23 - Segmentasi Citra (Bagian 2)

IF4073 Pemrosesan Citra Digital

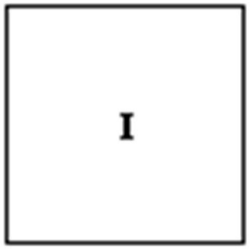
Oleh:
Dr. Ir. Rinaldi, M.T
Dr. Edward Ferdian



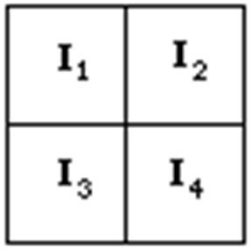
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2026

3. Split and Merge

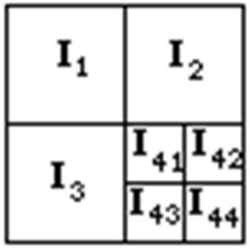
- Mengggunakan algoritma *divide and conquer*
- Citra dibagi (split) menjadi sejumlah region yang *disjoint*
- Gabung (*merge*) region-region bertetangga yang homogen



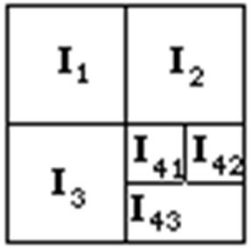
(a) Whole Image



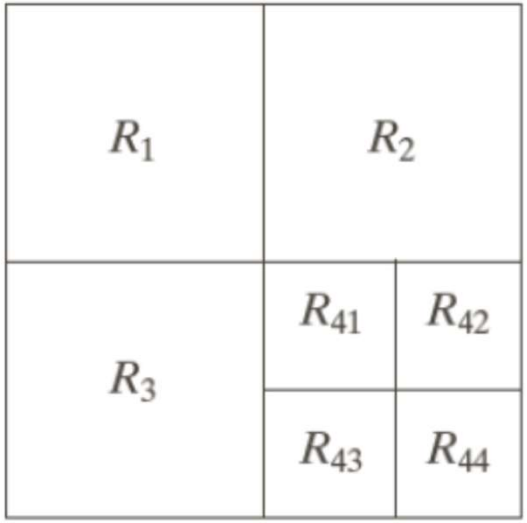
(b) First Split

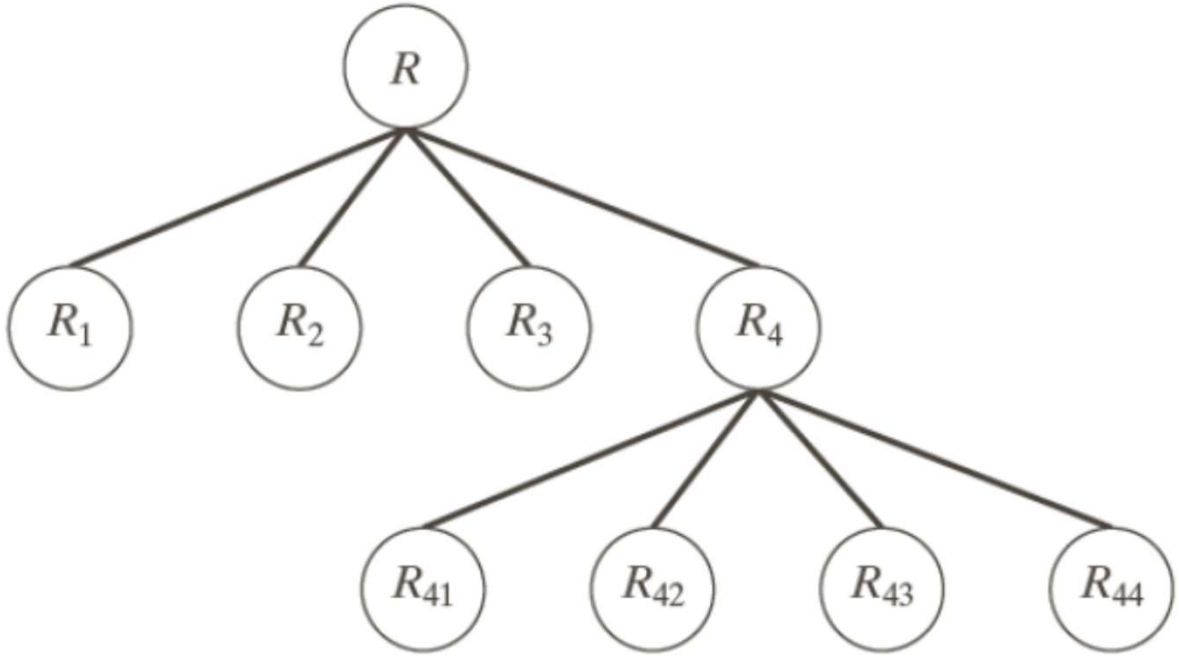


(c) Second Split



(d) Merge


R_1
R_2
R_3
R_{41}
R_{42}
R_{43}
R_{44}

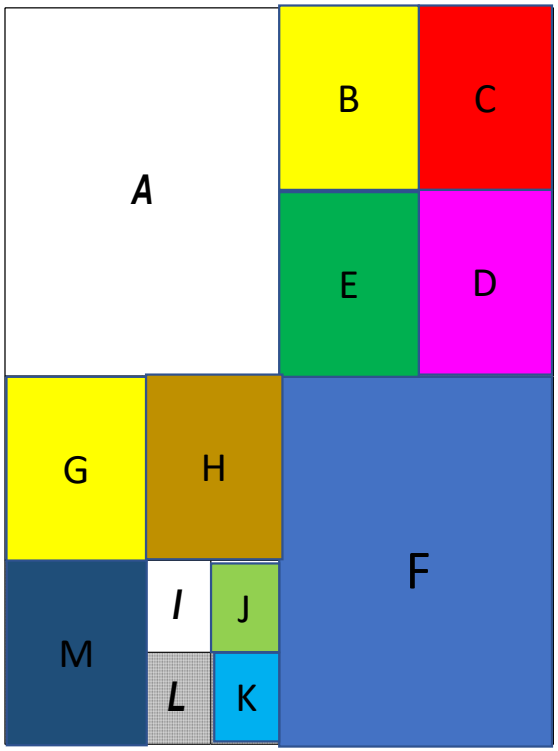

R
R_1
R_2
R_3
R_4
R_{41}
R_{42}
R_{43}
R_{44}

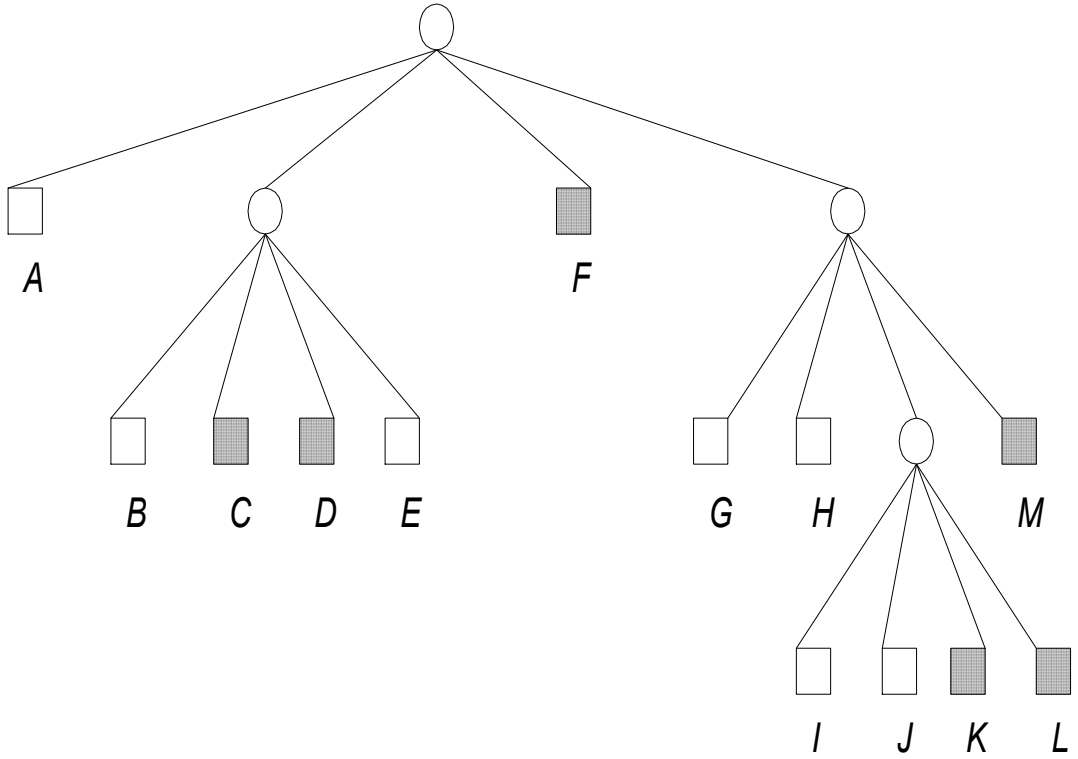
Algoritma *Split & Merge*

Given an image f and a predicate Q, the basic algorithm is:

1. $R_1 = f$
2. Subdivision in quadrants of each region R_i for which $Q(R_i) = \text{FALSE}$.
3. If $Q(R_i) = \text{TRUE}$ for every regions, merge those adjacent regions R_i and R_j such that $Q(R_i \cup R_j) = \text{TRUE}$; otherwise, repeat step 2.
4. Repeat the step 3 until no merging is possible.

Sumber: Image segmentation
Stefano Ferrari
Universit`a degli Studi di Milano
stefano.ferrari@unimi.it


A
B
C
E
D
G
H
F
M
I
J
L
K


A
F
B
C
D
E
G
H
M
I
J
K
L

16	6	22	12
14	8	9	7
7	23	22	5
6	25	4	3

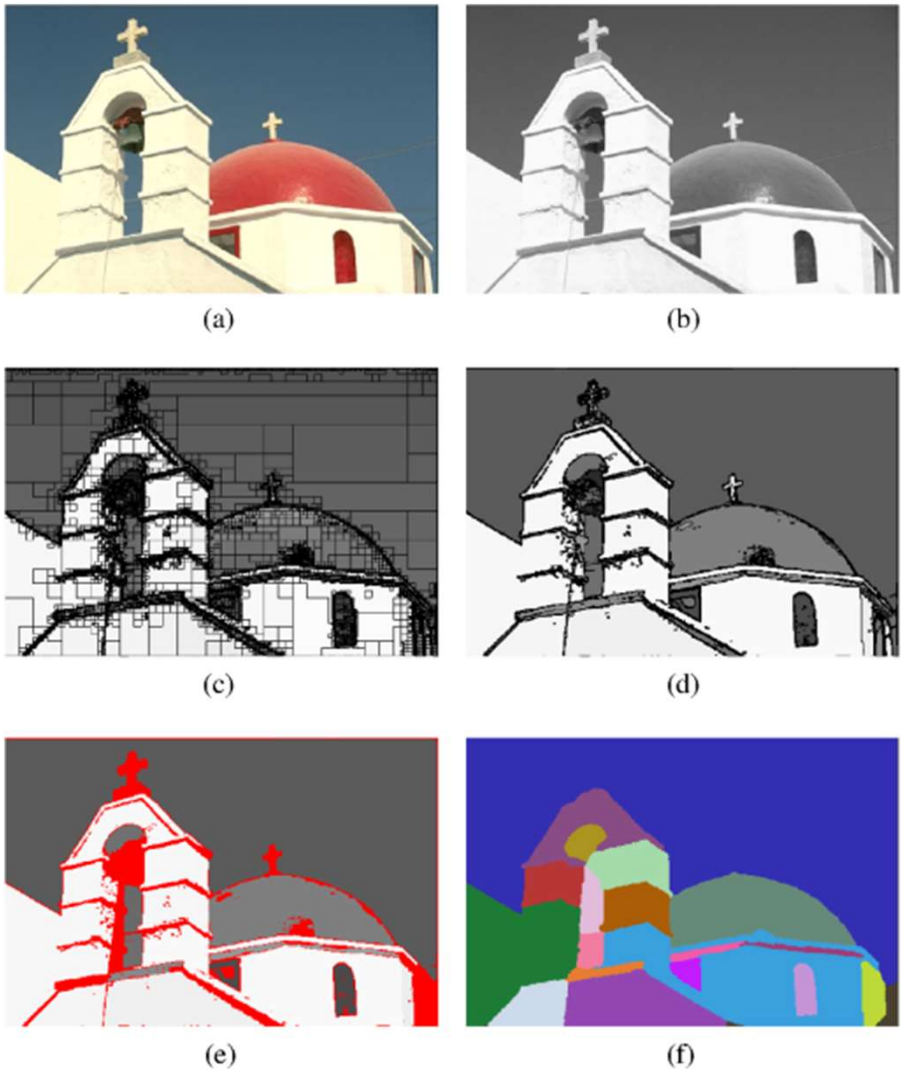


16	6	22	12
14	8	9	7
7	23	22	5
6	25	4	3



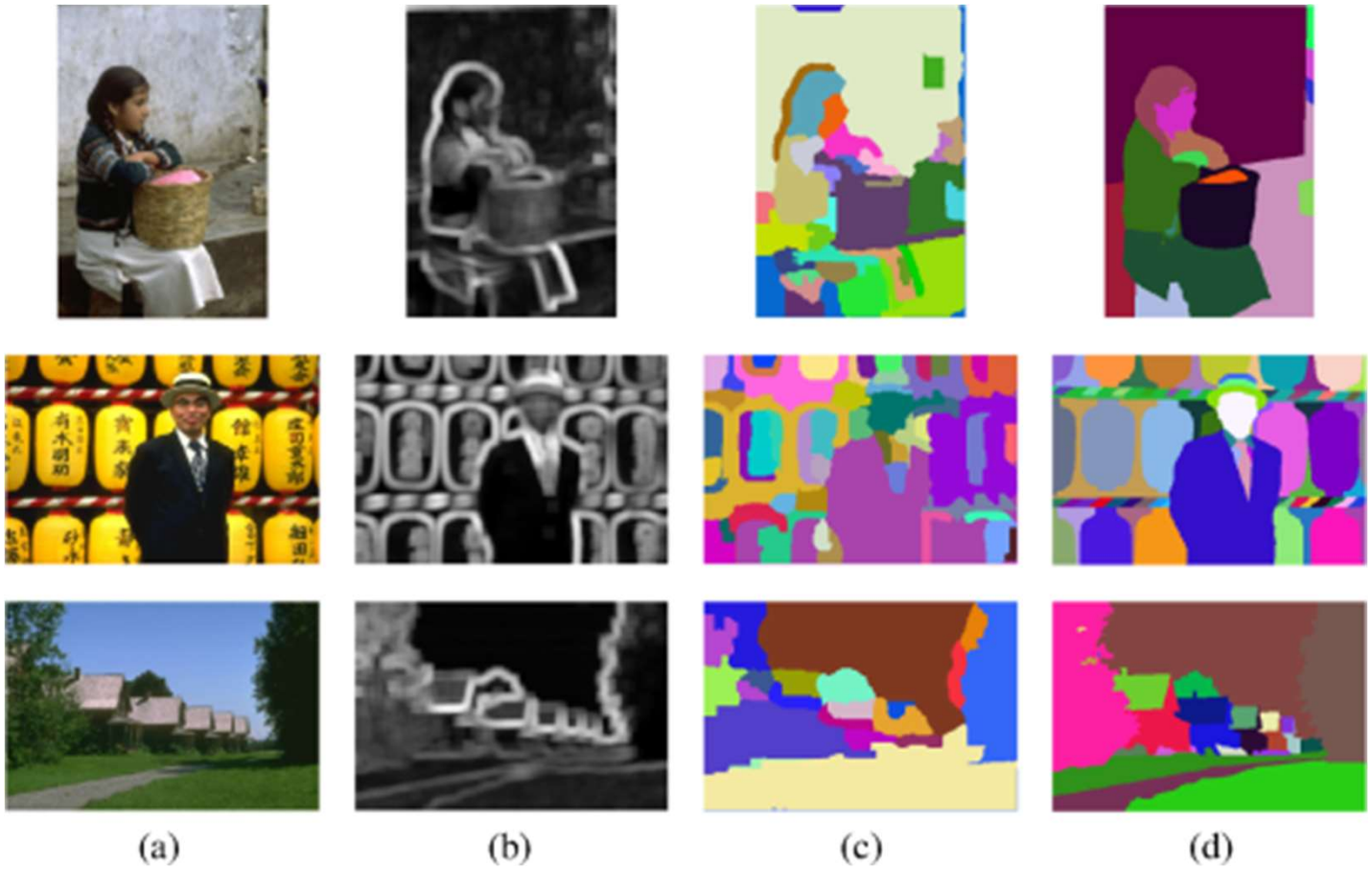
Sumber: Image Segmentation, by Dr. Rajeev Srivastava

Segmentation steps: Original image (a), feature description image (b), image after the splitting process (c), image after the merging process (d), elimination of small regions (e), and the resulting segmentation classes obtained after the region growing procedure (f).



Sumber: *Integral split-and-merge methodology for real-time image segmentation*
By Fernando E. Correa-Tome, Raul E. Sanchez-Yanez

Example results: the input image (a), the SD image (b), the segmentation of the SD image using ISM (c), and a human-made reference segmentation (d).



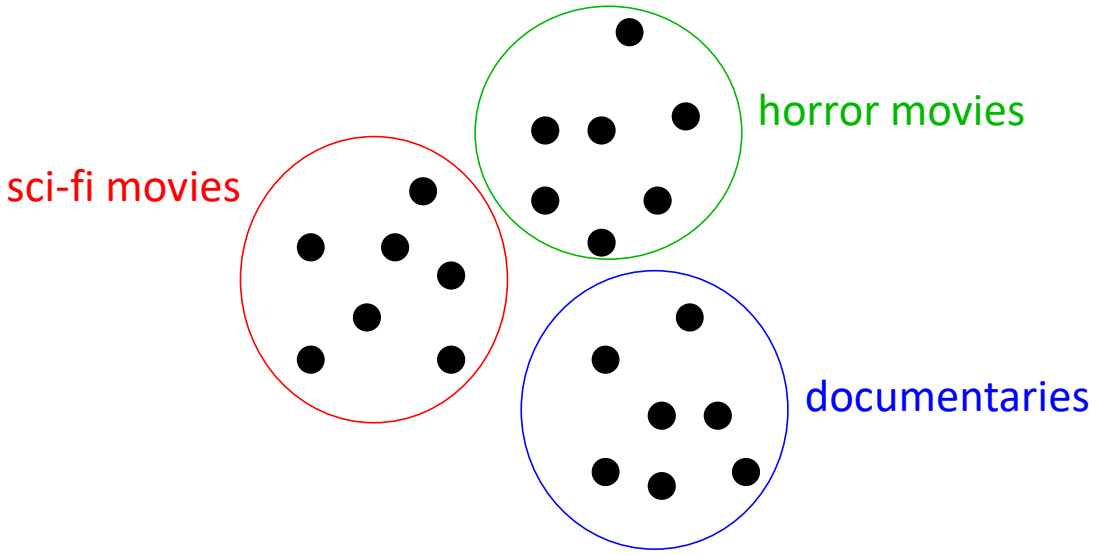
Sumber: *Integral split-and-merge methodology for real-time image segmentation*
By Fernando E. Correa-Tome, Raul E. Sanchez-Yanez

Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

4. Clustering

Prinsip *clustering* secara umum

- Misalkan terdapat N buah titik data (terokan, vektor fitur, dll), $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$
- Kelompokkan (*cluster*) titik-titik yang mirip dalam kelompok yang sama



Bagaimana kaitan *clustering* pada segmentasi citra?

- Nyatakan citra sebagai vektor fitur $\boldsymbol{x}_1,\ldots,\mathbf{x}_n$
  - Sebagai contoh, setiap *pixel* dapat dinyatakan sebagai vektor:
    - Intensitas → menghasilkan vektor dimensi satu
    - Warna → menghasilkan vektor berdimensi tiga (R, G, B)
    - Warna + koordinat, → menghasilkan vektor berdimensi lima

- Kelompokkan vektor-vektor fitur ke dalam ***k*** kluster

Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

citra input

9 4 2	7 3 1	8 6 8
8 2 4	5 8 5	3 7 2
9 4 5	2 9 3	1 4 4

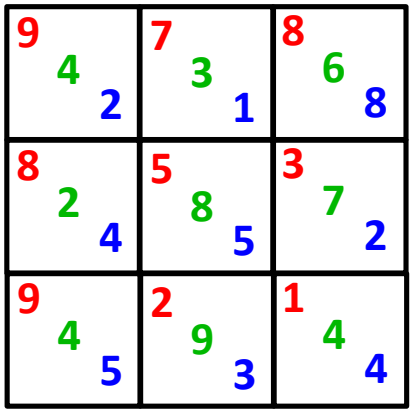
Vektor fitur untuk clustering berdasarkan warna

[9 4 2]	[7 3 1]	[8 6 8]
[8 2 4]	[5 8 5]	[3 7 2]
[9 4 5]	[2 9 3]	[1 4 4]

RGB (or YUV) space clustering

Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

citra input



Vektor fitur untuk clustering berdasarkan warna dan koordinat pixel

[9 4 2 0 0] [7 3 1 0 1] [8 6 8 0 2]

[8 2 4 1 0] [5 8 5 1 1] [3 7 2 1 2]

[9 4 5 2 0] [2 9 3 2 1] [1 4 4 2 2]

RGBXY (or YUVXY) space clustering

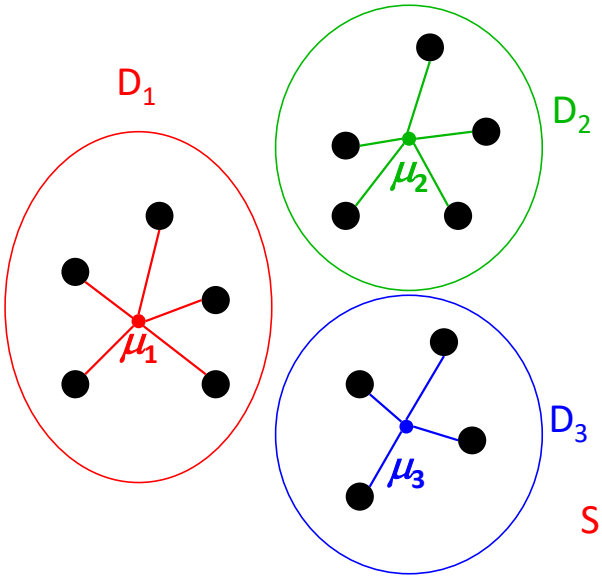
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

K-Means Clustering

- *K-means clustering* merupakan algoritma *clustering* yang paling populer
- Asumsikan jumlah cluster adalah *k*
- Mengoptimalkan (secara hampiran) fungsi objektif berikut untuk variabel D_i dan μ_i

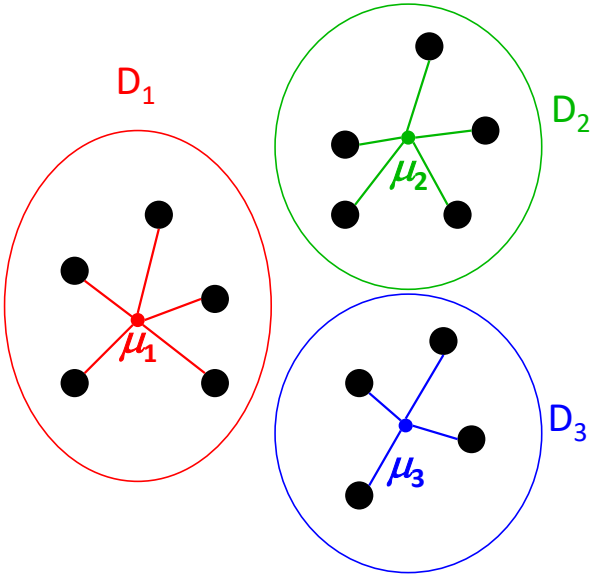
$$E_k = SSE = \sum_{i=1}^{k} \sum_{x \in D_i} \|x - \mu_i\|^2$$

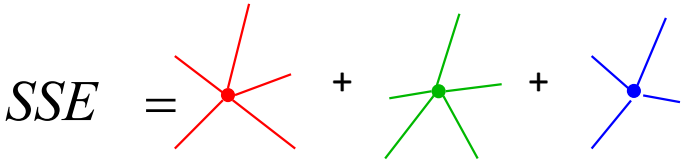
sum of squared errors dari kluster dengan pusat μ_i



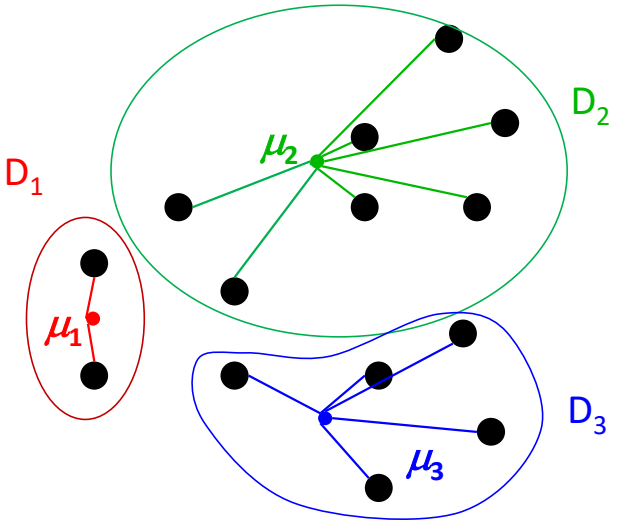
$$SSE = \quad + \quad +$$

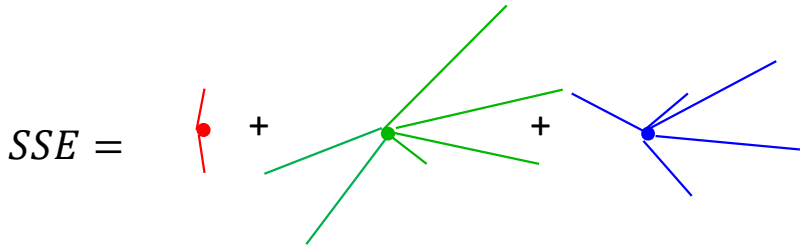
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation





Good (tight) clustering
smaller value of SSE

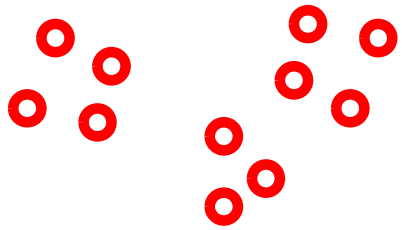




Bad (loose) clustering
larger value of *SSE*

Algoritma K-means Clustering

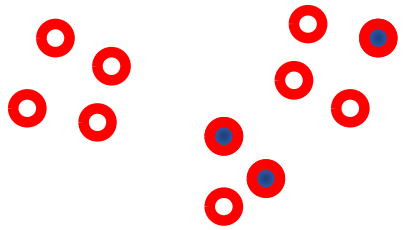
- Initialization step
  1. pick ***k*** cluster centers randomly



Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Algoritma K-means Clustering

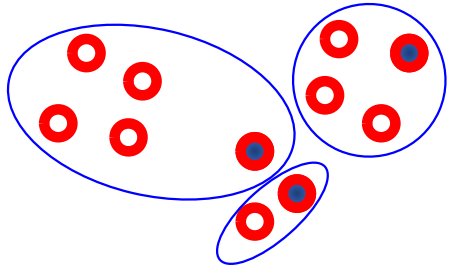
- Initialization step
  1. pick ***k*** cluster centers randomly



Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Algoritma K-means Clustering

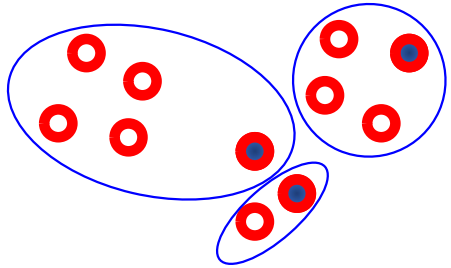
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

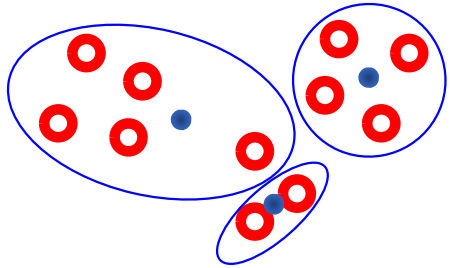
Algoritma K-means Clustering

- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



Algoritma K-means Clustering

- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center

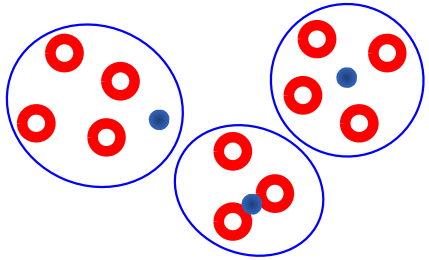


- Iteration steps
  1. compute means in each cluster $\mu_i = \frac{1}{|D_i|} \sum_{x \in D_i} x$

Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Algoritma K-means Clustering

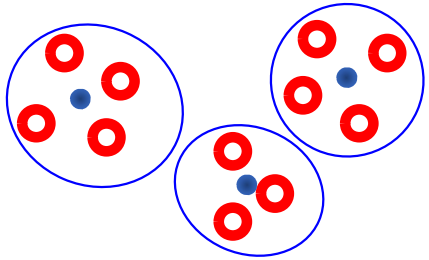
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_i = \frac{1}{|D_i|} \sum_{x \in D_i} x$
  2. re-assign each sample to the closest mean

Algoritma K-means Clustering

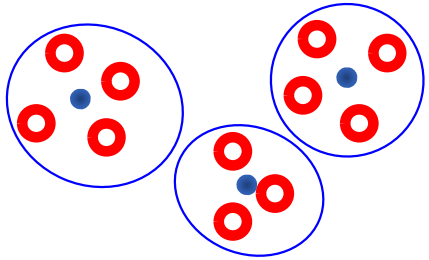
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_i = \frac{1}{|D_i|} \sum_{x \in D_i} x$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

Algoritma K-means Clustering

- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



- Iteration steps
  1. compute means in each cluster $\mu_i = \frac{1}{|D_i|} \sum_{x \in D_i} x$
  2. re-assign each sample to the closest mean
- Iterate until clusters stop changing

- This procedure decreases the value of the objective function

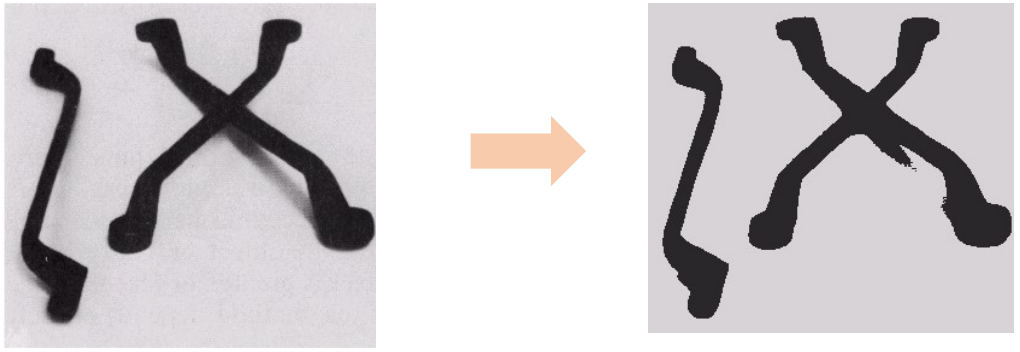
$$E_k(D, \mu) = \sum_{i=1}^{k} \sum_{x \in D_i} \|x - \mu_i\|^2$$

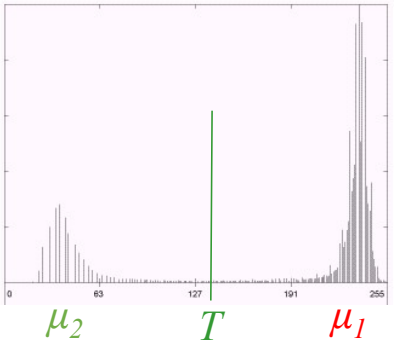
optimization variables

$$D = (D_1, \ldots, D_k)$$
$$\mu = (\mu_1, \ldots, \mu_k)$$

block-coordinate descent: step 1 optimizes μ, step 2 optimizes D

Contoh hasil *K-means clustering*

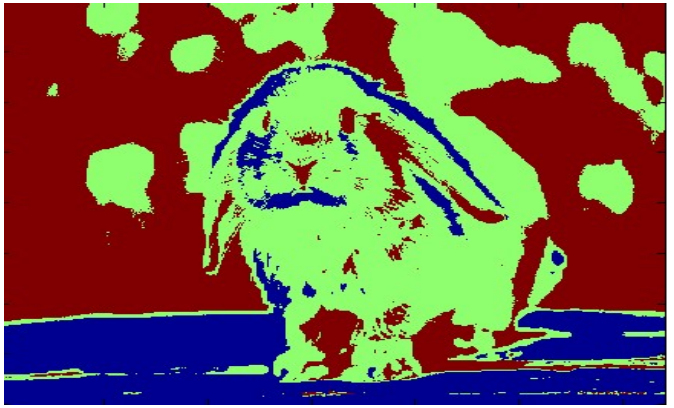




K-means menghasilkan
Pengelompokan yang kompak

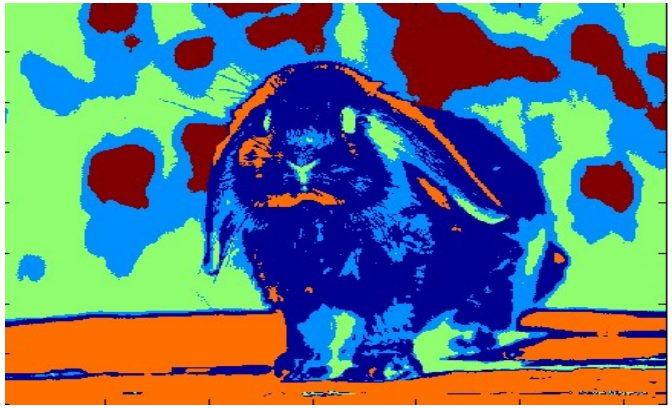
Pada kasus ini, K-means (K=2) secara otomatis menemukan nilai ambang yang bagus antara 2 cluster

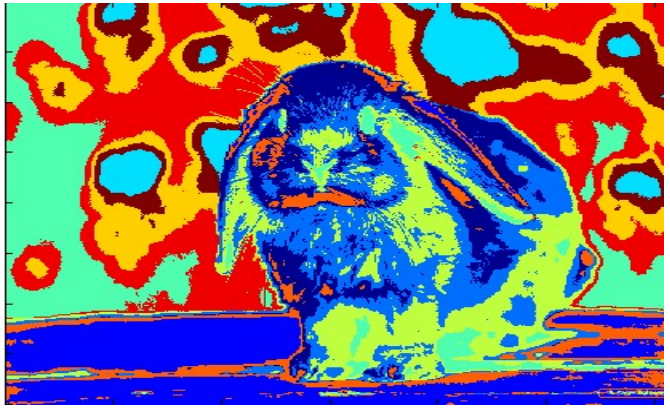


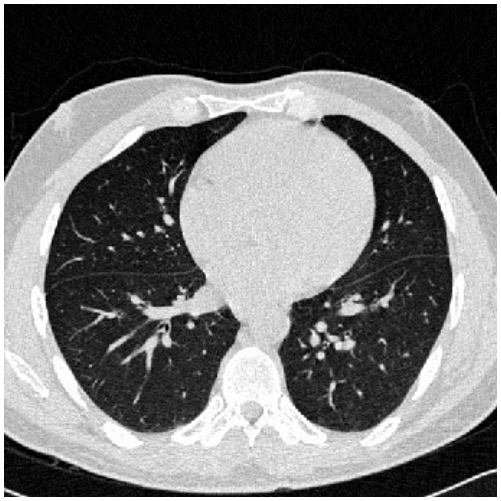


***k* = 3**

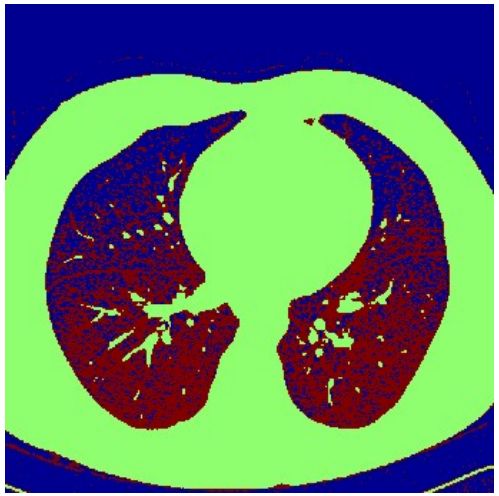
(random colors are used to better show segments/clusters)



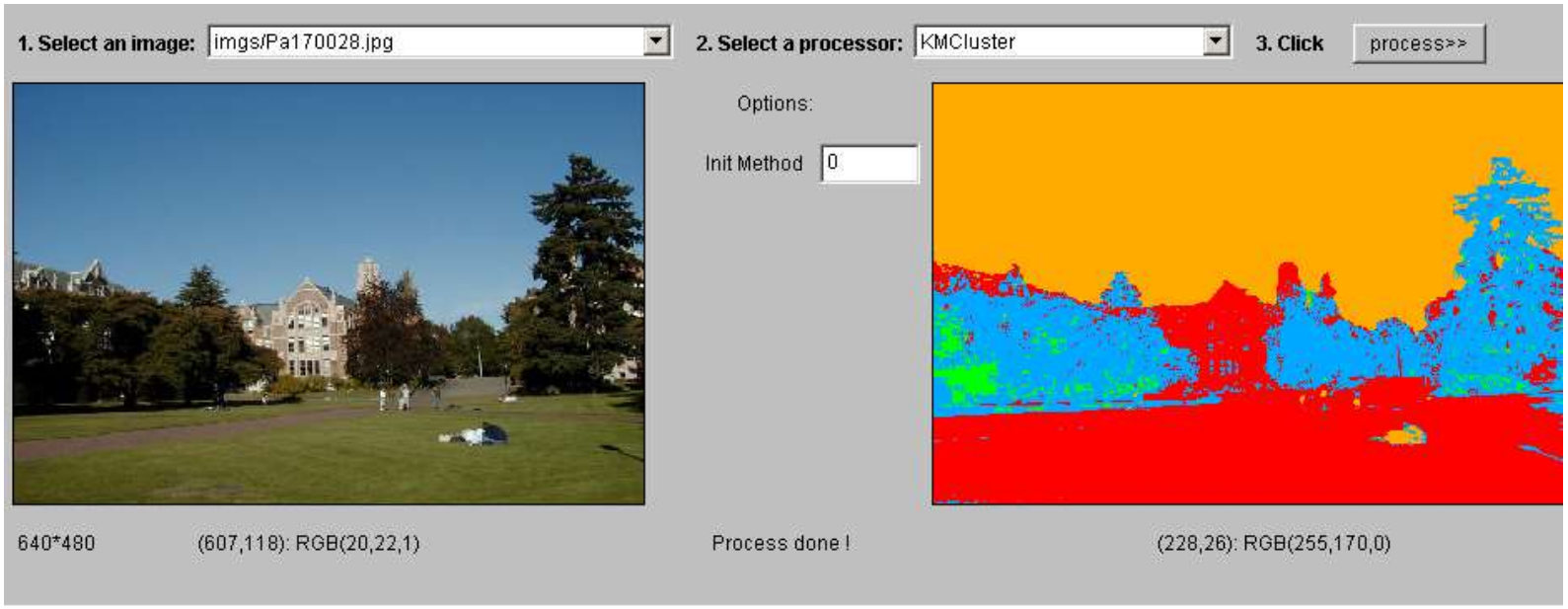


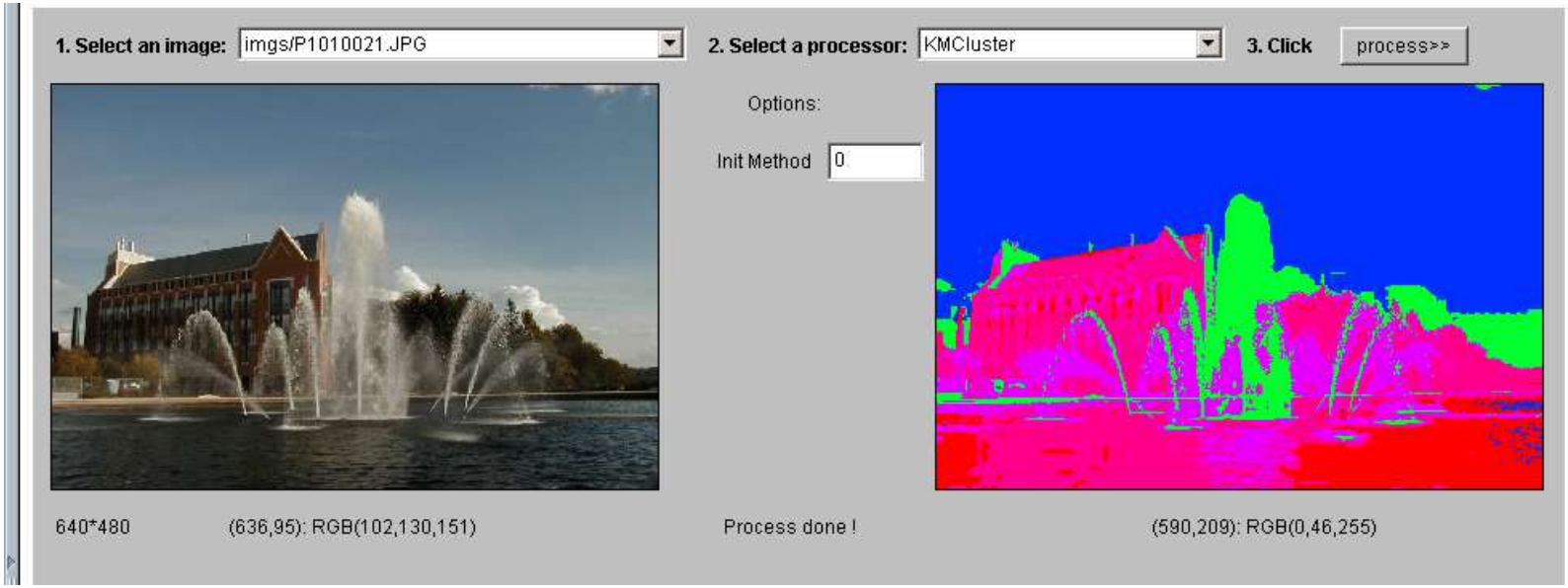


An image(I)

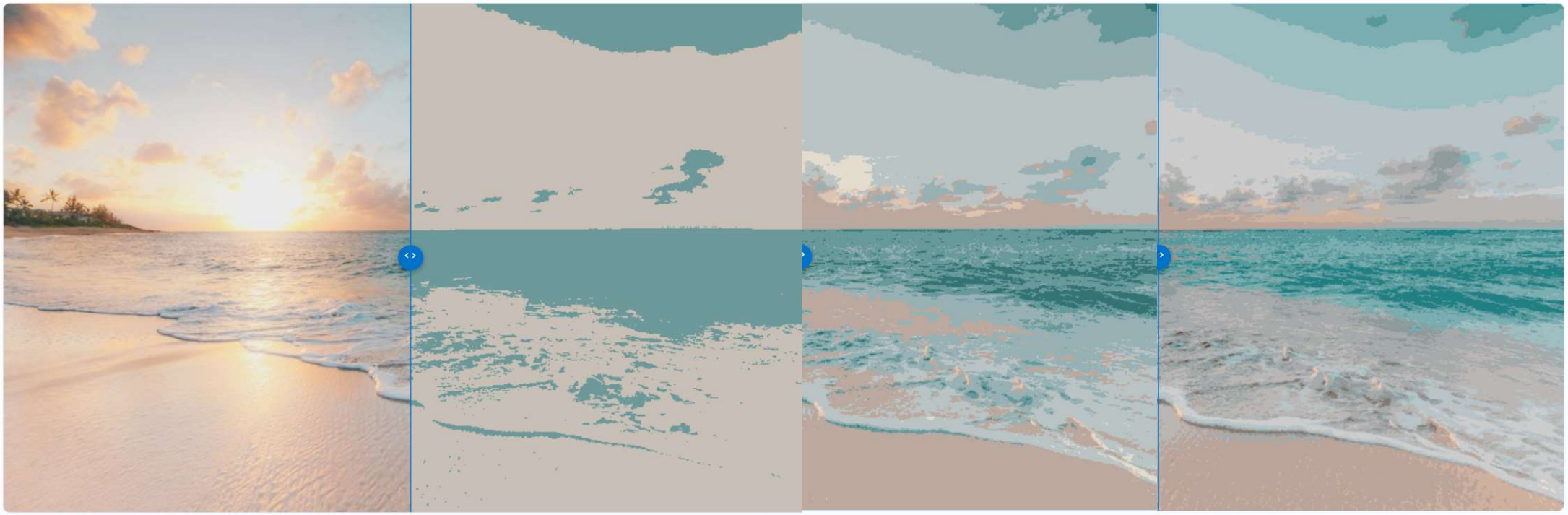


Three cluster image (J)on gray values of I


1. Select an image:
imgs/Pa170028.jpg
2. Select a processor:
KMCluster
3. Click
process>>
Options:
Init Method
0
640*480
(607,118): RGB(20,22,1)
Process done !
(228,26): RGB(255,170,0)


1. Select an image:
imgs/P1010021.JPG
2. Select a processor:
KMCluster
3. Click
process>>
Options:
Init Method
0
640*480
(636,95): RGB(102,130,151)
Process done !
(590,209): RGB(0,46,255)

K = 2 k = 7 k = 16



https://pixlane.media/kmeans-segmentation/

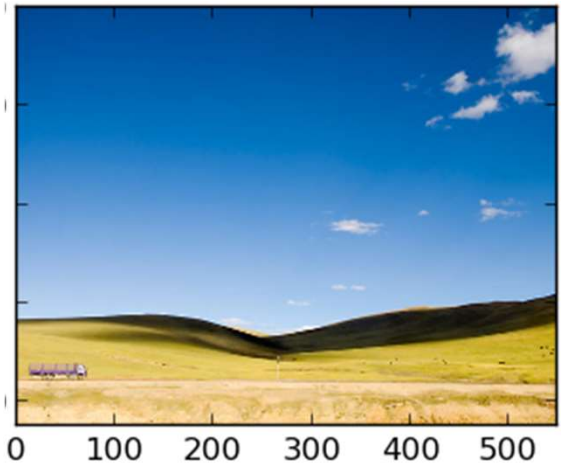


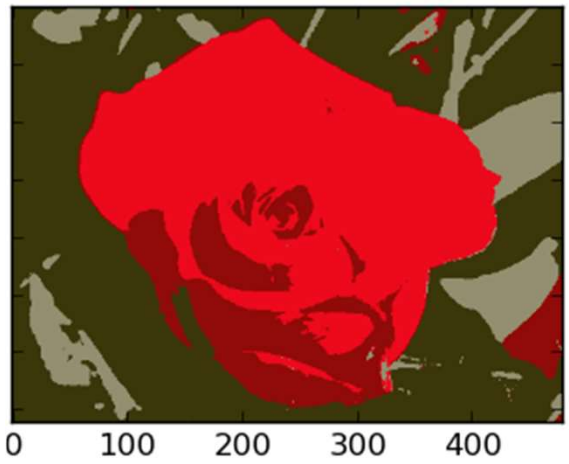
Sumber: https://www.mathworks.com/discovery/image-segmentation.html

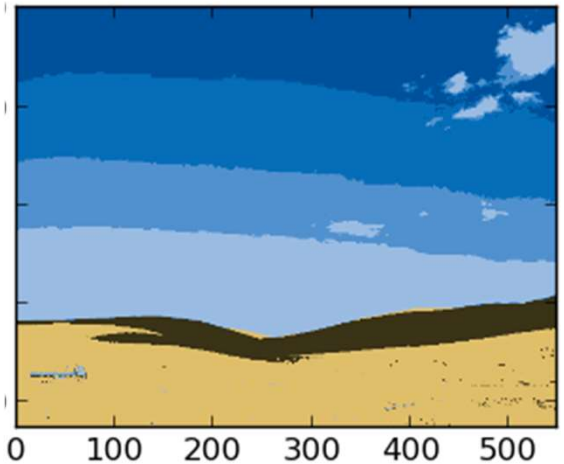
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Contoh hasil *K-means clustering (berdasarkan warna)*









Contoh hasil *K-means clustering (berdasarkan warna + koordinat)*

color quantization



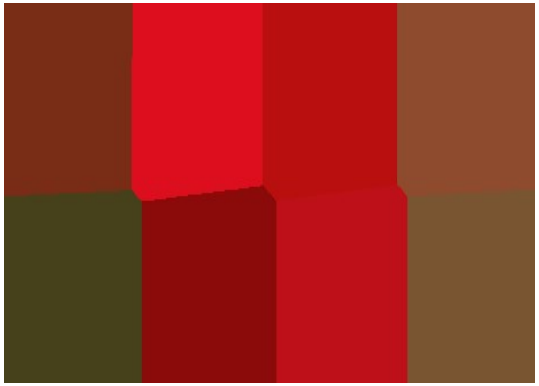
RGB features

superpixels



RGBXY features

Voronoi cells



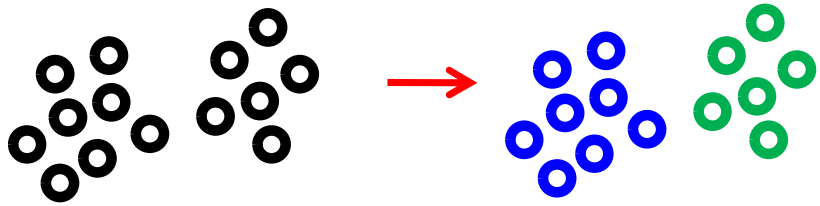
XY features only

Superpixels = perceptually meaningful pixel groups

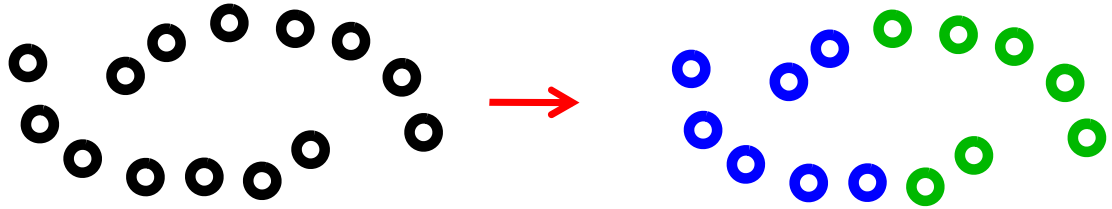
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Sifat-sifat K-means

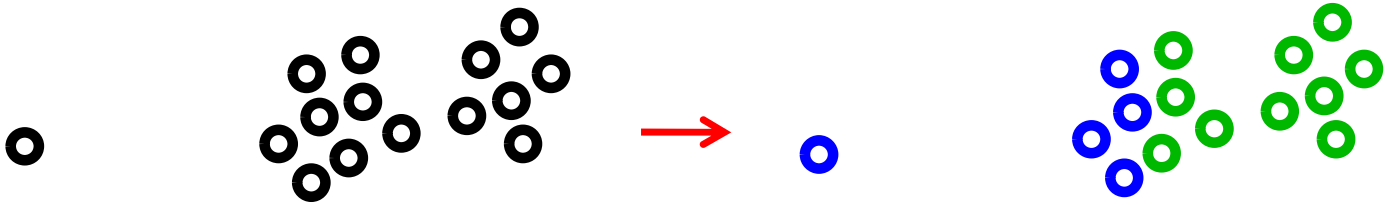
- Works best when clusters are spherical (blob like)



- Fails for elongated clusters
  - SSE is not an appropriate objective function in this case



- Sensitive to outliers



From Geometry to Probability: The Likelihood Connection

- **Moving Beyond Heuristics**: Shifting from a purely geometric view of clustering to a rigorous statistical foundation.
- **Maximum Likelihood Estimation (MLE)**: Finding the optimal cluster means (μ_i) by fitting them to probability distributions.
- **Gaussian Modeling**: Assuming that data points within each cluster follow a normal (Gaussian) distribution.
- **The Mathematical Bridge**: Demonstrating that minimizing the K-Means squared error (E_k) is mathematically equivalent to minimizing the Negative Log-Likelihood.

maximum likelihood (ML) fitting
of parameters μ_i (means) of Gaussian distributions

K-means cost function (SSE)

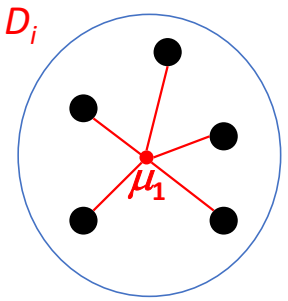
$$E_k = \sum_{i=1}^{k} \sum_{x \in D_i} \|x - \mu_i\|^2$$

equivalent (easy to check)

MLE fitting

$$E_k \sim -\sum_{i=1}^{k} \sum_{x \in D_i} \log P(x \mid \mu_i) + const$$

Gaussian distribution $P(x \mid \mu_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\| x - \mu_i \|^2}{2\sigma^2}\right)$

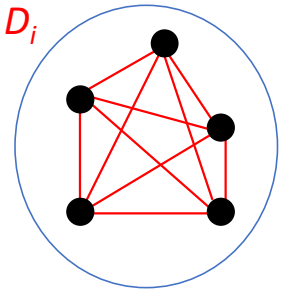


just plug-in expression

$$\mu_i = \frac{1}{|D_i|} \sum_{y \in D_i} y$$

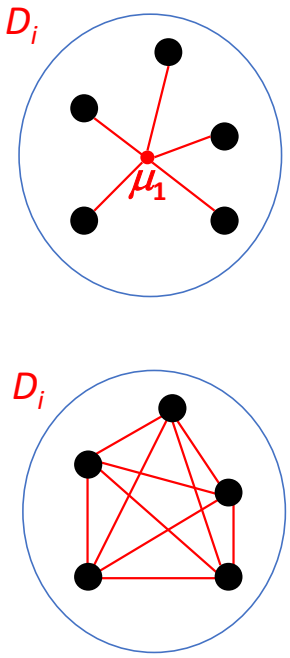
$$E_k = \sum_{i=1}^{k} \sum_{x \in D_i} \| x - \mu_i \|^2$$

equivalent (easy to check)



$$E_k = \sum_{i=1}^{k} \sum_{x, y \in D_i} \frac{\| x - y \|^2}{2 \cdot | D_i |}$$

sample variance: $\text{var}(D_i) = \frac{1}{|D_i|} \sum_{x \in D_i} \| x - \mu_i \|^2 = \frac{1}{2|D_i|^2} \sum_{x, y \in D_i} \| x - y \|^2$



both formulas can be written as

$$E_k = \sum_{i=1}^{k} |D_i| \cdot \mathrm{var}(D_i)$$

sample variance: $\mathrm{var}(D_i) = \frac{1}{|D_i|} \sum_{x \in D_i} \| x - \mu_i \|^2 = \frac{1}{2|D_i|^2} \sum_{x,y \in D_i} \| x - y \|^2$

Rangkuman K-means

- Advantages
  - Principled (objective function) approach to clustering
  - Simple to implement (the approximate iterative optimization)
  - Fast
- Disadvantages
  - Only a local minimum is found (sensitive to initialization)
  - May fail for non-blob like clusters
  - Sensitive to outliers
  - Sensitive to choice of ***k***

K-means fits Gaussian models

Quadratic errors are such

Can add sparsity term and make k an additional variable

$$E = \sum_{i=1}^{k} \sum_{x \in D_i} \left\| x - \mu_i \right\|^2 + \gamma \cdot | k |$$

Akaike Information Criterion (AIC) or
Bayesian Information Criterion (BIC)

https://www.youtube.com/watch?app=desktop&v=pw6sTn55fZY

Program Matlab untuk image segmentation dengan K-means

- Fungsi imsegkmeans hanya tersedia untuk Matlab R2022a

```
I = imread('camera.bmp');
imshow(I)
title('Original Image');
[L,Centers] = imsegkmeans(I,3); % Segmentasi citra menjadi tiga
label dengan K-means clustering
B = labeloverlay(I,L);
imshow(B)
title('Labeled Image')
```

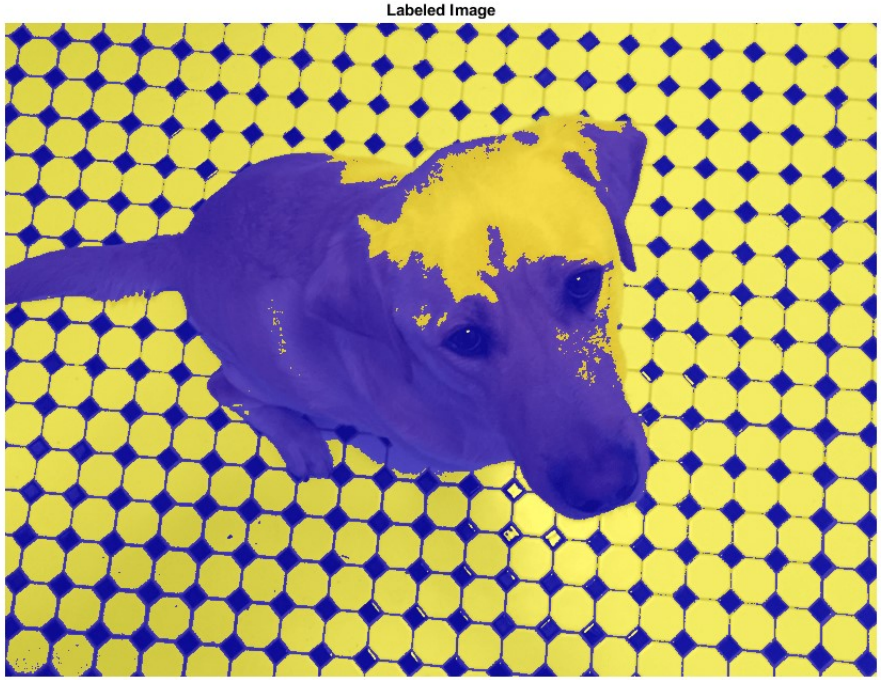
Original Image



Labeled Image



```
RGB = imread("kobi.png");
RGB = imresize(RGB,0.5);
imshow(RGB)
L = imsegkmeans(RGB,2);
B = labeloverlay(RGB,L);
imshow(B)
title("Labeled Image")
```



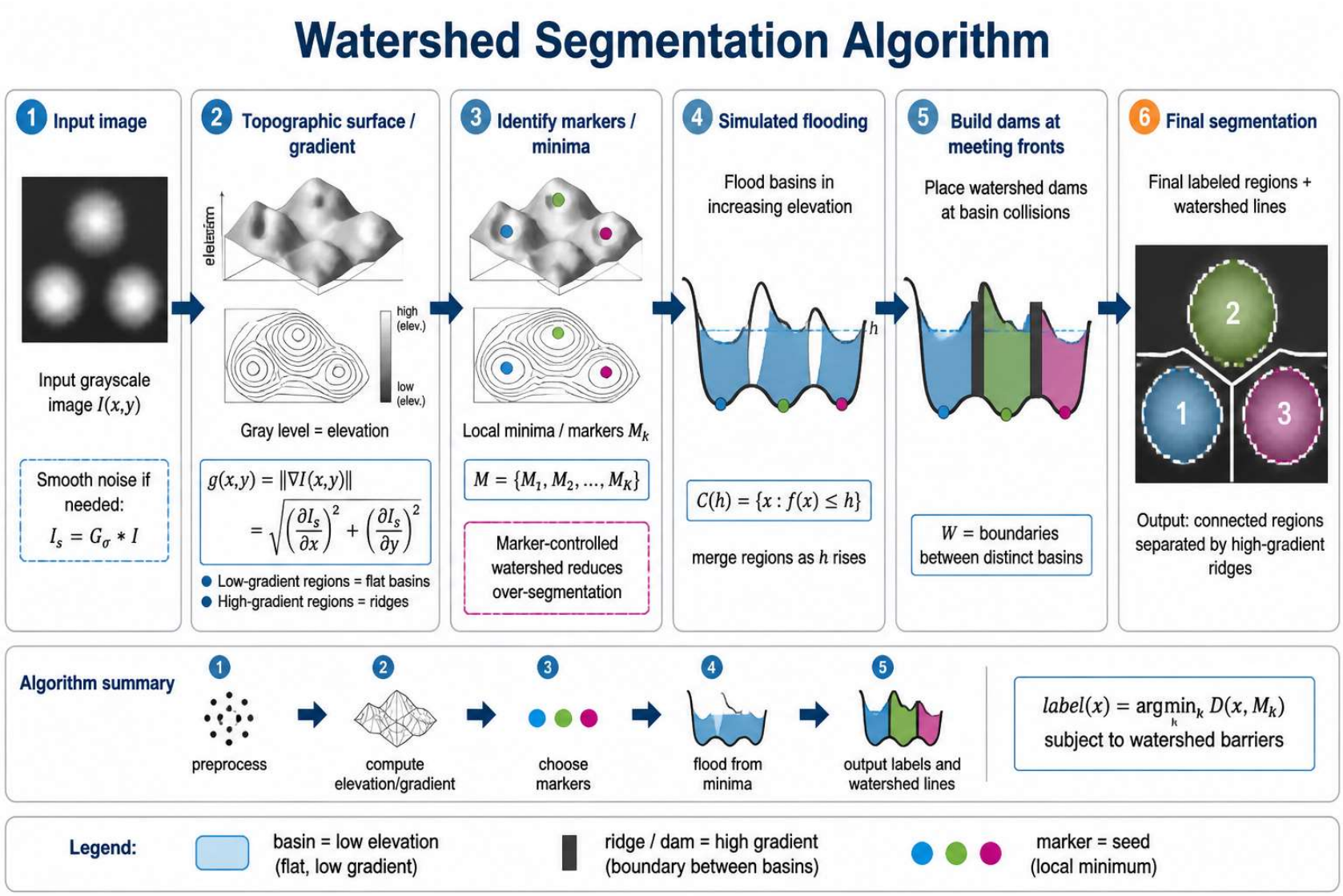


Watershed Segmentation

- **Concept:** Treats an image like a topographic map where pixel intensities represent elevation (low values are valleys or catchment basins, high values are peaks). As "water" floods the valleys from local minima, dams are built where waters meet to form segmentation boundaries. [1, 2, 3]
- **Strengths:** Fast, computationally simple, and effective at separating touching or overlapping objects. [1, 2, 3]
- **Weaknesses:** Prone to **over-segmentation** caused by image noise or minor intensity variations, requiring markers or pre-processing (like distance transforms) to control. [1, 2, 3]



https://opencv.org/watershed-segmentation-using-opencv/
https://pixlane.media/watershed-segmentation/



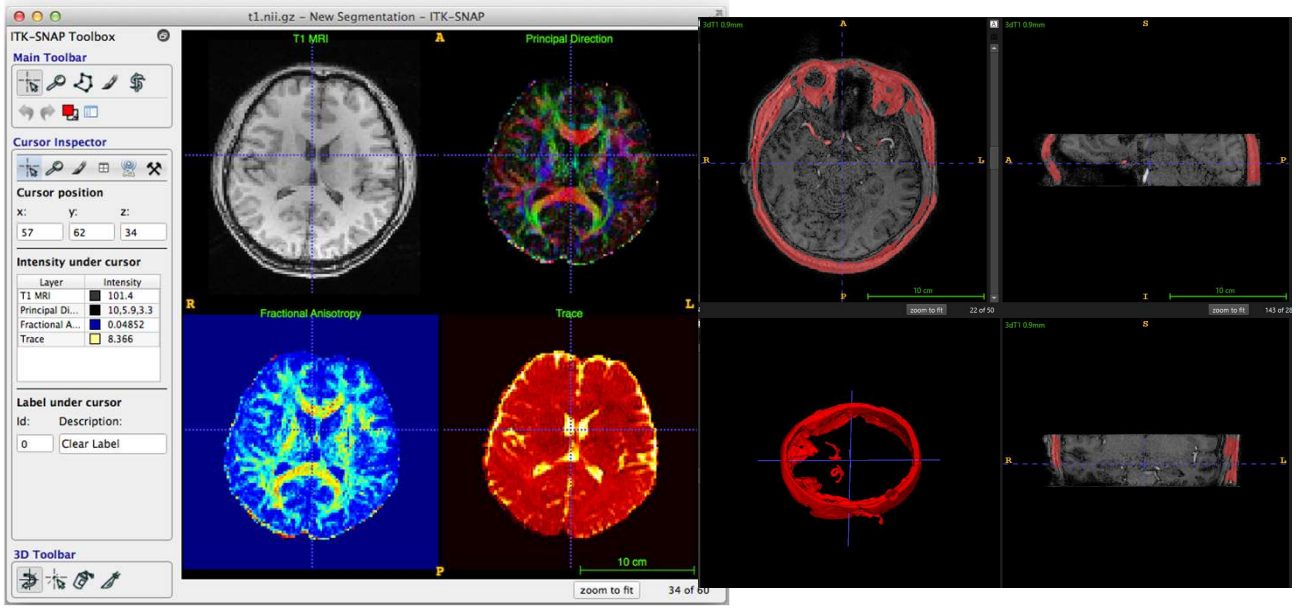
Generated by Scispace

Active Contour Segmentation (Snakes)

- **Concept:** Uses an evolving curve (a "snake") that deforms iteratively within an image by minimizing an internal energy (smoothness, stiffness) and external energy (image gradients, edges) to snap onto object boundaries. [1, 2]

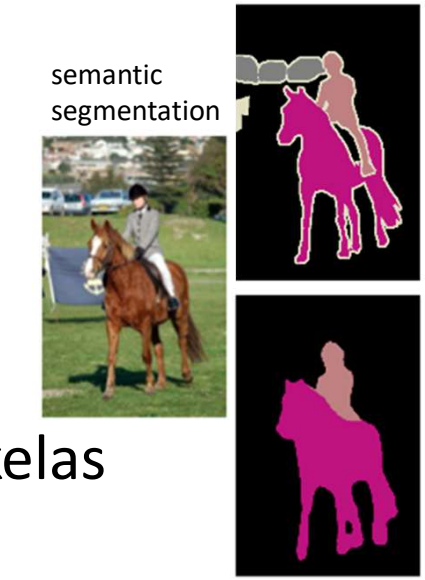
- **Strengths:** Produces smooth, continuous closed contours and handles generalisation well. [1, 2]

- **Weaknesses:** Sensitive to initial positioning (poor capture range; requires starting close to the target boundary) and prone to getting stuck in local minima. [1, 2, 3]

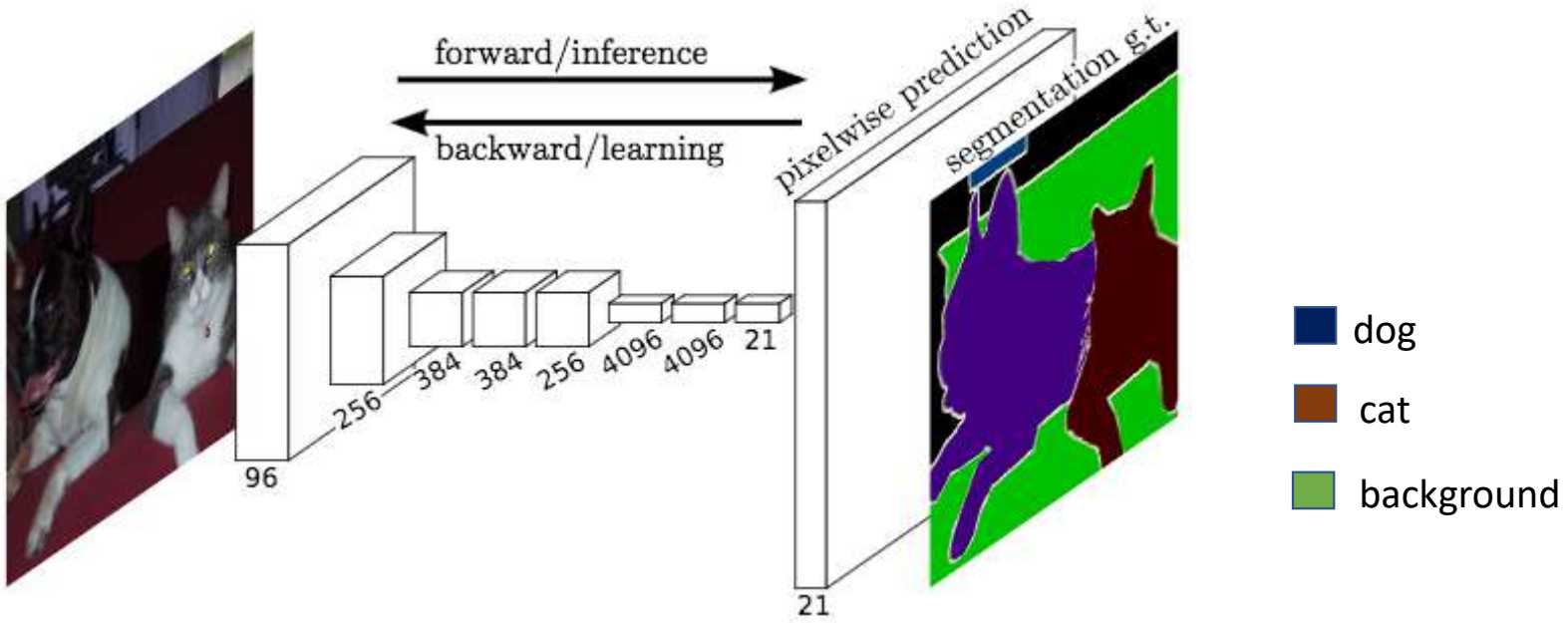


https://www.itksnap.org/pmwiki/pmwiki.php

Segmentasi Citra dengan *Deep Learning*



- Disebut juga *semantic segmentation*
- Tiap *pixel* di dalam citra diasosiasikan dengan sebuah label kelas



Segmentasi Semantik

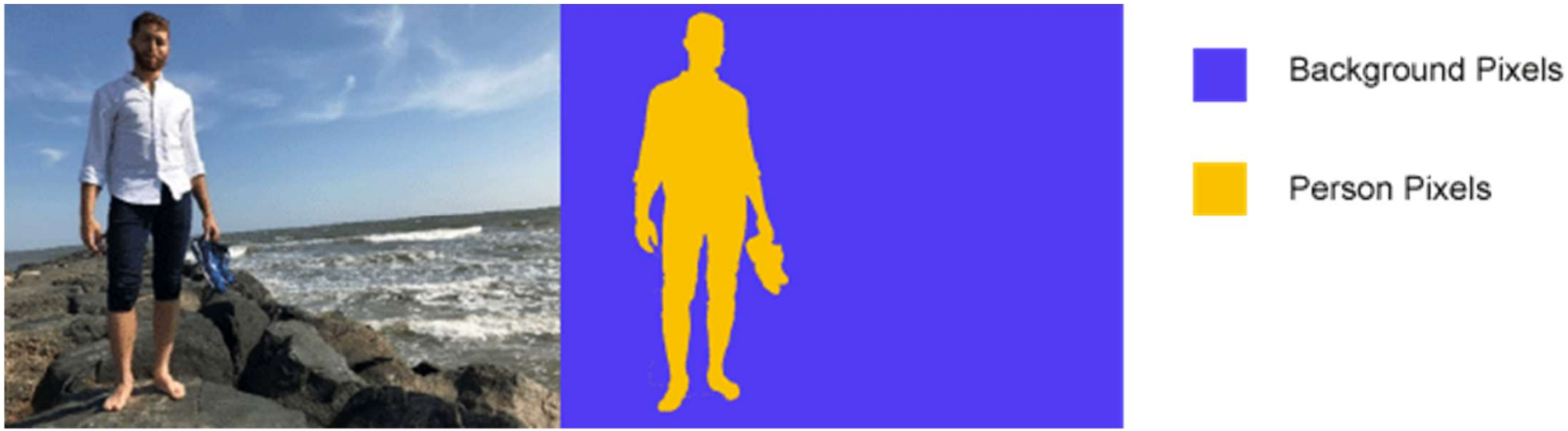
- Segmentasi semantik adalah algoritma pembelajaran mendalam yang mengaitkan label atau kategori dengan setiap piksel dalam sebuah gambar.

- Ini digunakan untuk mengenali kumpulan piksel yang membentuk kategori berbeda. Misalnya, kendaraan otonom perlu mengidentifikasi kendaraan, pejalan kaki, rambu lalu lintas, trotoar, dan fitur jalan lainnya.

- Segmentasi semantik digunakan dalam banyak aplikasi seperti mengemudi otomatis, pencitraan medis, dan inspeksi industri.

Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

- Contoh sederhana dari segmentasi semantik adalah memisahkan gambar menjadi dua kelas. Misalnya, pada Gambar 1, gambar yang memperlihatkan seseorang di pantai dipasangkan dengan versi yang menunjukkan piksel gambar yang disegmentasi menjadi dua kelas terpisah: orang dan latar belakang.



Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html





person
grass
trees
motorbike
road

- Karena segmentasi semantik memberi label pada piksel dalam suatu gambar, maka segmentasi ini lebih tepat dibandingkan bentuk deteksi objek lainnya.

- Hal ini membuat segmentasi semantik berguna untuk aplikasi di berbagai industri yang memerlukan peta gambar yang tepat, seperti:

  a) Mengemudi otonom—untuk mengidentifikasi jalur yang dapat dilalui mobil dengan memisahkan jalan dari rintangan seperti pejalan kaki, trotoar, tiang, dan mobil lain

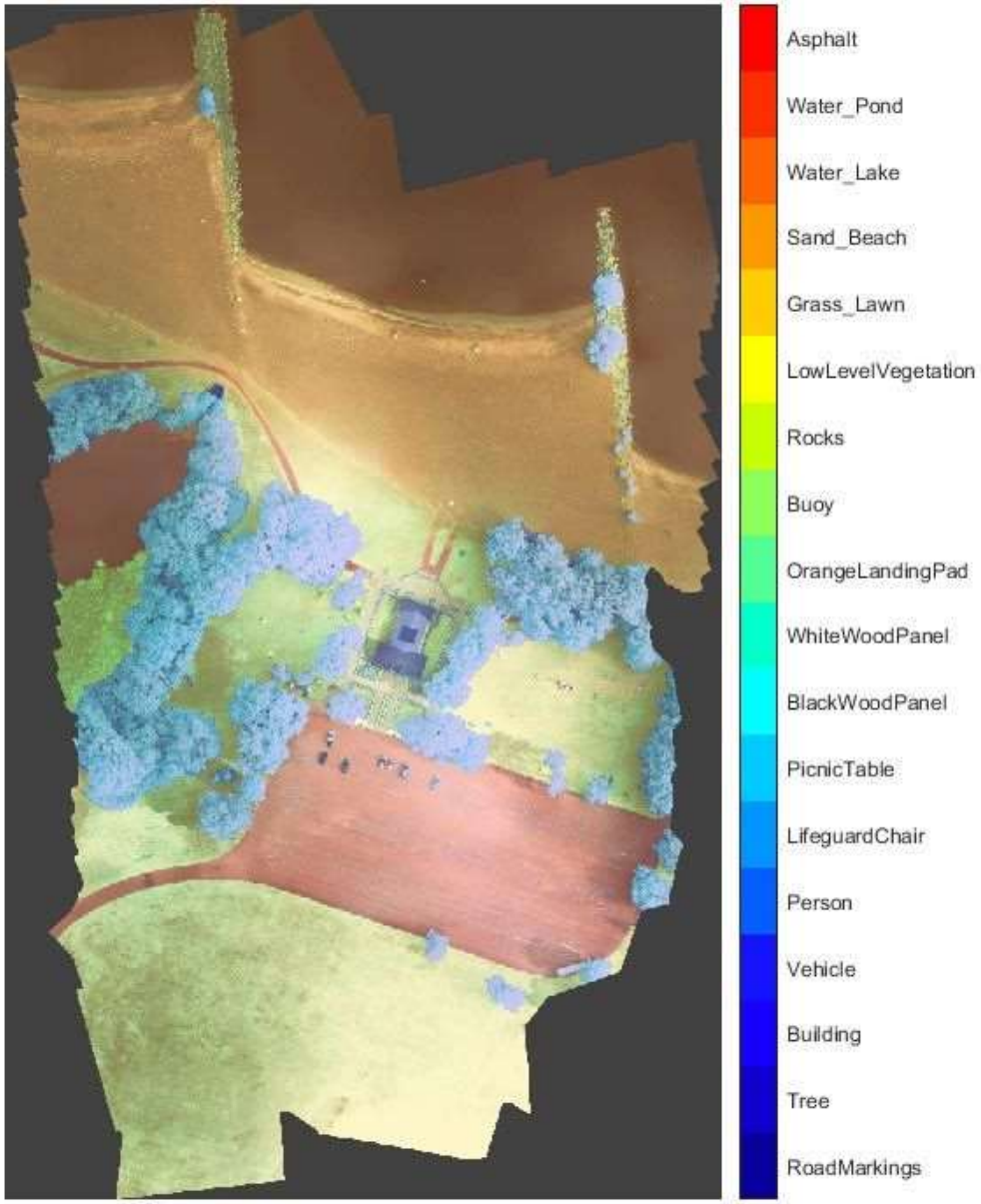
  b) Inspeksi industri—untuk mendeteksi cacat pada bahan, seperti inspeksi wafer

  c) Citra satelit—untuk mengidentifikasi gunung, sungai, gurun, dan medan lainnya

  d) Pencitraan medis—untuk menganalisis dan mendeteksi anomali kanker dalam sel Visi robotik—untuk mengidentifikasi dan menavigasi objek dan medan

Sumebr: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

Segmentasi semantik dari citra satelit multispektral.





Gambar Segmentasi semantik untuk aplikasi mengemudi otomatis.

Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

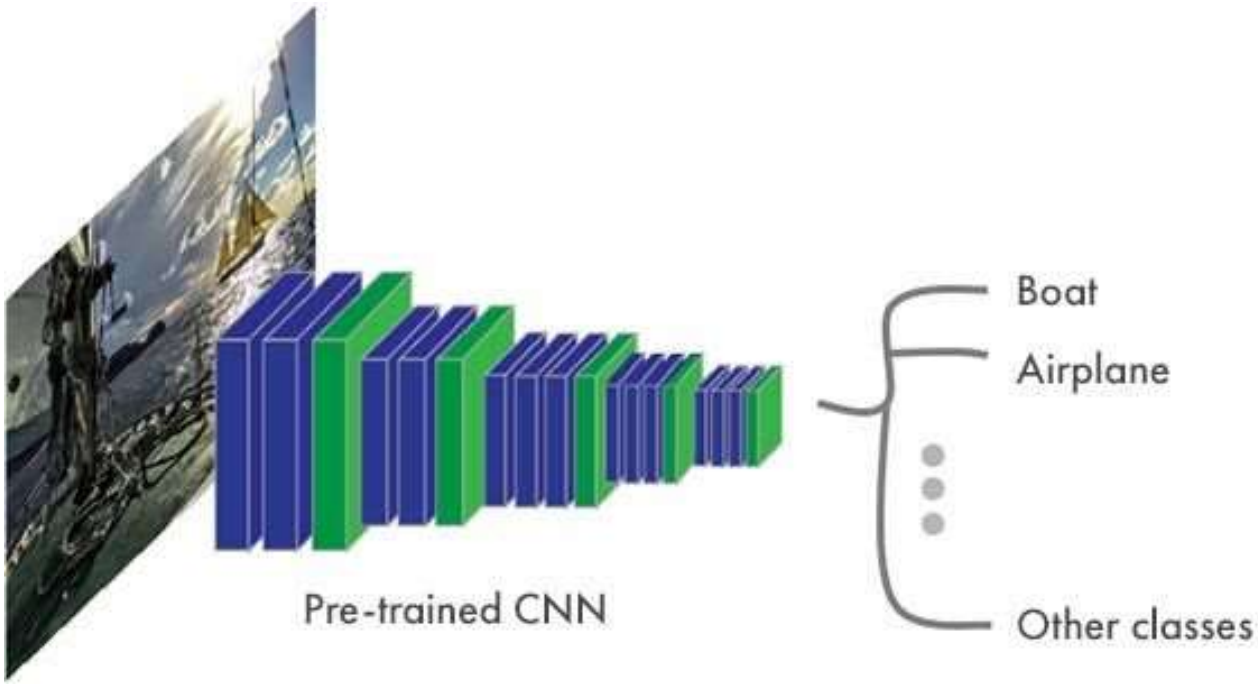
Cara Kerja Segmentasi Semantik

Proses pelatihan jaringan segmentasi semantik untuk mengklasifikasikan gambar mengikuti langkah-langkah berikut:

1. Analisis kumpulan gambar berlabel piksel.
2. Buat jaringan segmentasi semantik.
3. Latih jaringan untuk mengklasifikasikan gambar ke dalam kategori piksel.
4. Menilai keakuratan jaringan.

Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

- Segmentasi semantic menggunakan CNN



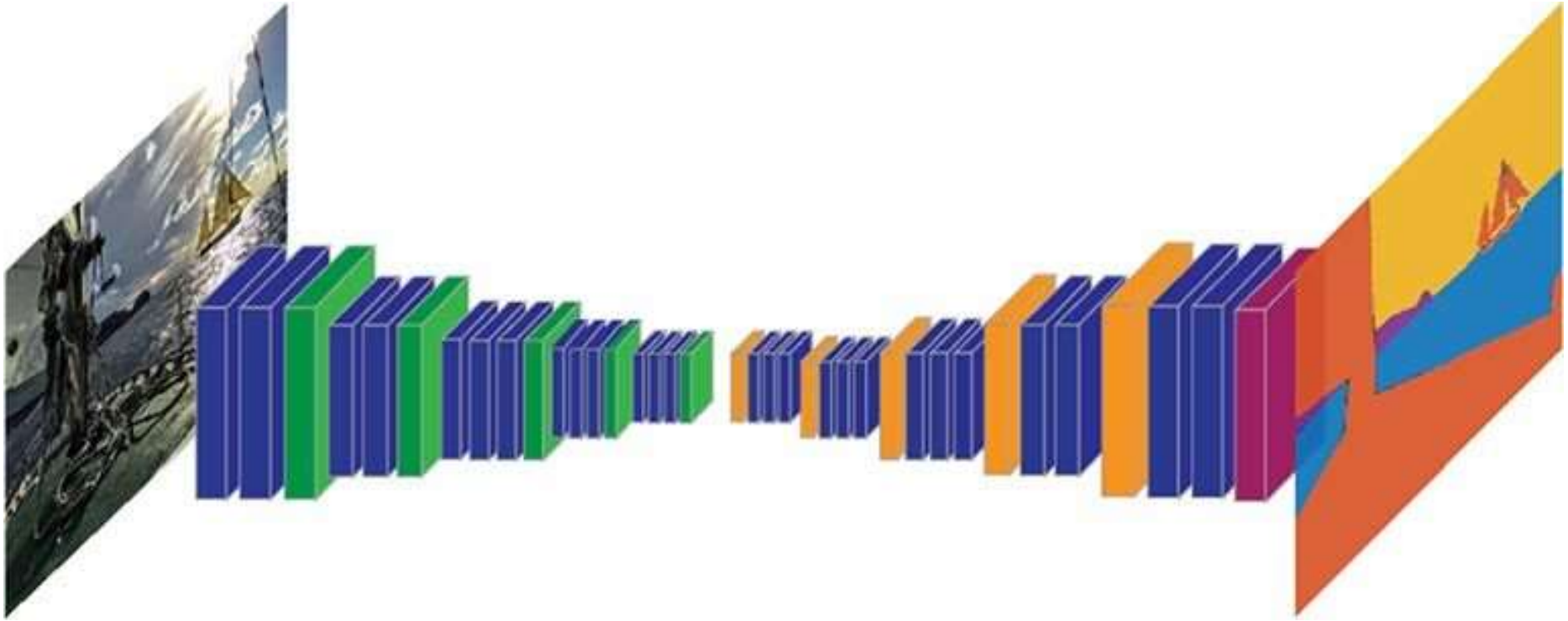
Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

- Untuk mengklasifikasikan pada tingkat piksel, bukan keseluruhan gambar, kita dapat menambahkan implementasi CNN terbalik.

- Proses upsampling dilakukan dengan jumlah yang sama dengan proses downsampling untuk memastikan gambar akhir berukuran sama dengan gambar masukan.

- Terakhir, lapisan keluaran klasifikasi piksel digunakan, yang memetakan setiap piksel ke kelas tertentu. Ini membentuk arsitektur encoder-decoder, yang memungkinkan segmentasi semantik.

Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html



Gambar 6: CNN menjalankan fungsi terkait gambar di setiap lapisan dan kemudian melakukan downsampling gambar menggunakan lapisan penyatuan (hijau). Proses ini diulangi beberapa kali untuk paruh pertama jaringan. Output dari paruh pertama diagram ini diikuti oleh lapisan unpooling dalam jumlah yang sama (oranye).

Sumber: https://www.mathworks.com/solutions/image-video-processing/semantic-segmentation.html