objek dari citra. Filter Gabor  $G(x, y)$  didefinisikan sebagai fungsi kernel gaussian yang dimodulasi oleh gelombang sinusoidal.

$$G(x, y, f, \theta) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \cos(2\pi f x'),$$

$$x' = x \cos(\theta) + y \sin(\theta),$$

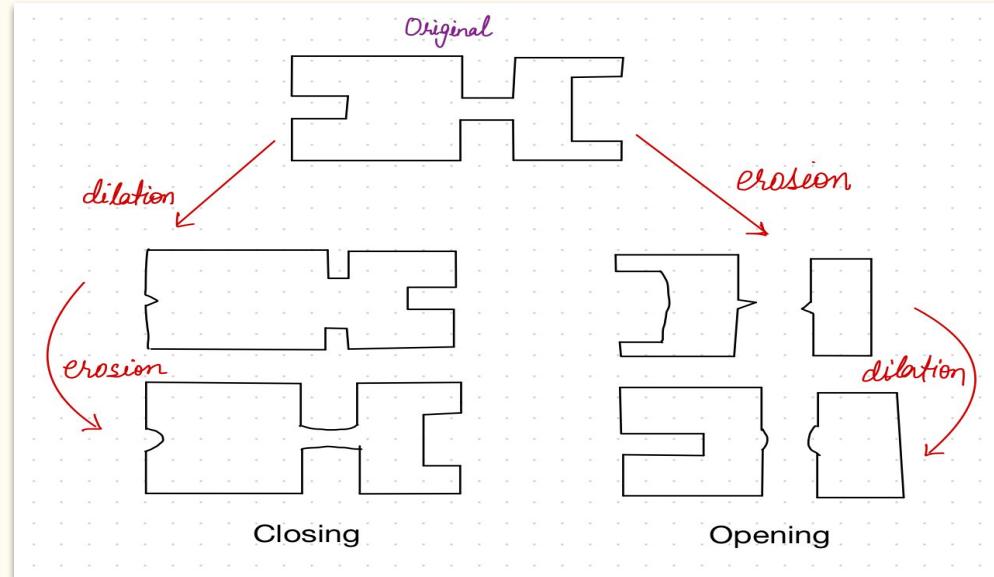
$$y' = y \cos(\theta) - x \sin(\theta).$$

- $x, y$  = Titik tengah dari filter
- $f$  = Frekuensi spasial
- $\theta$  = Orientasi
- $\sigma$  = Standar deviasi kernel

# Morfologi

Morfologi adalah pendekatan untuk mengekstrak fitur citra berdasarkan struktur geometris objek, yang disebut sebagai elemen struktur atau strel.

Melibatkan operasi seperti erosi, dilasi, atau gabungan keduanya, yang dilakukan sesuai dengan bentuk strel yang digunakan.



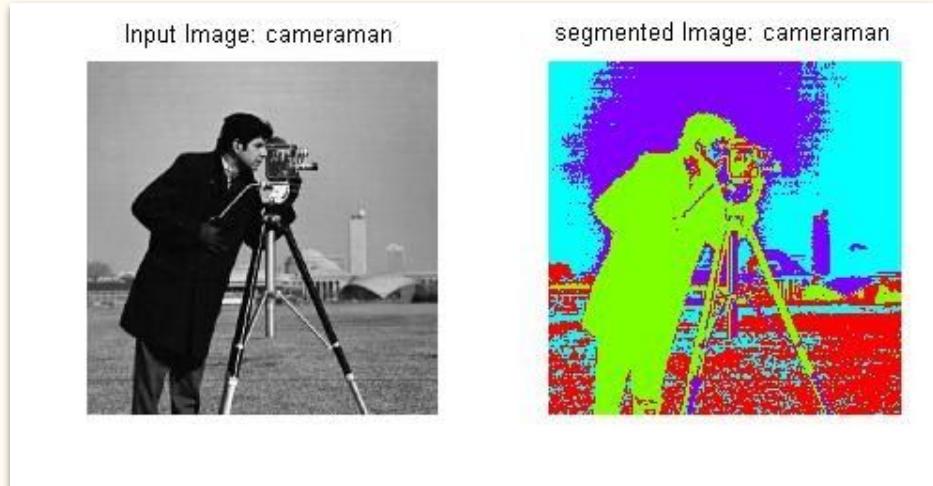
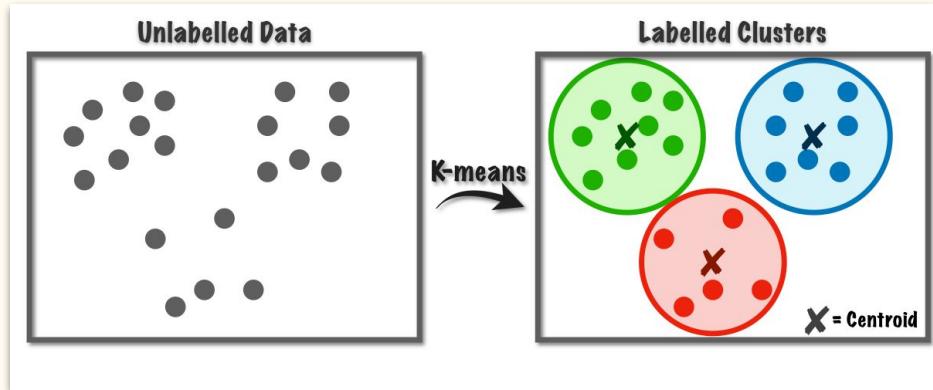
# SLIC

Simple Linear Iterative Clustering (SLIC) adalah metode untuk membentuk superpixel pada citra dengan menggunakan satu parameter  $k$  untuk menentukan jumlah superpixel yang diinginkan. Prosesnya melibatkan penetapan titik tengah awal cluster, pengelompokan piksel ke dalam cluster terdekat dalam area pencarian, dan pembaruan titik tengah cluster menjadi rata-rata piksel di dalamnya



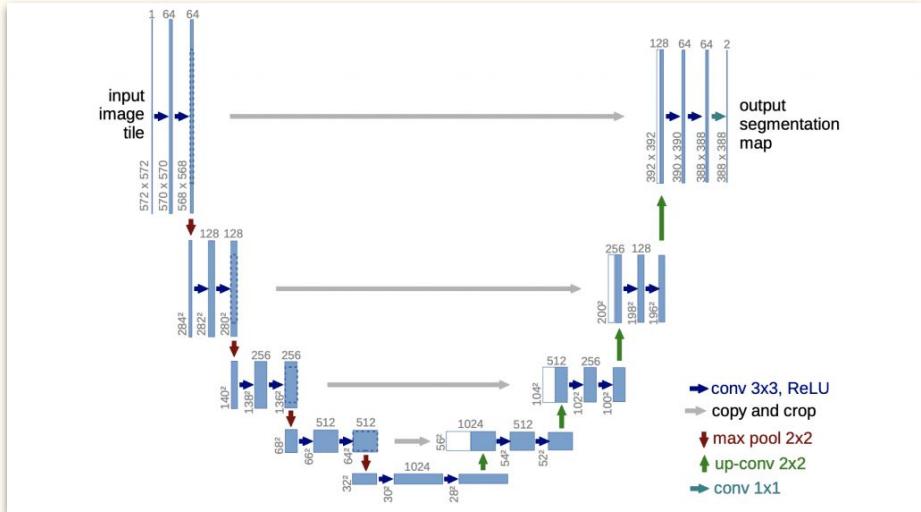
# K-means Clustering

K-Means clustering adalah algoritma klasifikasi yang mengelompokkan data ke dalam cluster berdasarkan jarak antar data, dengan masing-masing cluster memiliki titik tengah yang disebut centroids.



# U-Net

Unet merupakan arsitektur deep learning yang dirancang untuk segmentasi gambar. Dengan arsitektur ini, citra mengalami proses downsampling dan upsampling melalui konvolusi pada setiap tahapnya. Downsampling dilakukan hingga mencapai resolusi tertentu, dan kemudian dilakukan upsampling untuk mengembalikannya ke resolusi asal.

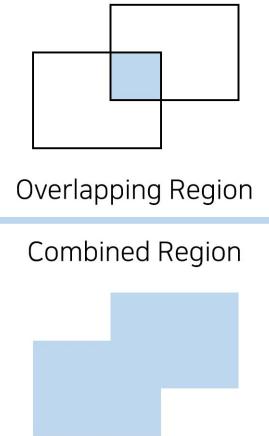


Sky	Building	Pole	Road	Sidewalk	Vegetation	Sign	Fence	Car	Pedestrian	Cyclist	Void
-----	----------	------	------	----------	------------	------	-------	-----	------------	---------	------

# Intersection over Union

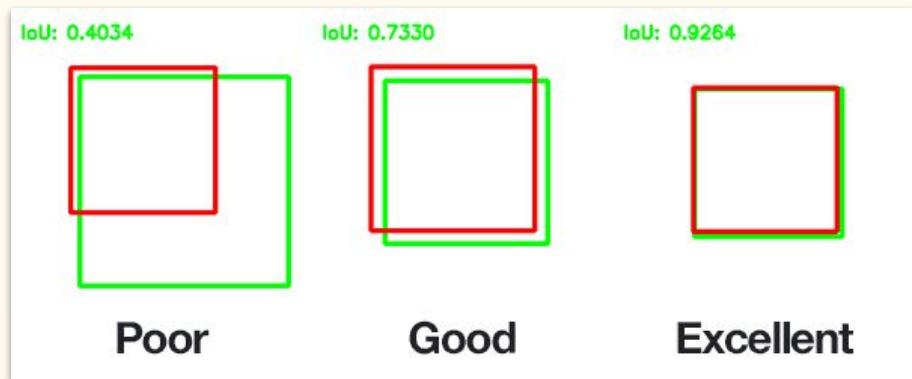
IoU atau Jaccard Index adalah metrik evaluasi untuk kemiripan dua objek, membandingkan mereka dengan menarik kotak yang mencakup keduanya dan mengukur persentase area tumpang tindih. Metrik ini dinormalisasi untuk tidak terpengaruh oleh ukuran absolut objek. Karena itu, IoU merupakan metrik yang cocok digunakan pada citra satelit, terutama pada objek seperti jalanan yang cenderung kecil.

$$\text{IOU} =$$



# Robustness

Robustness, dalam evaluasi hasil pemrosesan citra, menunjukkan sejauh mana suatu algoritma dianggap kuat jika menghasilkan error di bawah batas tertentu saat ada noise pada citra. Pada penelitian ini, evaluasi robustness dilakukan dengan membandingkan nilai IoU antara dataset yang mengandung noise tambahan dan dataset aslinya.

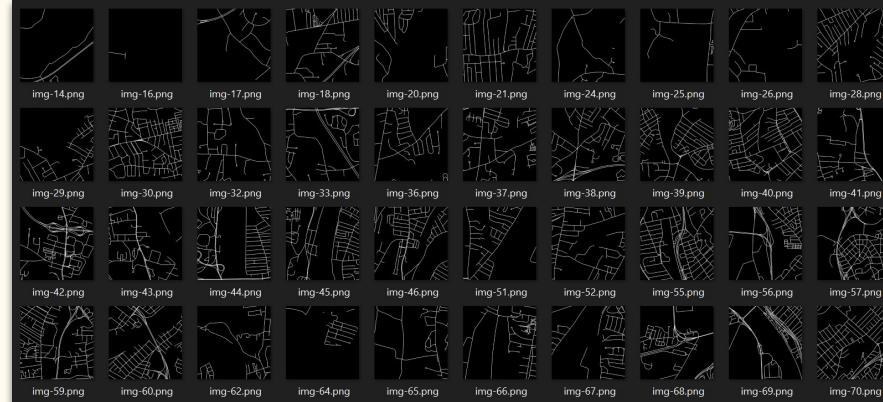
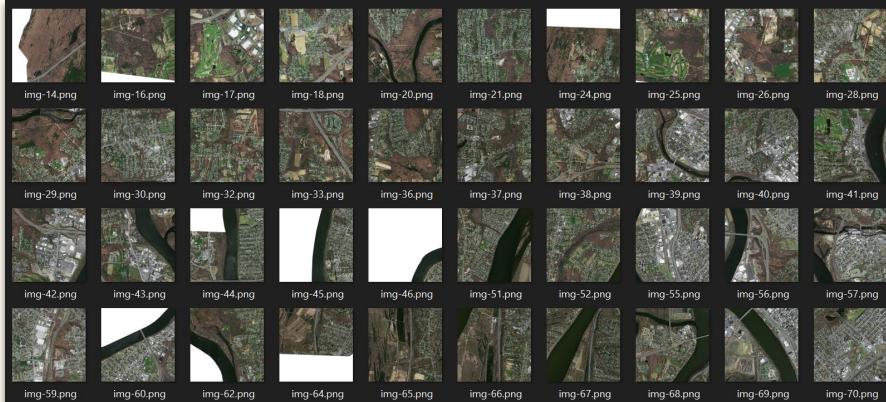


# Metodologi

# Dataset

Penelitian ini menggunakan ***Massachusetts Road and Building Datasets***.

Untuk mengevaluasi *robustness* dari metode konvensional dan *deep learning*, tiap citra diberikan ***gaussian noise*** dengan variansi sebesar **0.01**.



# Graph-based

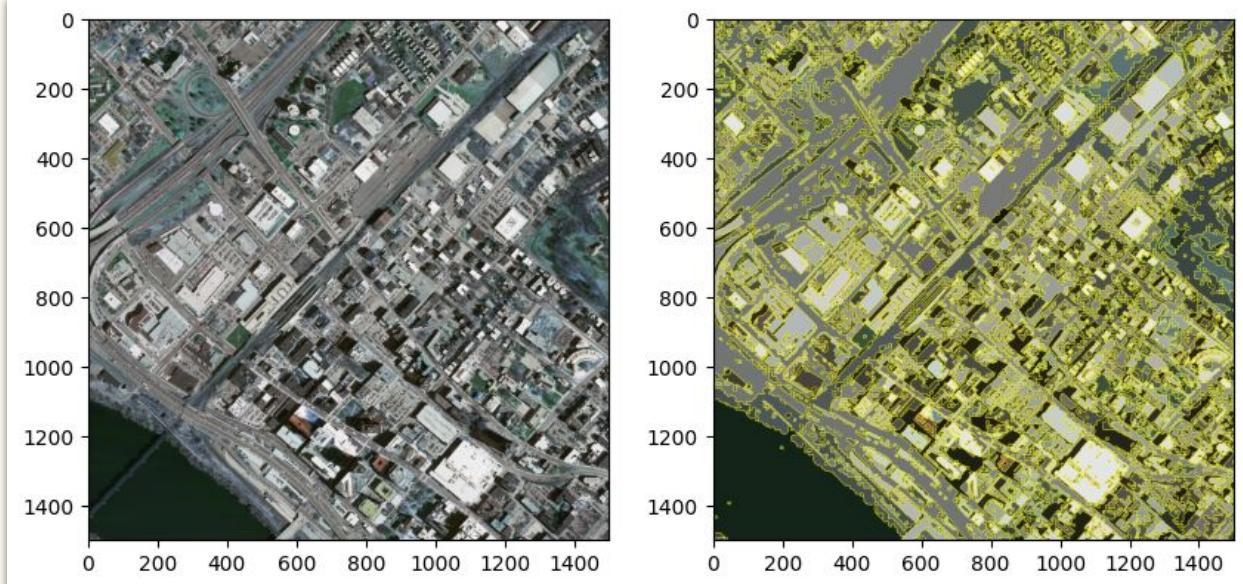
Proses segmentasi secara *graph-based*.

1. SLIC
2. Ekstraksi fitur dengan Gabor
3. Ekstraksi fitur dengan morfologi
4. Inisialisasi jalan awal
5. Filtering SLIC
6. Graph merging
7. Thresholding

# Graph-based

## 1. SLIC

Mengubah citra asli  
menjadi *superpixels*

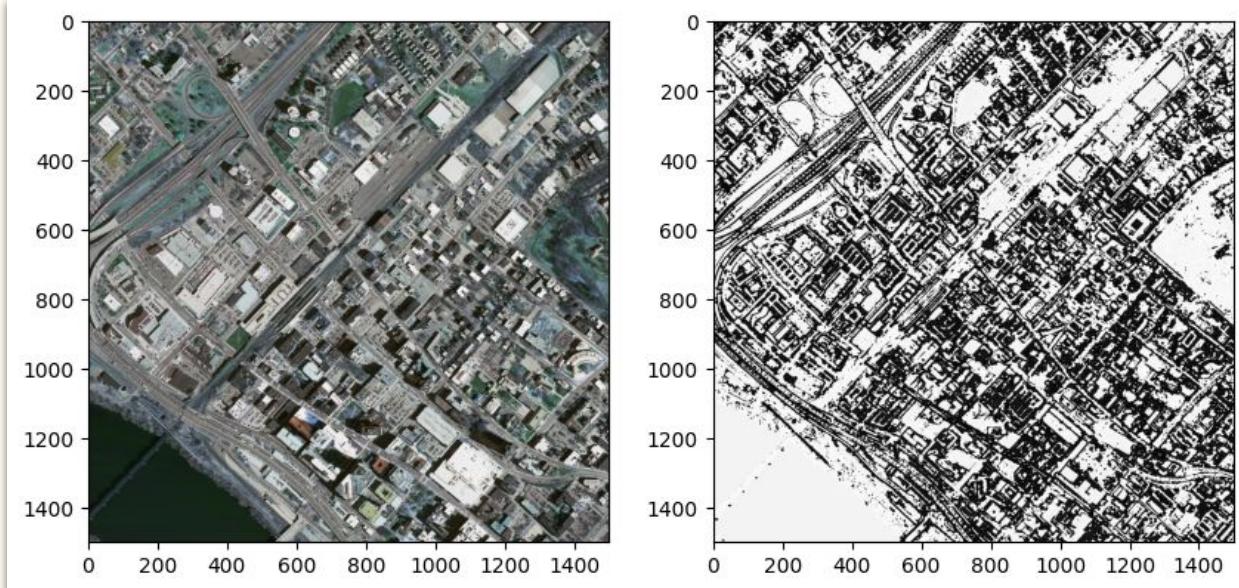


`slic(img, n_segments=36000, compactness=25, sigma=1)`

# Graph-based

## 2. Ekstraksi fitur Gabor

Mengekstrak fitur dari citra asli menggunakan Gabor lalu dilakukan PCA 1 komponen dan *thresholding*

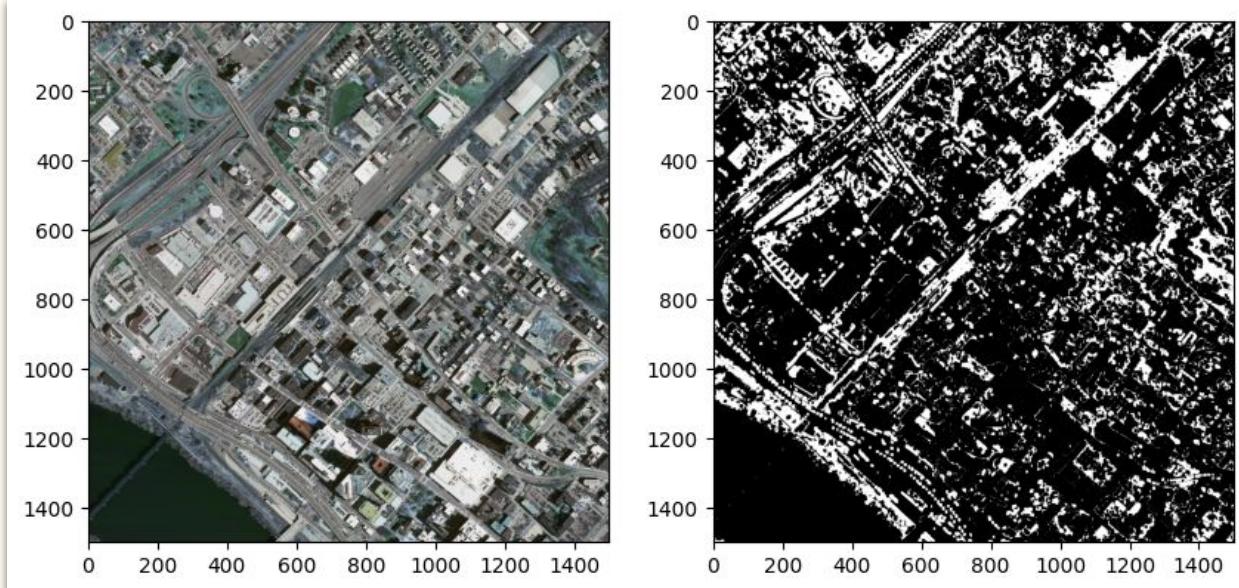


```
freqencies = [0.1, 0.2, 0.3, 0.4, 0.5]
thetas = np.arange(0, np.pi, np.pi / 6)
```

# Graph-based

## 3. Ekstraksi fitur Morfologi

Mengekstrak fitur  
dari citra asli  
menggunakan *closing*  
lalu dilakukan PCA 1  
komponen dan  
*thresholding*



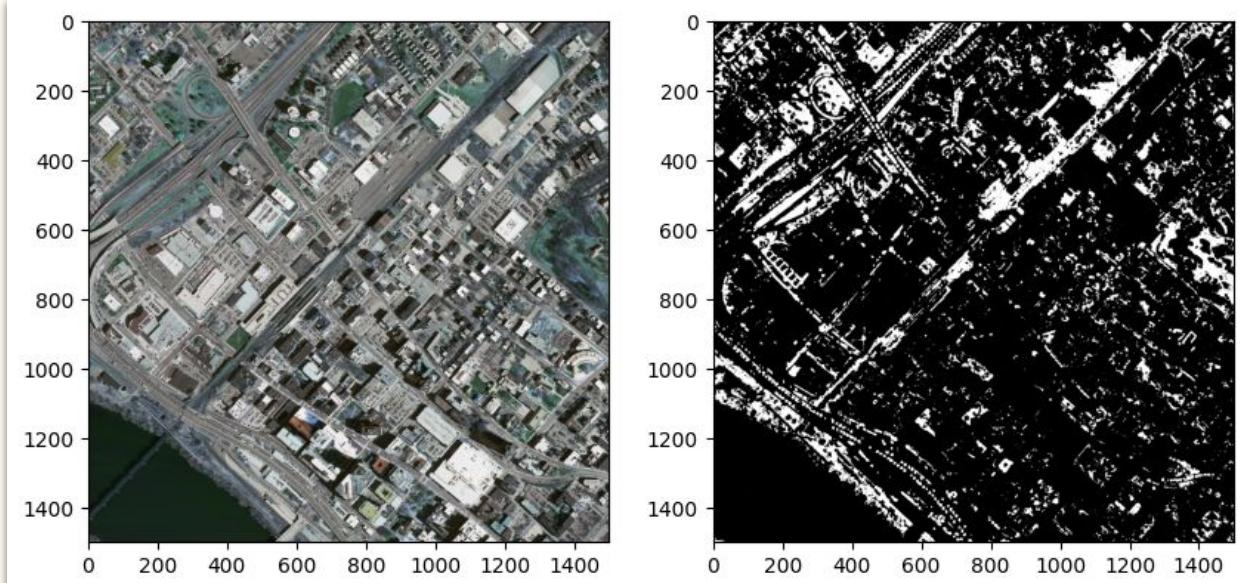
```
horizontal = morphology.rectangle(3, 7)  
vertical = morphology.rectangle(7, 3)
```

# Graph-based

## 4. Inisialisasi jalan awal

Menggabungkan hasil dari tahap 2 dan 3

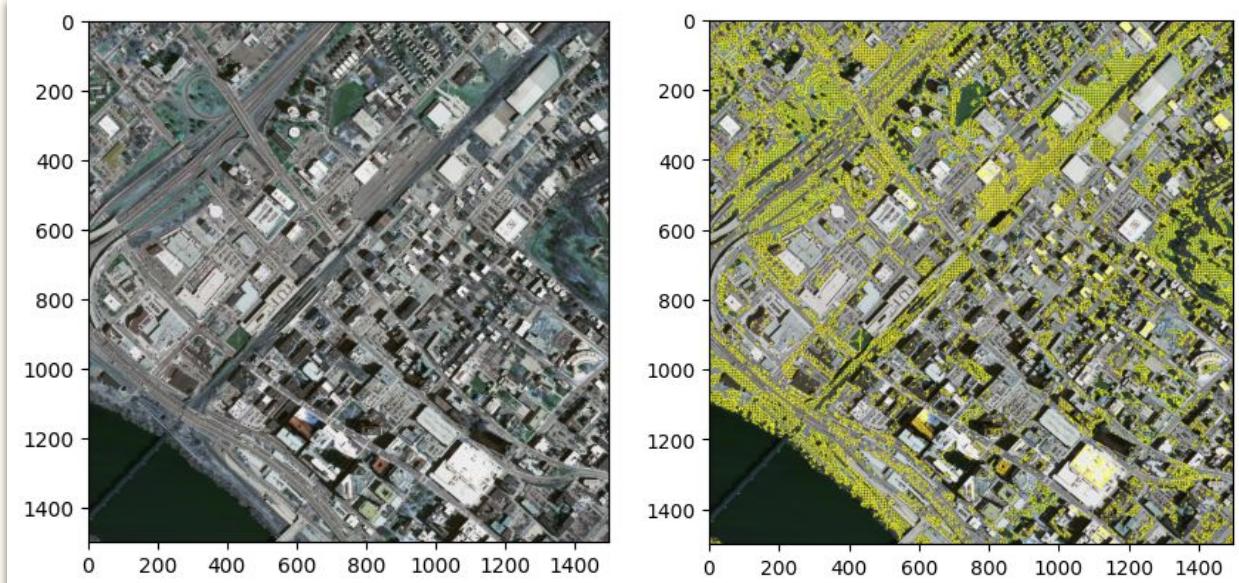
(irisan dari tahap 2 dan 3)



# Graph-based

## 5. Filtering SLIC

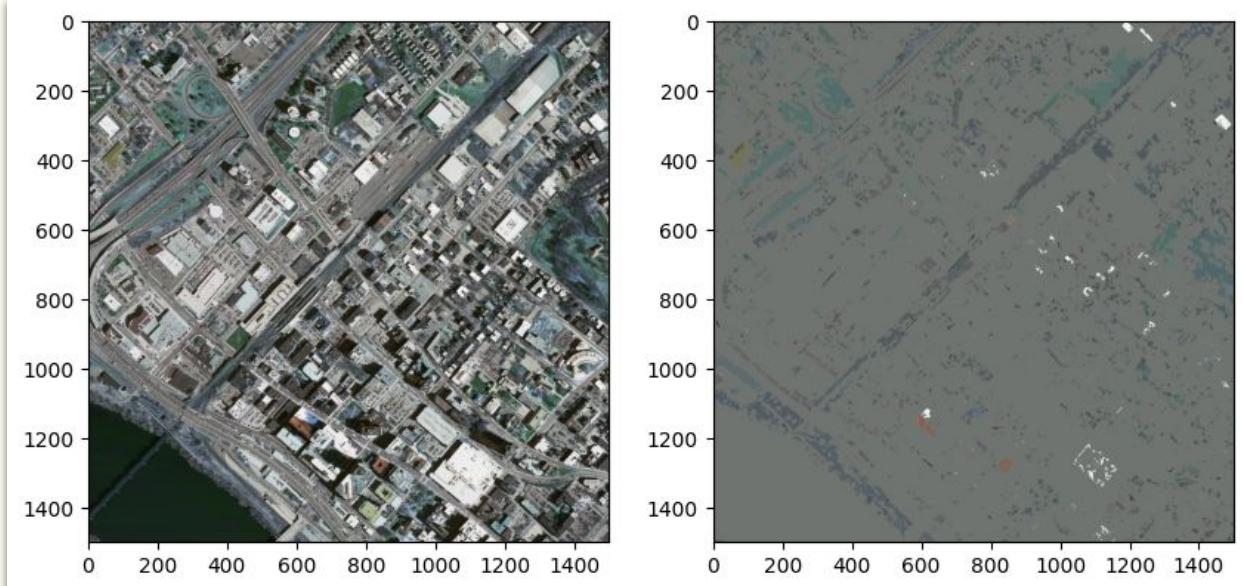
*Superpixels* pada tahap 1 yang tidak ada pada tahap 4 dibuang



# Graph-based

## 6. Graph Merging

Hasil tahap 4 diubah ke dalam bentuk RAG dan dilakukan *merging* berdasarkan rataan nilai RGB

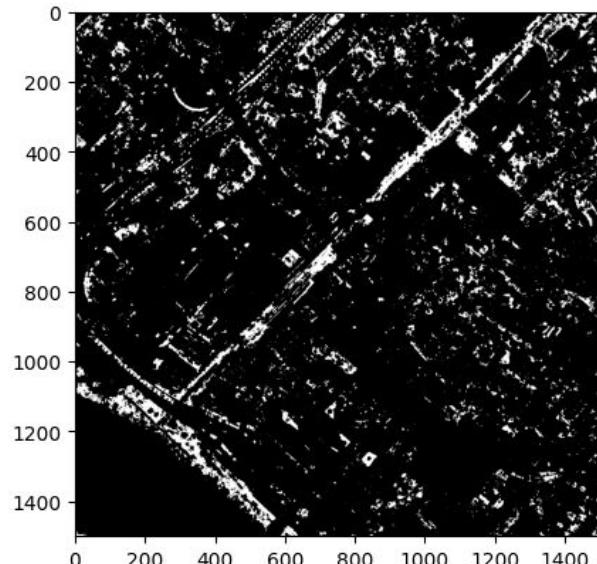
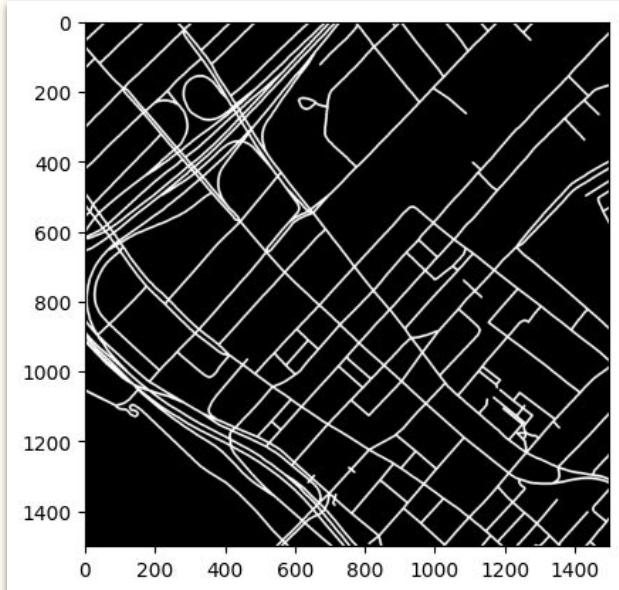


$merge\_threshold = 20$

# Graph-based

## 7. Thresholding

Hasil tahap 6  
dilakukan  
*thresholding* untuk  
mendapatkan citra  
biner



$threshold = 0.4$

# Math Morphology & Connected Components

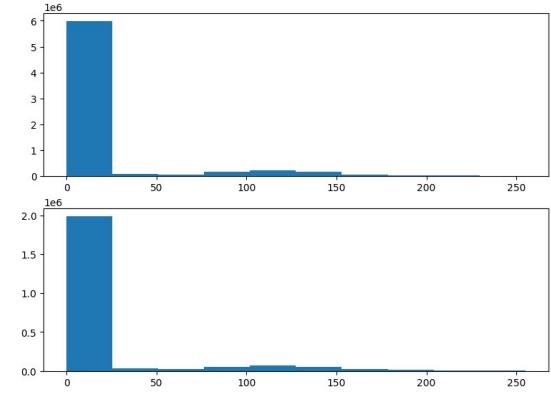
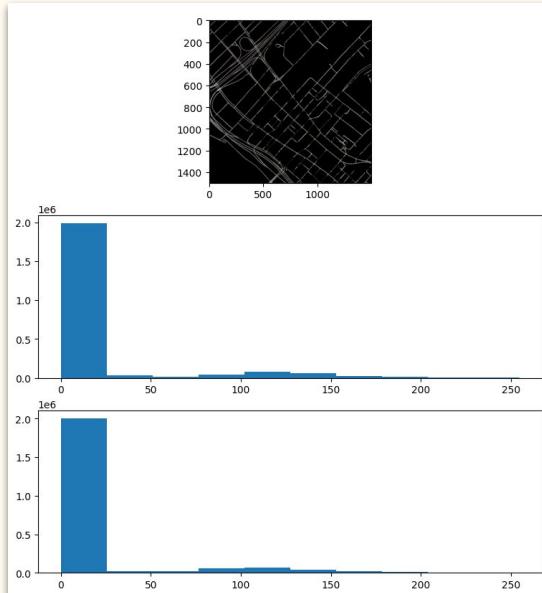
Proses segmentasi secara morfologi matematis dan *connected components*.

1. Feature extraction dengan RGB channels
2. Segmentasi RGB
3. Konversi ke gray
4. Closing
5. Konversi ke binary
6. Connected component filtering

# Math Morphology & Connected Components

## 1. Feature extraction dengan RGB channels

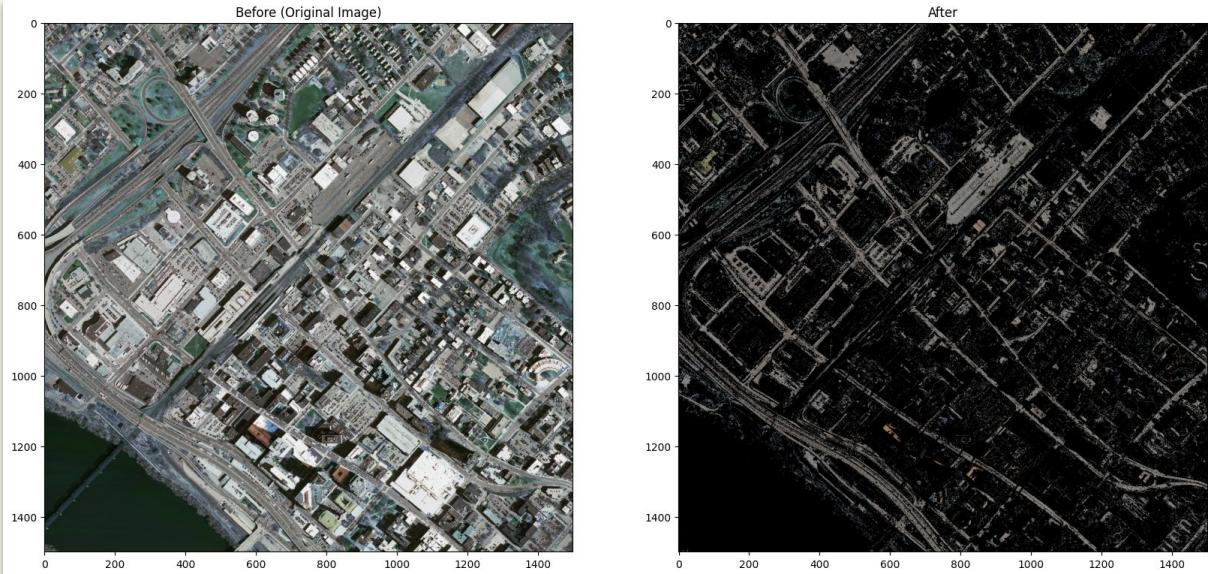
Diambil kemungkinan range nilai R, G, dan B dari warna jalan, dengan melakukan masking menggunakan ground truth yang ada.



# Math Morphology & Connected Components

## 2. Segmentasi RGB

Diambil kemungkinan range nilai R, G, dan B dari warna jalan.

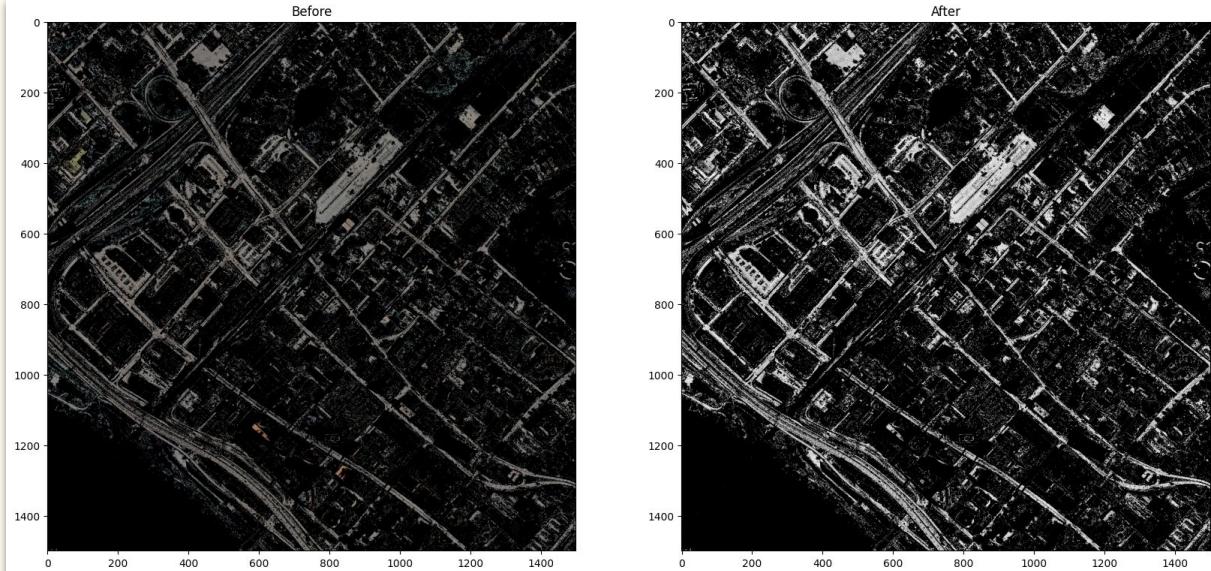


$$\begin{aligned} R &= 100 \text{ to } 175 \\ G &= 100 \text{ to } 150 \\ B &= 80 \text{ to } 140 \end{aligned}$$

# Math Morphology & Connected Components

## 3. Konversi ke gray

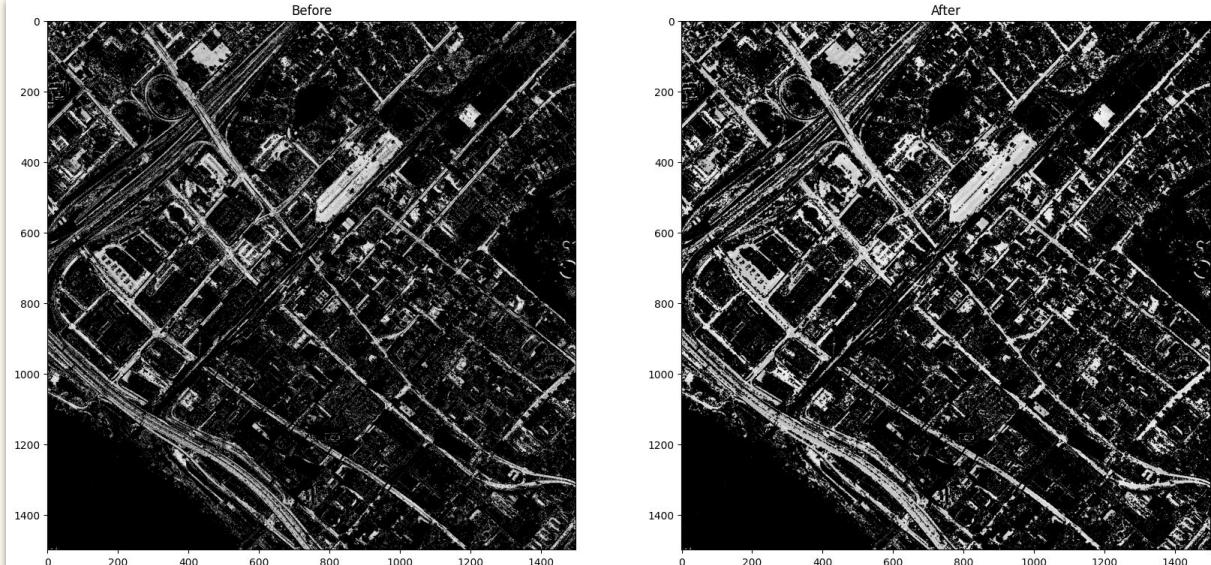
Hasil segmentasi  
warna kemudian  
dikonversi ke gray



# Math Morphology & Connected Components

## 4. Closing

Kemudian diterapkan closing untuk memperjelas objek yang dominan dan sedikit meredupkan objek objek noise

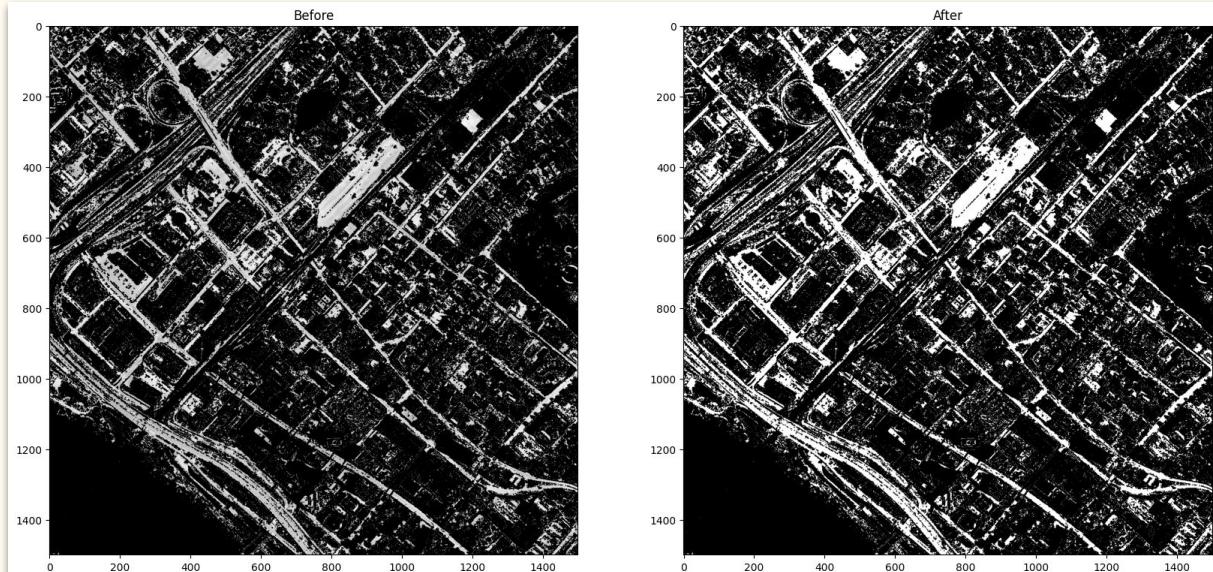


$Strel = disk(1)$

# Math Morphology & Connected Components

## 5. Konversi ke binary

Gambar kemudian di konversi ke binary dengan binary thresholding untuk diproses lebih lanjut

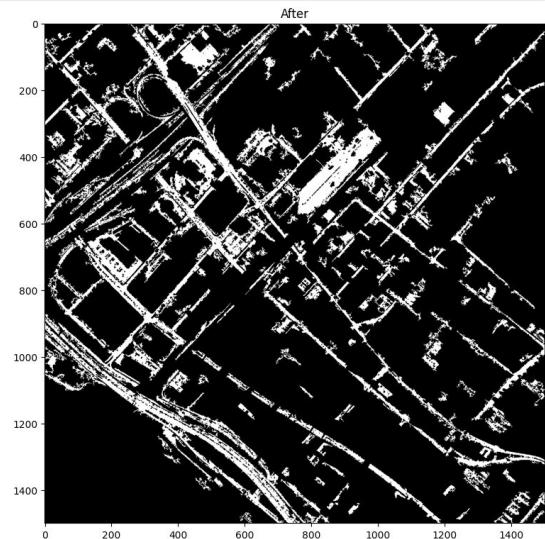
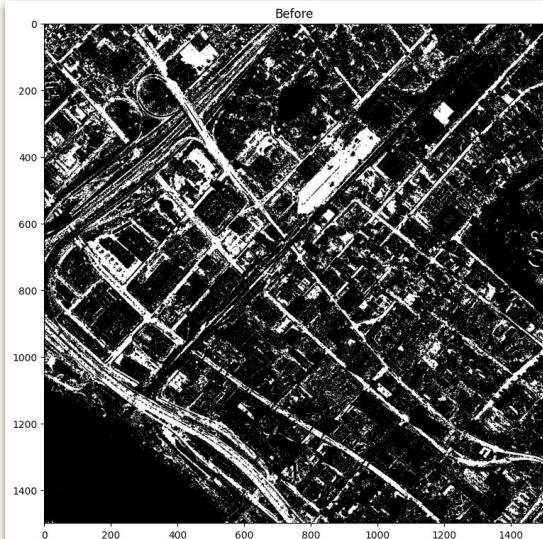


$threshold = 80$

# Math Morphology & Connected Components

## 6. Connected component filtering

Gambar binary kemudian di-filter komponennya berdasarkan besar koneksi pixelnya, komponen yang berkoneksi besar seperti jalan akan dipertahankan.



*Size threshold = 200*

# K-Means

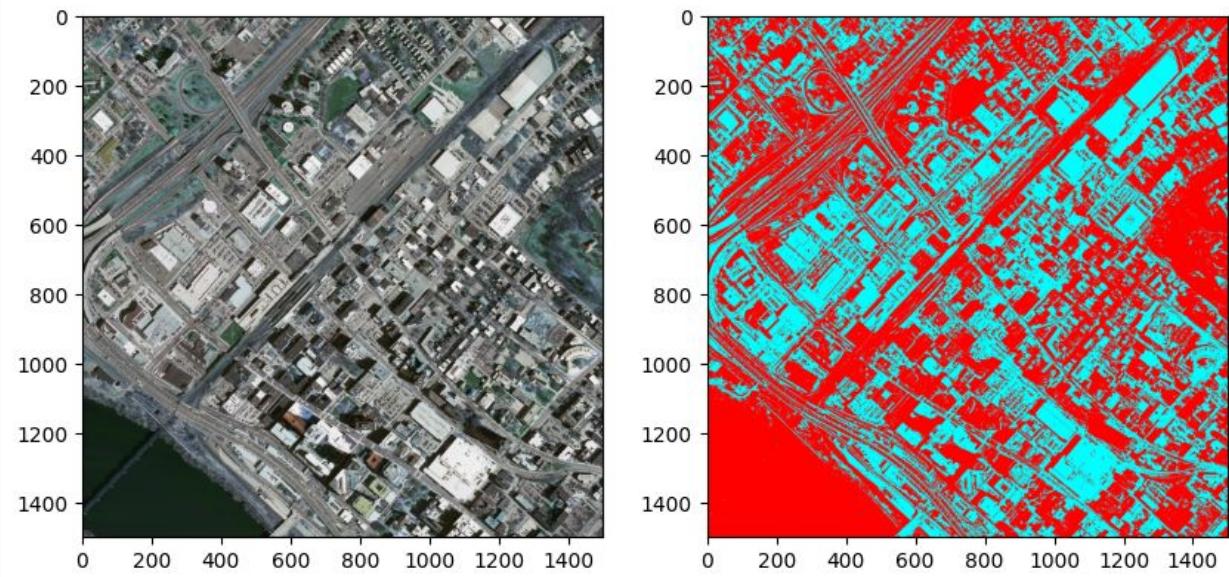
## Proses segmentasi menggunakan K-Means.

1. Melakukan *clustering* dengan K-Means
2. Segmentasi hasil *clustering*
3. Melakukan morfologi *opening*
4. Melakukan *skeletonization*
5. Melakukan morfologi *dilation*

# K-Means

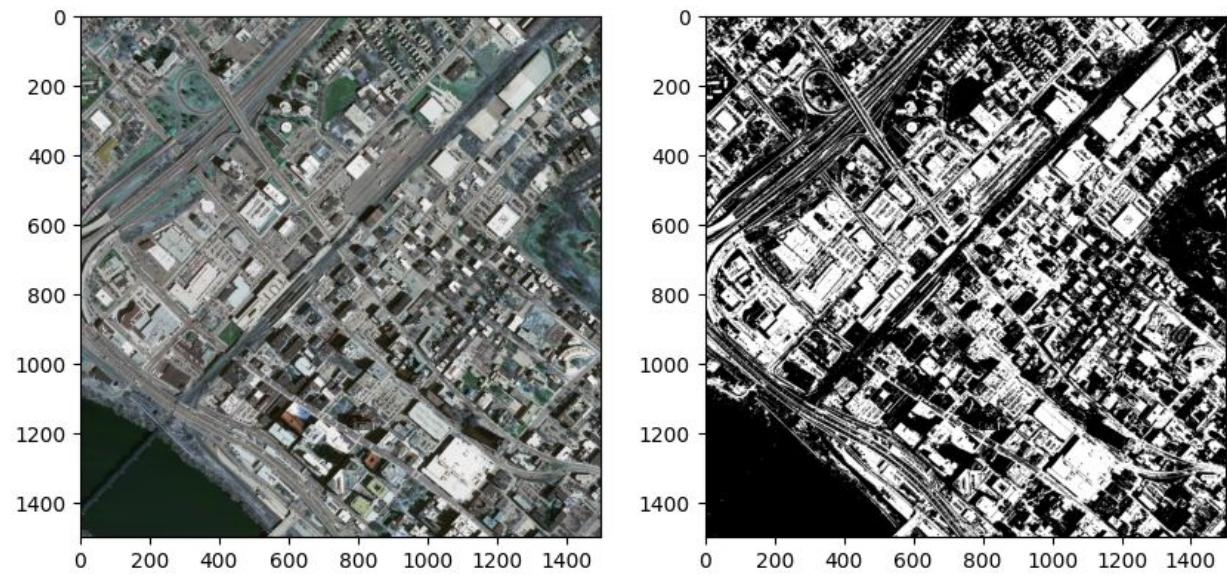
## 1. Melakukan clustering dengan K-Means

*Clustering dengan jumlah cluster 2*



# K-Means

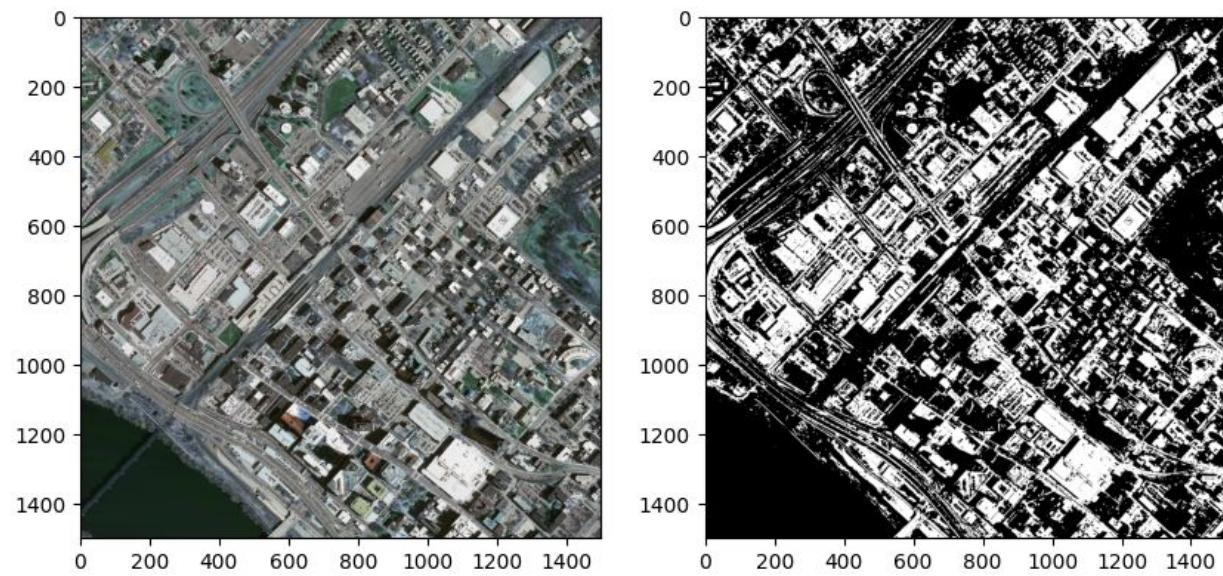
## 2. Segmentasi hasil clustering



# K-Means

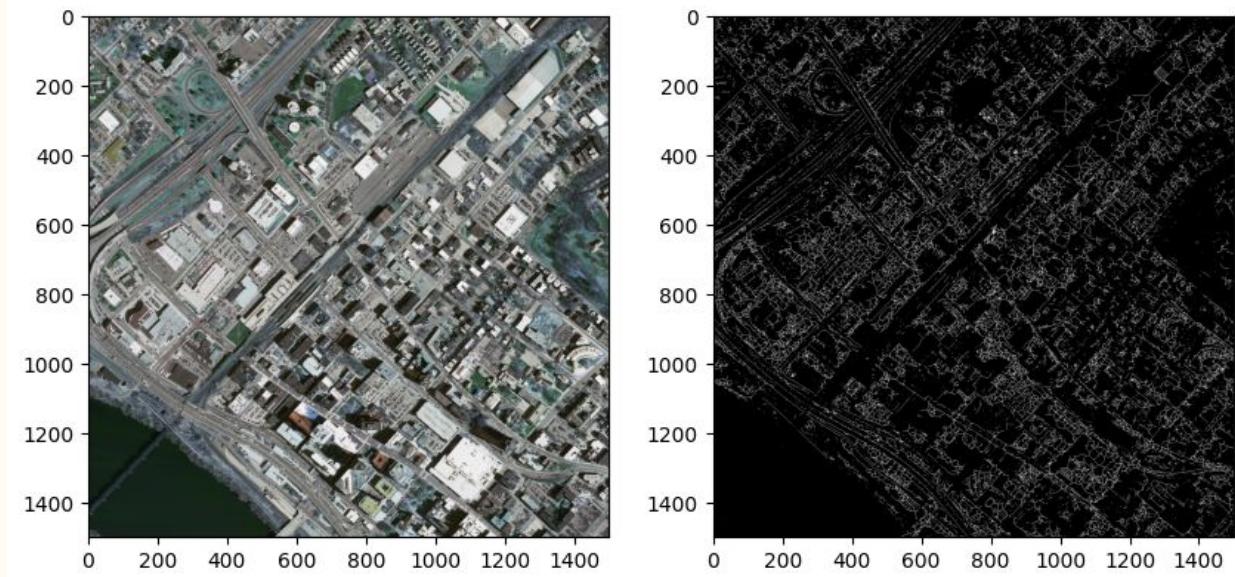
## 3. Melakukan morfologi opening

*Opening (erosion to dilation)* dilakukan menggunakan strel  $2 \times 2$



# K-Means

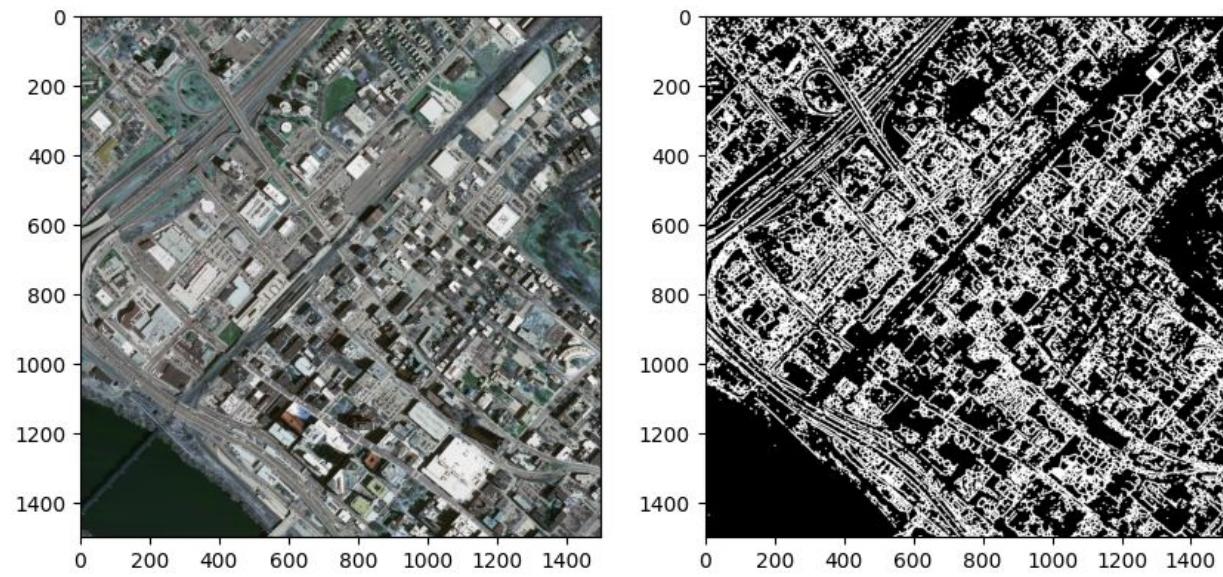
## 4. Melakukan skeletonization



# K-Means

## 5. Melakukan morfologi dilation

*Dilation* dilakukan menggunakan strel 4x4



## U-Net

**Penelitian menggunakan U-Net dengan program yang sudah ada dengan beberapa modifikasi.**

Dilakukan modifikasi pada model dengan mengubah target segmentasi dari jalan, bangunan dan latar menjadi jalan dan latar saja. Model memiliki **empat tahap** untuk *upsampling* dan *downsampling*.

## Downsampling

Untuk setiap tahap, dilakukan dua konvolusi. Pada konvolusi pertama, jumlah fitur digandakan. Sedangkan pada konvolusi kedua, jumlah fitur tidak diubah (contohnya, jika jumlah fitur pada tahap sebelumnya adalah 2, maka keluarannya akan menjadi 4). Kemudian dilakukan *max pooling 2x2* dan *dropout*. Setelah empat tahap selesai, dilakukan dilakukan konvolusi lagi untuk menggandakan jumlah fitur dan dilanjut dengan *upsampling*.

## Upsampling

Untuk setiap tahap, dilakukan dekonvolusi, dilanjut dengan *dropout* dan terakhir dilakukan dua konvolusi (sama seperti *downsampling*) untuk mengurangi jumlah fitur. Setelah tahap *upsampling* selesai, dilakukan dilakukan konvolusi sekali lagi untuk mengubah jumlah fitur sesuai dengan jumlah kelas segmentasi, yaitu dua.

# U-Net

## Parameter

- *Hidden* (jumlah *hidden layer*): 64
- *Kernel size* (ukuran *kernel* konvolusi): 3
- *Padding* (ukuran *padding* konvolusi): 1
- P1 (*dropout rate* 1): 0.25
- P2 (*dropout rate* 2): 0.5
- *Learning rate*: 0.0001

## Training

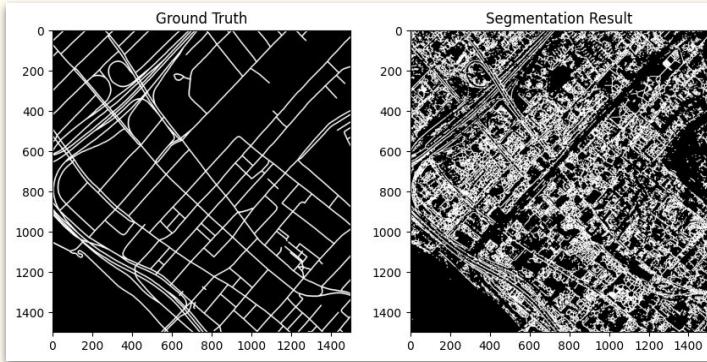
Pada tahap ini, dilakukan iterasi sebanyak *epoch* yang ditentukan. Untuk setiap iterasi, citra *training* dimasukkan ke dalam model U-Net. Hasil kemudian dibandingkan dengan *ground truth* untuk mendapatkan nilai *loss*-nya. Kemudian dilakukan *backpropagation* agar model dapat belajar dari hasil tersebut.

Setelah selesai, dilakukan validasi terhadap model. Jika metrik pada tahap validasi memiliki hasil terbaik, model pada tahap tersebut akan disimpan sebagai representasi model terbaik saat ini.

# Hasil dan Analisis

# Metode Konvensional

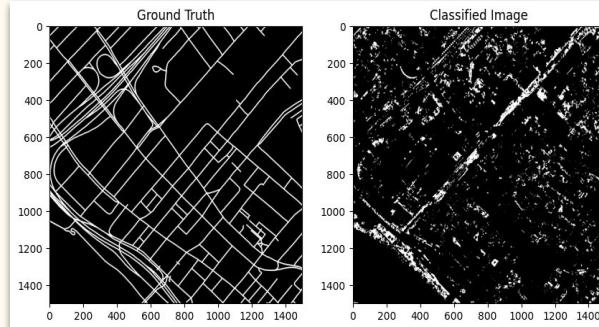
## K-means clustering



IoU = 0.0732

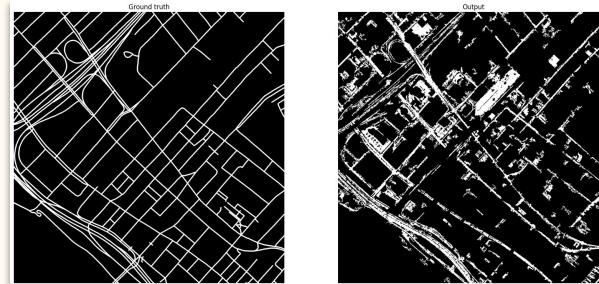
IoU noise = 0.0732

## Graph based



IoU = 0.017

## Math morphology & connected components



IoU = 0.17

\*on average

\* dataset yang digunakan untuk testing metode ini hanya sebagian karena keterbatasan waktu dan resource

## Metode Konvensional

Dilihat dari IoU, metode *graph-based* dan K-means masih sangat kurang sedangkan *Math Morphology* cukup bagus.

Hasil segmentasi *graph-based* tidak berhasil dalam mengidentifikasi jalanan. Dan K-Means kurang berhasil dalam mengabaikan bangunan. Sedangkan segmentasi *math morphology* mampu megidentifikasi jalanan, namun kurang mampu mengidentifikasi ketersambungan jalan, sehingga terdapat jalan yang putus-putus.

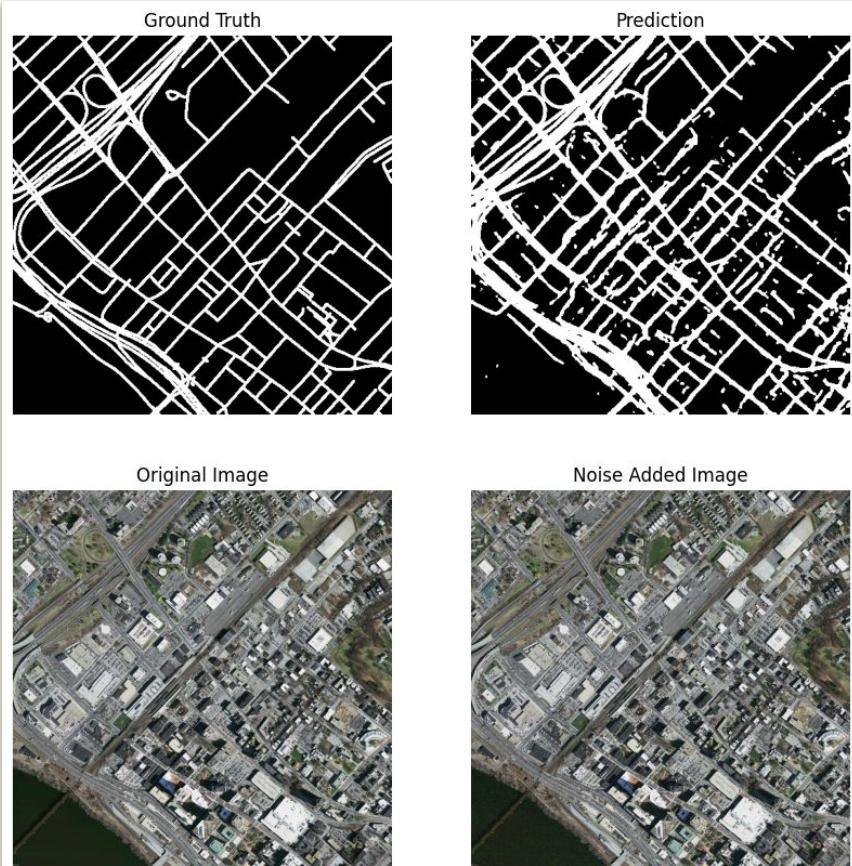


# Metode Deep Learning

## U-Net



$\text{IoU} = 0.831$

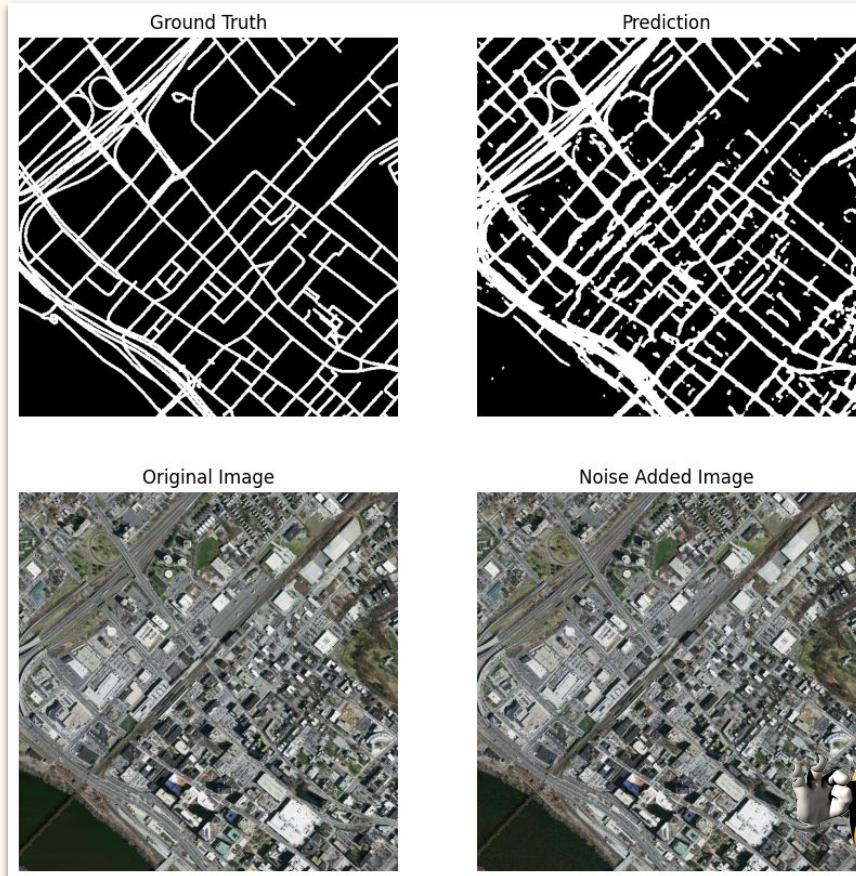


$\text{IoU noise} = 0.804$

# Metode Deep Learning

Model mampu melakukan segmentasi jalan terisolasi dengan baik, namun masih kesulitan dengan jalanan yang saling berdekatan. Masih terdapat beberapa *noise* dari bangunan sekitar pada prediksi citra asli maupun citra dengan *noise*.

Model ini dapat dikatakan *robust* karena memiliki IoU yang tidak jauh berbeda pada dataset dengan *noise*.

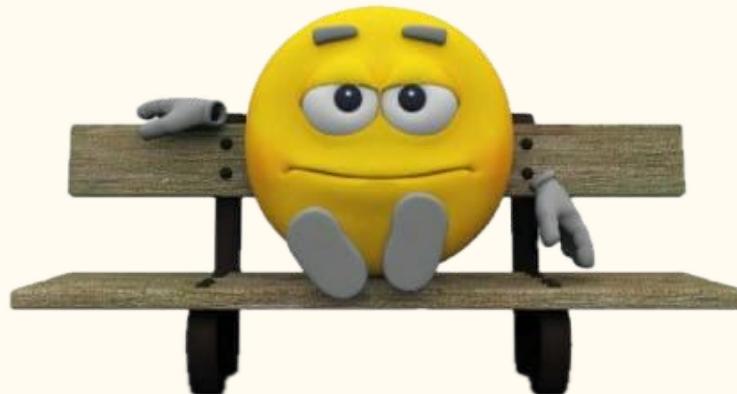


# Penutup

# Kesimpulan

Segmentasi jalan dari citra satelit dapat dilakukan dengan hasil yang cukup baik menggunakan model deep learning U-Net. Dengan waktu dan dataset yang lebih besar, model U-Net mungkin menghasilkan hasil yang lebih baik lagi.

Untuk metode konvensional, hasilnya masih sangat kurang. Hal ini dikarenakan keterbatasan penulis dalam mengimplementasikan metode-metode tersebut.



## Saran

Penelitian yang menggunakan metode konvensional dalam segmentasi jalanan membutuhkan tingkat pengetahuan yang tinggi, mewajibkan peneliti untuk menginvestasikan waktu dalam memahami dasar-dasar teknik tersebut.

Penggunaan GPU dapat mempercepat implementasi metode konvensional.

Sementara itu, untuk metode deep learning, entry level untuk hasil yang memuaskan relatif rendah, terutama jika hanya memanfaatkan model yang sudah ada. Namun, pengembangan atau pemahaman rinci terhadap cara kerja *deep learning* dari awal mungkin lebih sulit dibandingkan dengan metode konvensional.

## Refleksi Kelompok

Kami menyimpulkan bahwa metode *deep learning* seperti U-Net dianggap lebih mudah dipahami karena mengandalkan teknik simpel seperti konvolusi, aktivasi ReLU, dan pooling.

Sebaliknya, metode konvensional lebih beragam dan langsung bekerja pada tingkat piksel. Hal ini membuat *entry level* metode konvensional lebih tinggi.

Keuntungan dari *deep learning* adalah arsitekturnya yang memang dirancang untuk GPU sehingga dapat melakukan komputasi dengan lebih cepat. Sementara untuk metode konvensional, biasanya perlu dibuat dari awal (tidak menggunakan *library*) lagi untuk memanfaatkan GPU. Hal ini membuat percobaan harus dilakukan sesegera mungkin karena waktu yang dibutuhkan cukup lama jika tidak menggunakan GPU.

Terima kasih