23 - Segmentasi Citra (Bagian 2)

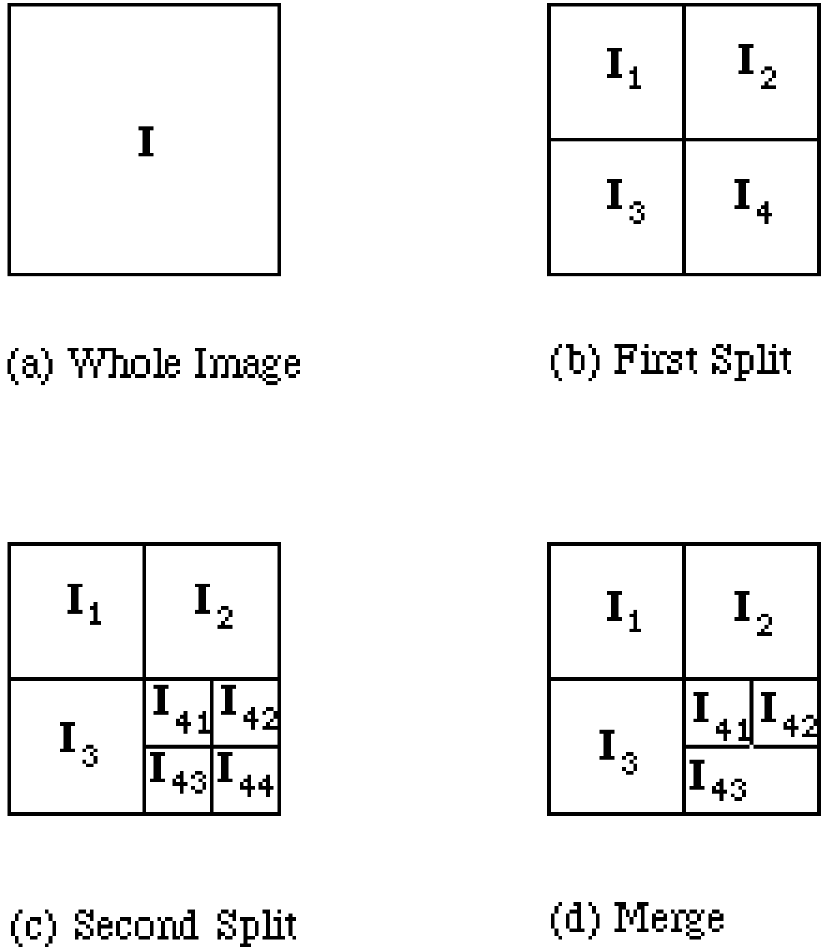
IF4073 Interpretasi dan Pengolahan Citra

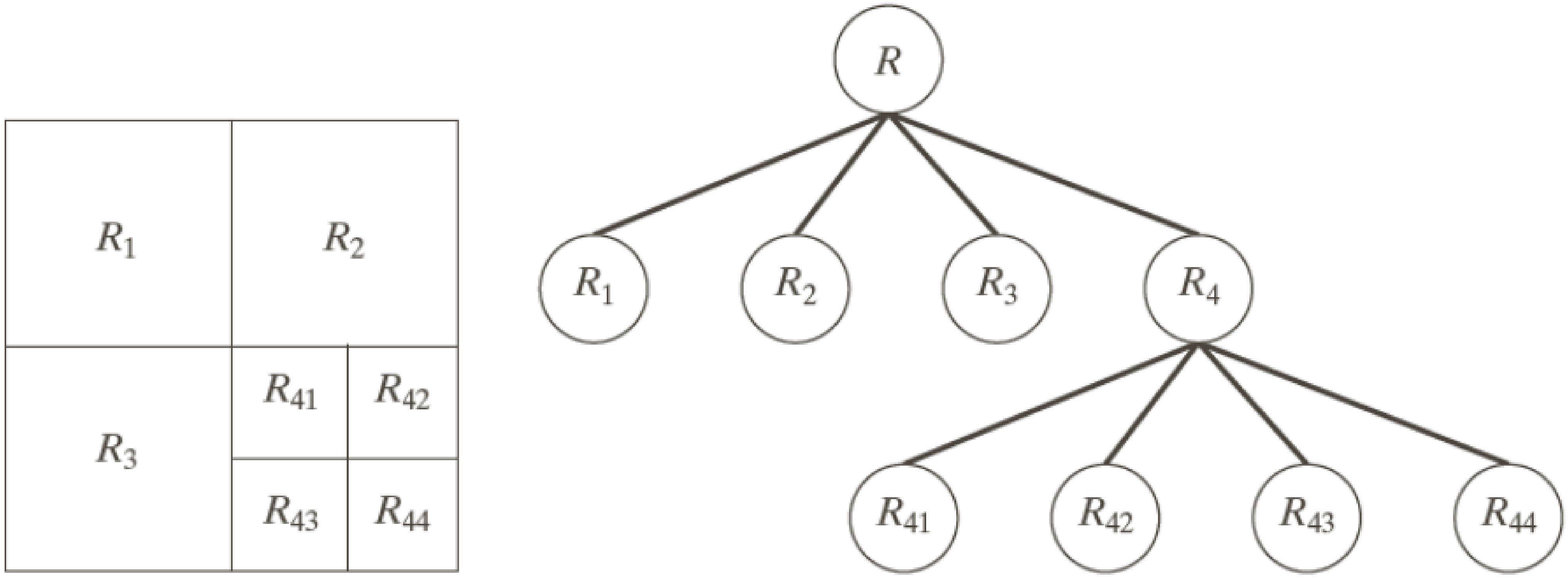
Oleh: Rinaldi Munir

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

3. Split and Merge

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



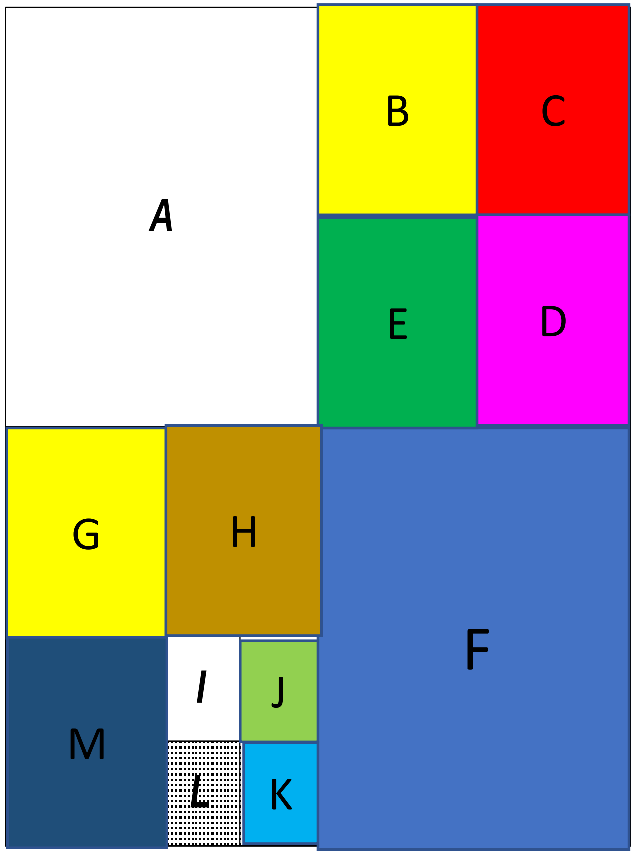

R_1
R_2
R_3
R_{41}
R_{42}
R_{43}
R_{44}
R
R_1
R_2
R_3
R_4
R_{41}
R_{42}
R_{43}
R_{44}

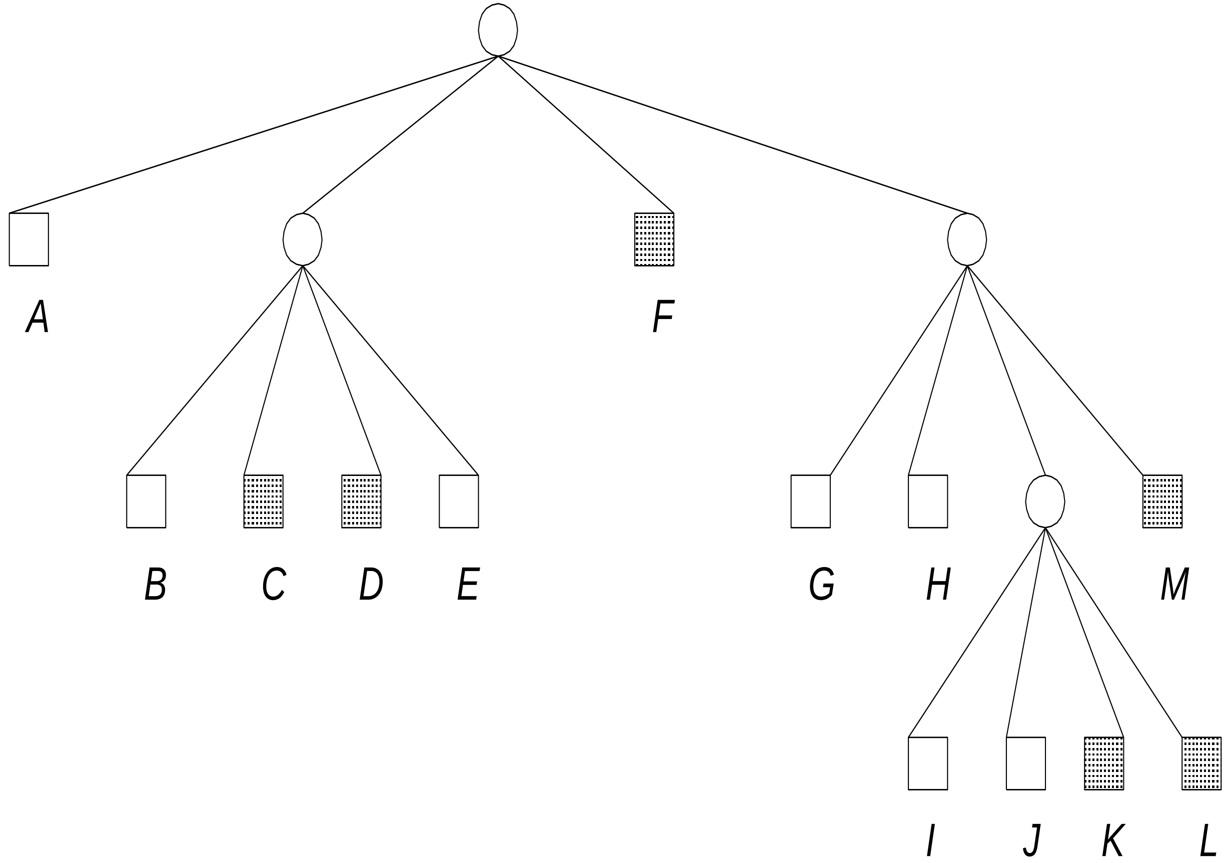
Algoritma *Split & Merge*

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

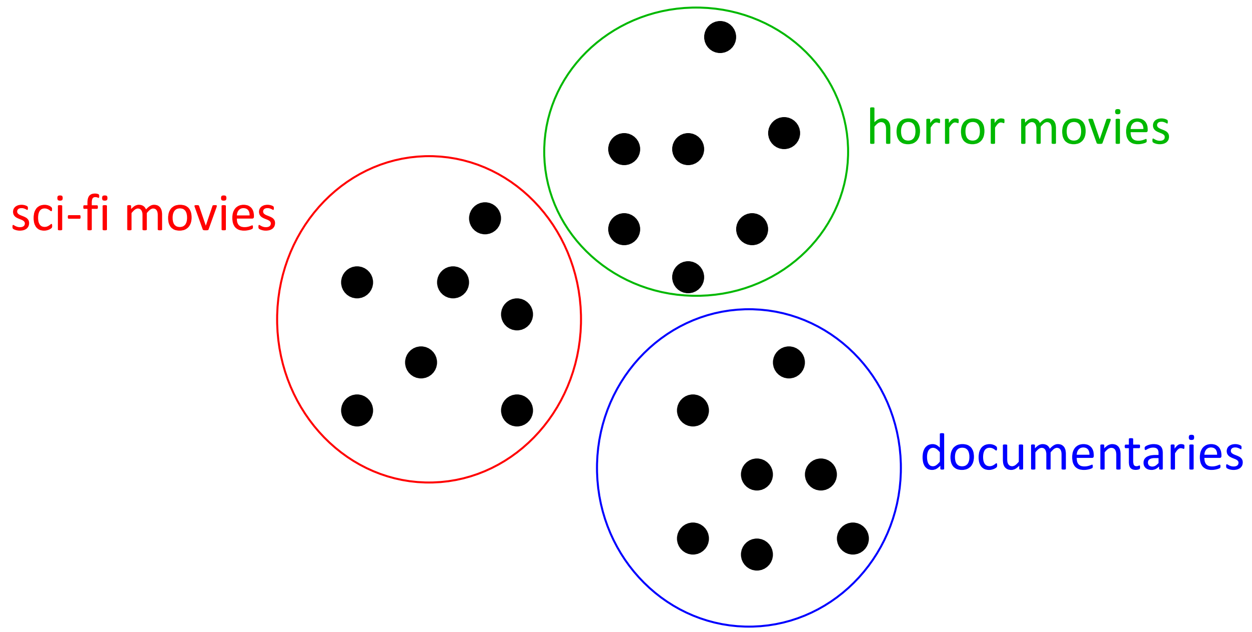

A
F
B
C
D
E
G
H
M
I
J
K
L

Sumber: Image Segmentation, by Dr. Rajeev Srivastava

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 $\boldsymbol{x}_1,\dots,\mathbf{x}_n$
  - 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

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 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

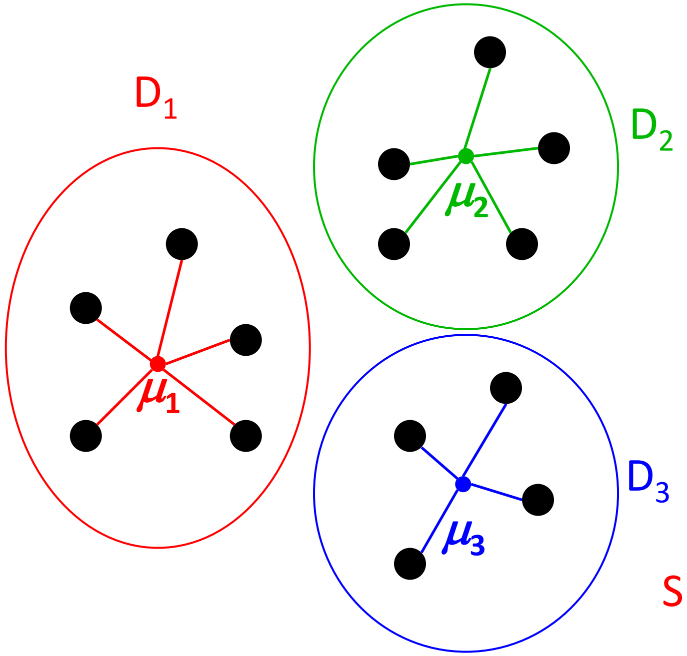
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

K-Means Clustering

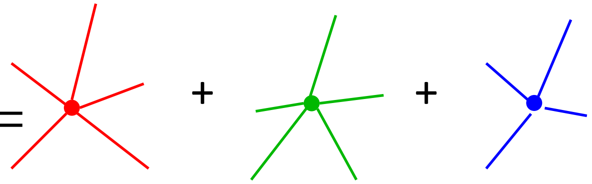
- *K-means clustering* merupakan algoritma *clustering* yang paling populer
- Asumsikan jumlah cluster adalah *k*
- Mengoptimalkan (secara hampiran) fungsi objektif berikut untuk variabel D_i dan μ_i

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

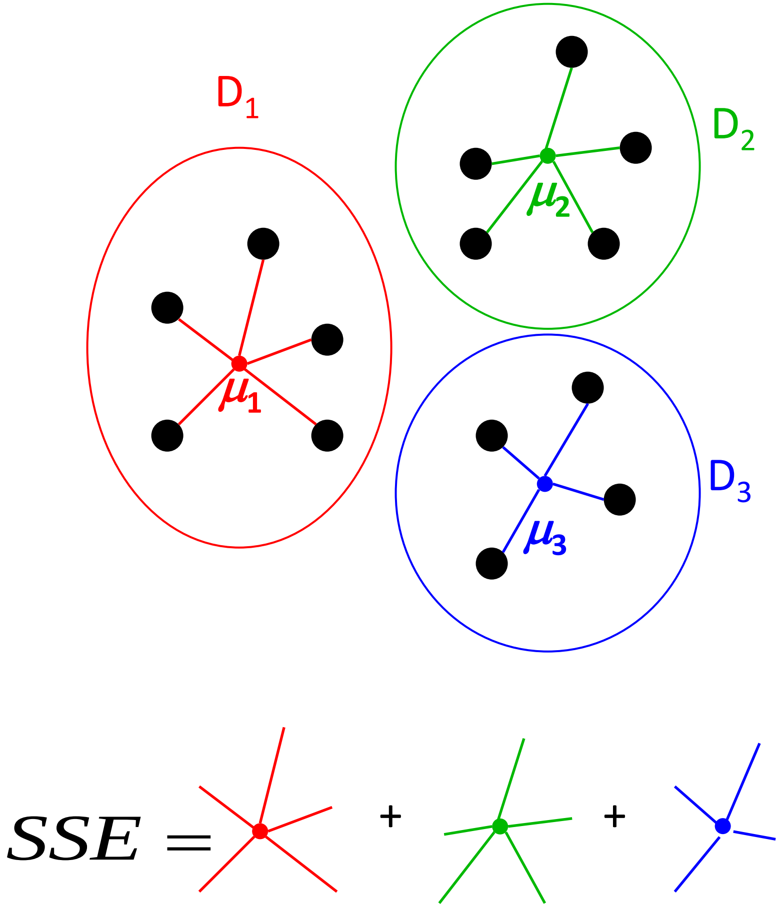
sum of squared errors dari kluster dengan pusat μ_i



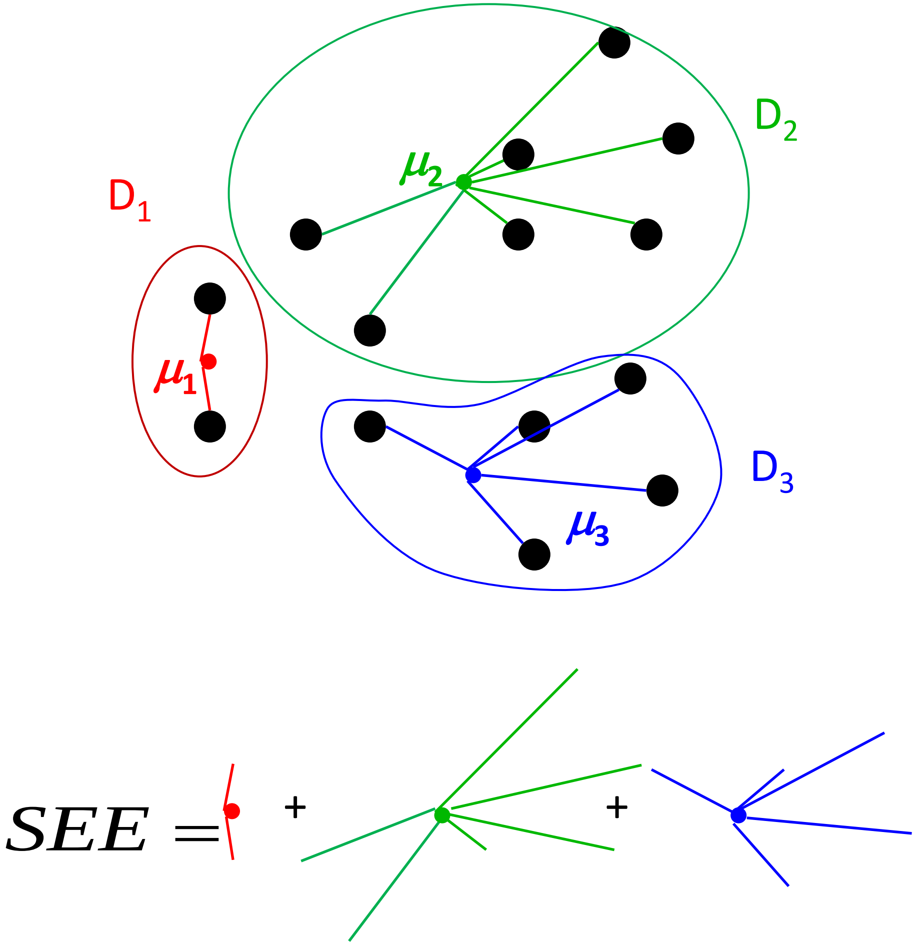
$SSE =$



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



Good (tight) clustering
smaller value of SSE

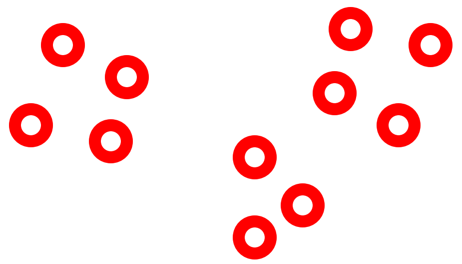


Bad (loose) clustering
larger value of *SSE*

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

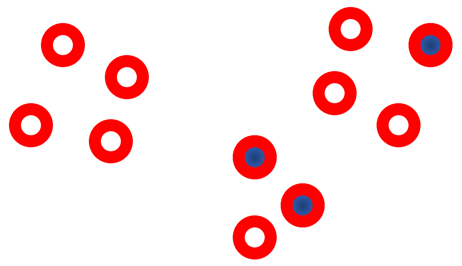
Algoritma K-means Clustering

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



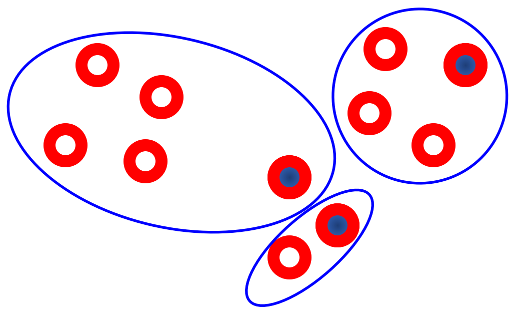
Algoritma K-means Clustering

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



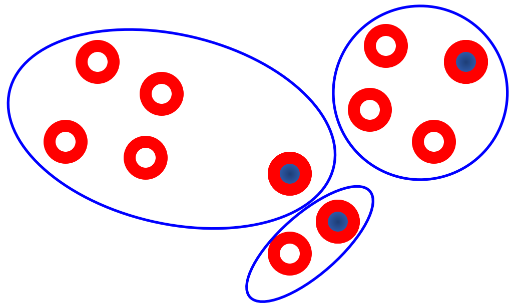
Algoritma K-means Clustering

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



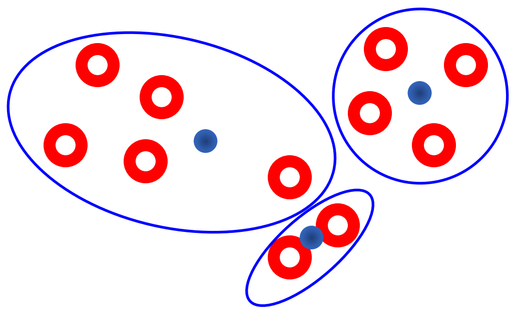
Algoritma K-means Clustering

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



Algoritma K-means Clustering

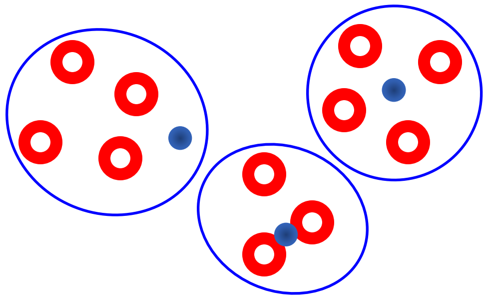
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



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

Algoritma K-means Clustering

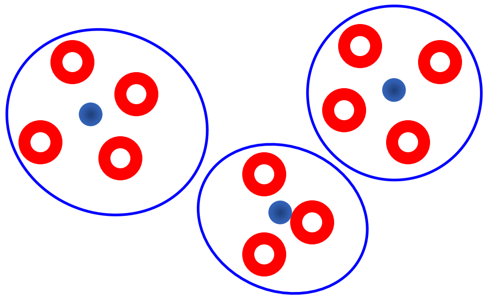
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



- 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

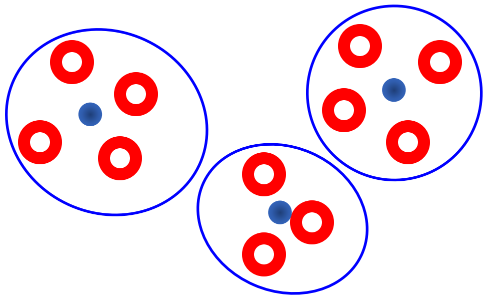
- Initialization step
  1. pick ***k*** cluster centers randomly
  2. assign each sample to closest center



- 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

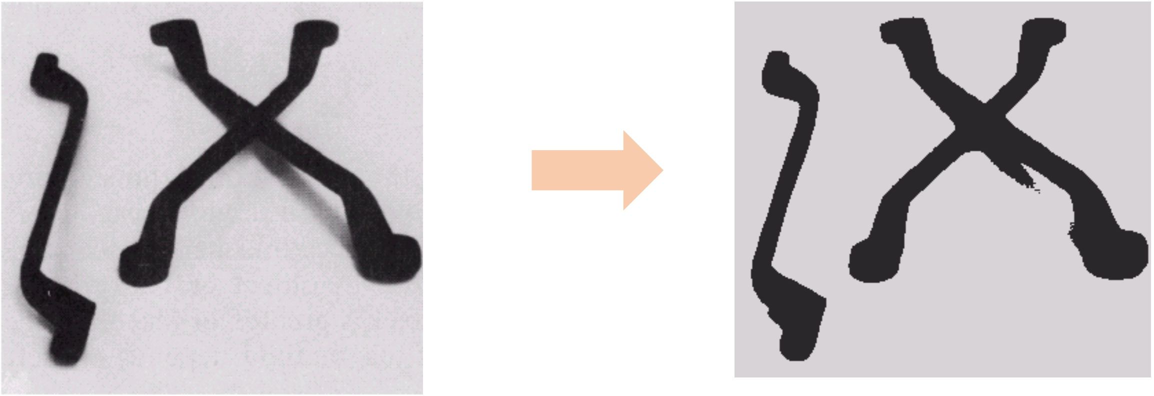
$$E_k(D, \mu) = \sum_{i=1}^{k} \sum_{x \in D_i} \left\| x - \mu_i \right\|^2$$

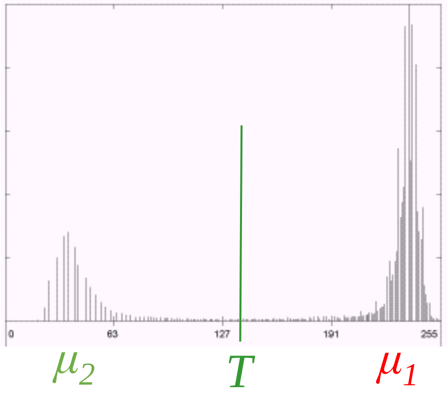
optimization variables

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

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

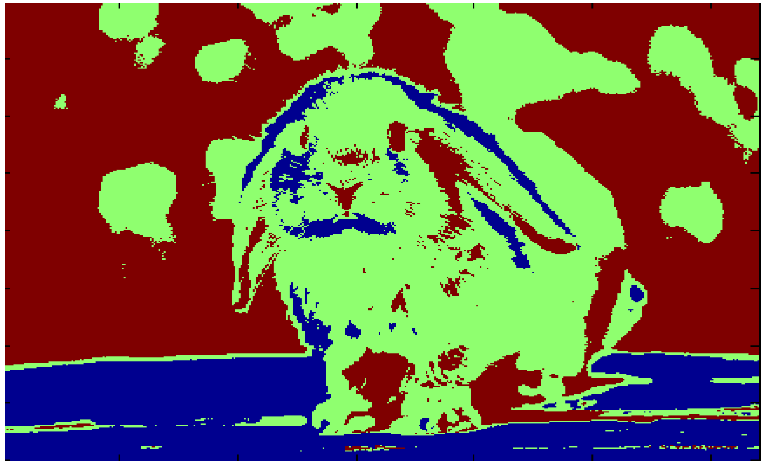
Contoh hasil *K-means clustering*





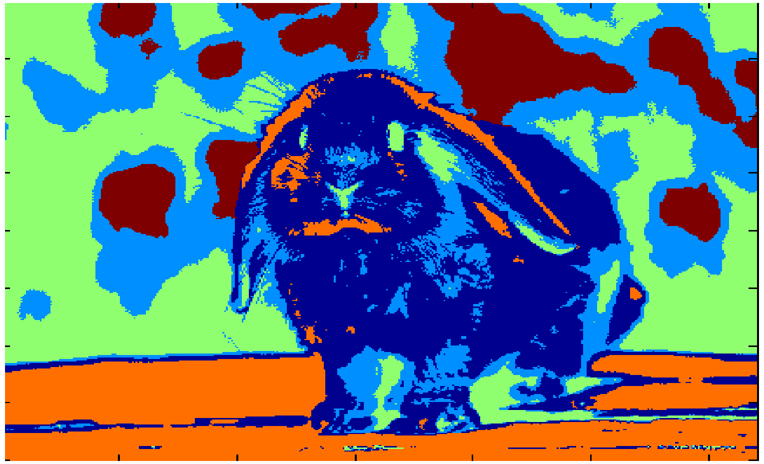
K-means menghasilkan
Pengelompokan yang kompak

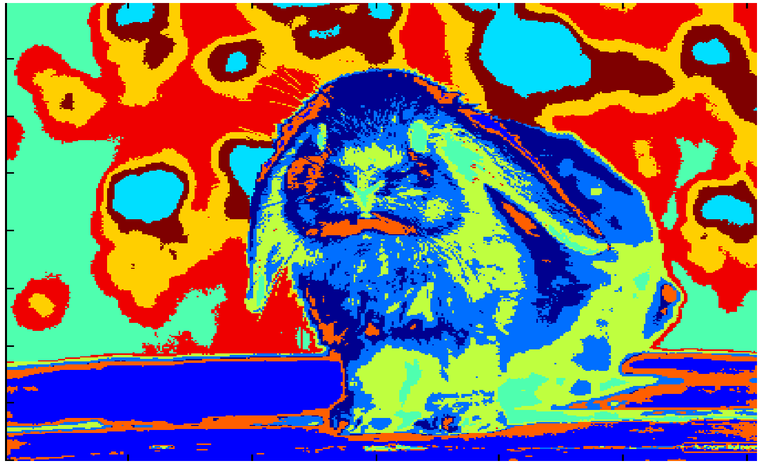
Pada kasus ini, K-means (K=2) secara otomatis
menemukan nilai ambang yang bagus antara 2 cluster

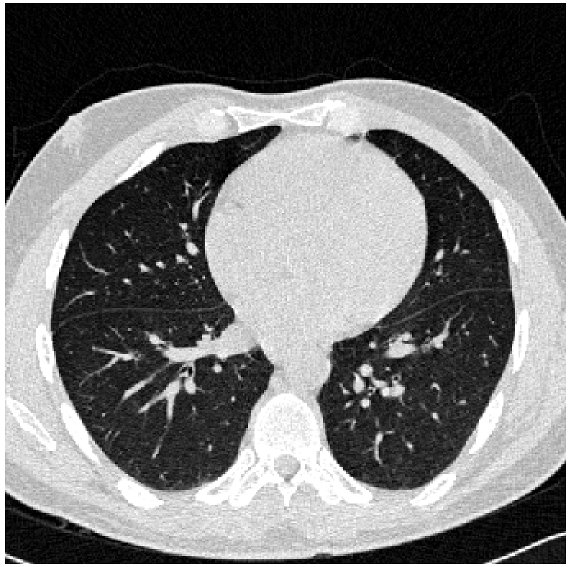


***k* = 3**

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

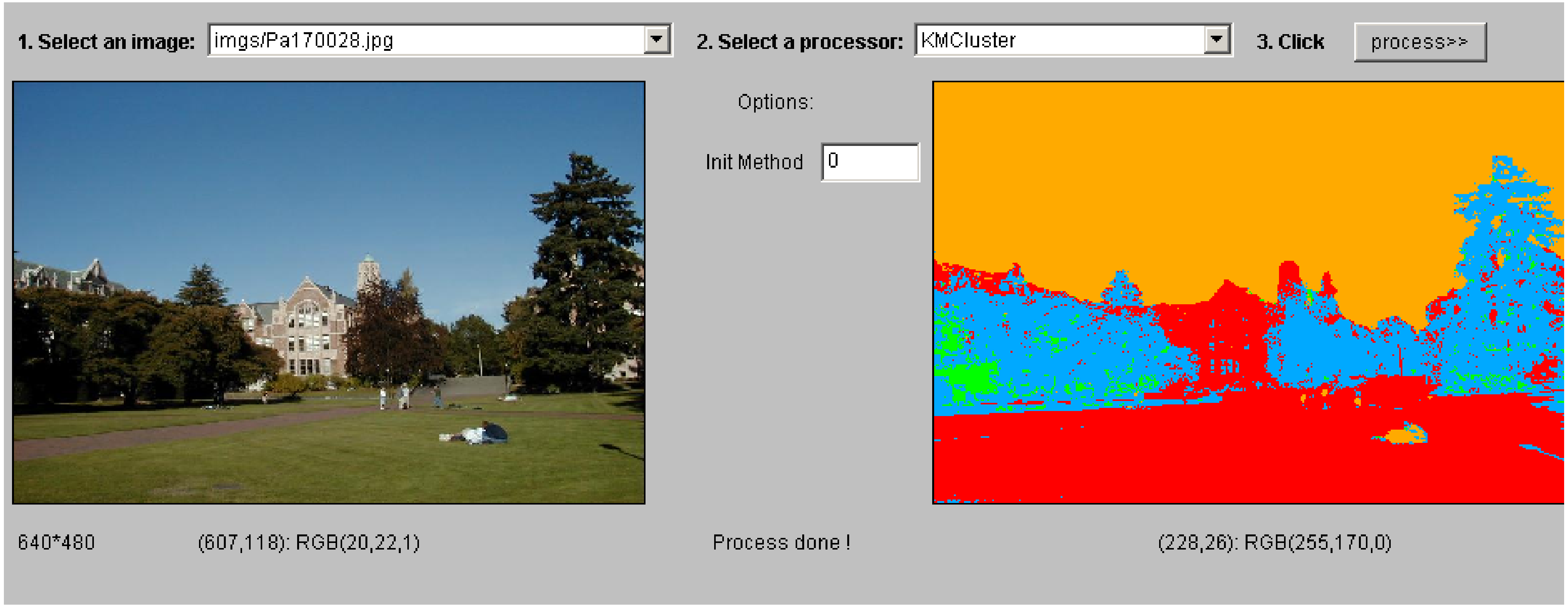


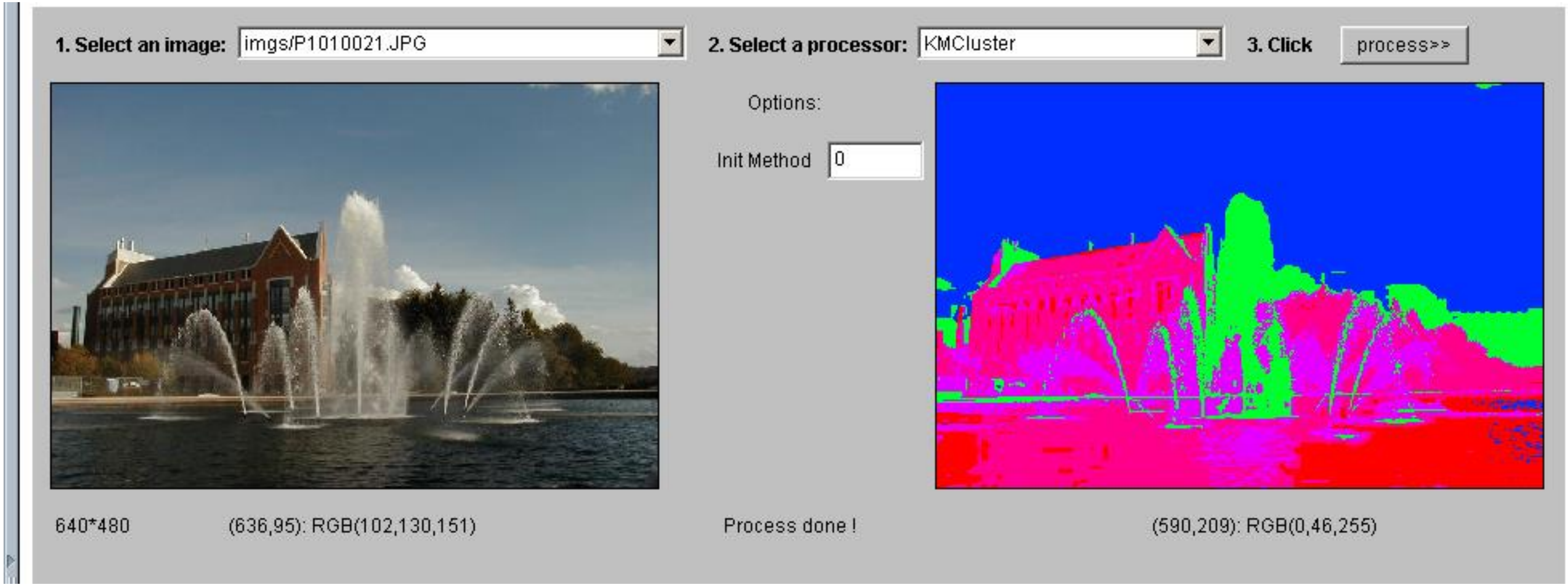




An image(I)

Three cluster image (J)on gray values of I

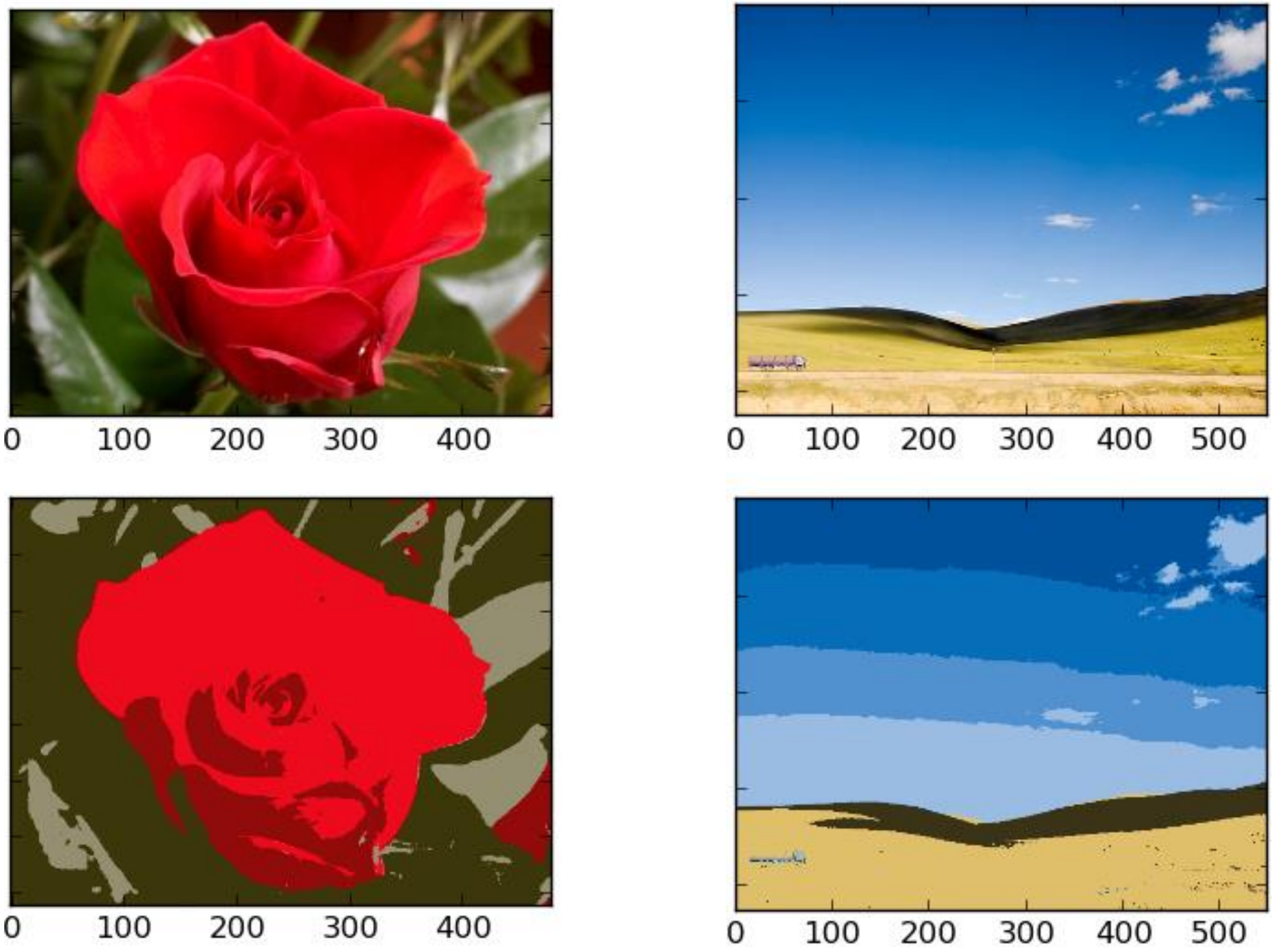

1. Select an image:
imgs/Pa170028.jpg
2. Select a processor:
KMCluster
3. Click
process>>
Options:
Init Method
0
640*480
(607,118): RGB(20,22,1)
Process done !
(228,26): RGB(255,170,0)


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)

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

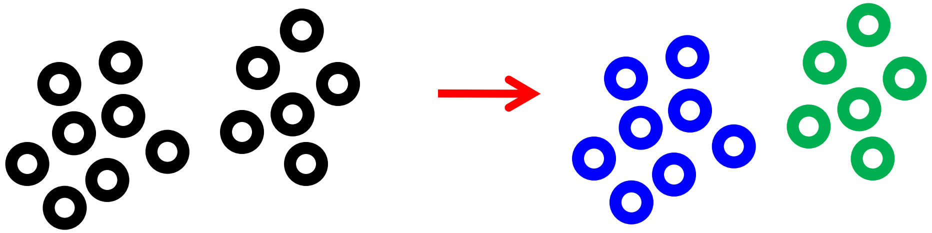
XY features only

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

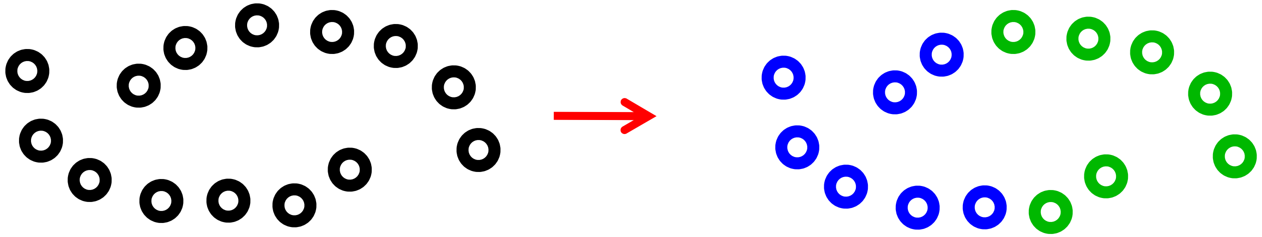
Sumber: CS 4487/9587 Algorithms for Image Analysis: Basic Image Segmentation

Sifat-sifat K-means

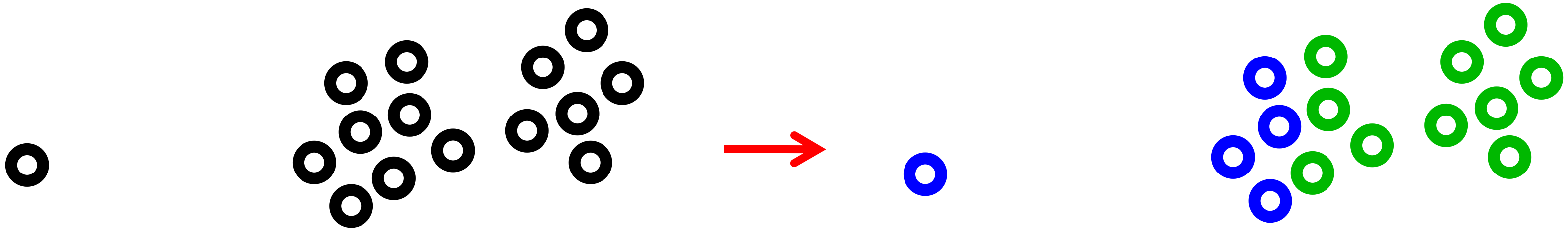
- Works best when clusters are spherical (blob like)



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



- Sensitive to outliers



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

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

equivalent (easy to check)

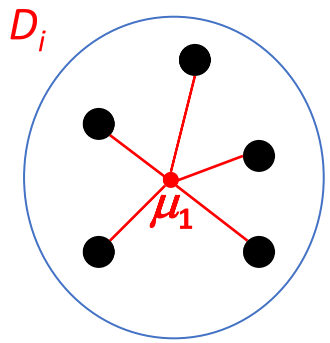
$$E_k \sim -\sum_{i=1}^{k} \sum_{x \in D_i} \log P(x \mid \mu_i) \quad + const$$

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

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

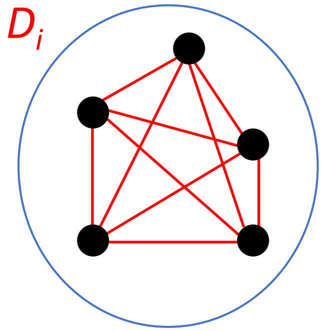
just plug-in expression

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



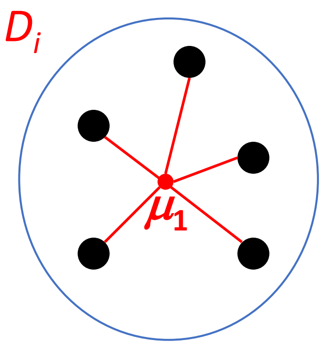
$$E_k = \sum_{i=1}^{k} \sum_{x \in D_i} \| x - \mu_i \|^2$$

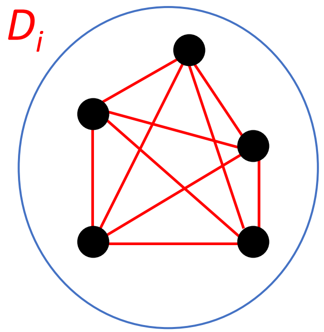
equivalent (easy to check)



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

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





both formulas can be written as

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

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

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 ← K-means fits Gaussian models
  - Sensitive to outliers ← Quadratic errors are such
  - Sensitive to choice of ***k*** ← 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)

Program Matlab untuk image segmentation dengan K-means

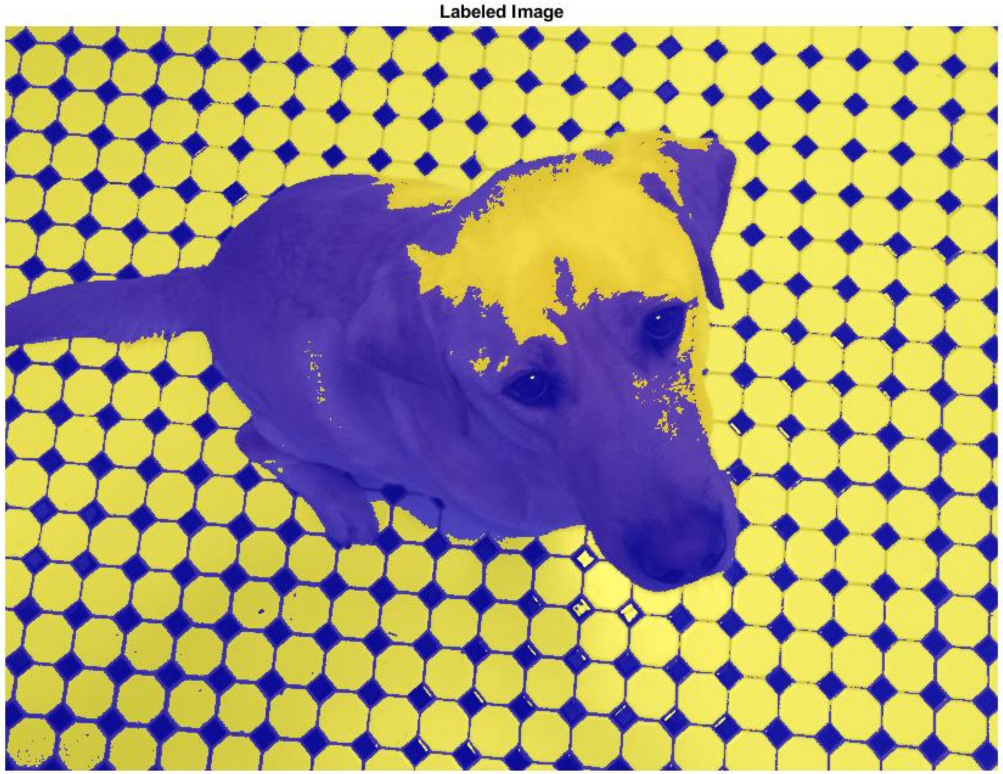
- Fungsi imsegkmeans hanya tersedia untuk Matlab R2022a

```
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")
```



Segmentasi Citra dengan Deep Learning

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