

# Interpretasi dan Pengolahan Citra

## Pengolahan dan Analisis Citra Berbasis Graph

Akmal  
Rinaldi Munir

Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung

1

## Outline



- Pendahuluan
- Teori Graph Dasar
  - Terminologi
  - Tree
    - MST →Kruskal, Prim, Boruvka
- Representasi Citra Menggunakan Graf
- Segmentasi citra Berbasis Graph
  - Berdasarkan MST
- Ekstraksi graph dengan Region Adjacency Graph (RAG)
- Penerapan : Graph Matching →Image Mining

2



## Pendahuluan

- Pertumbuhan citra digital → aplikasi citra meningkat
  - Meliputi integrasi multimedia, animasi computer, video games, seni digital, kedokteran, biometri dll
- Area pengolahan dan analisis citra sangat luas dan area riset mulai dari pemrosesan *low level* sd analisis *high level*
- Graph muncul sebagai suatu cara representasi untuk pengolahan dan analisis citra

3

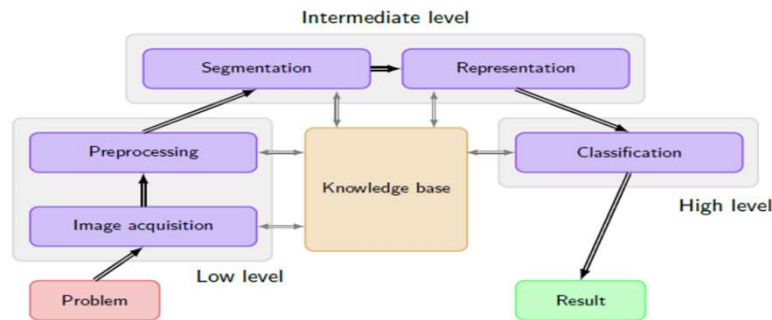


## Pengantar

- Proses segmentasi citra dan ekstraksi fitur merupakan proses yang penting sebelum dilakukan proses analisis citra.
- Terdapat beberapa metoda dalam segmentasi citra dalam rangka membagi citra menjadi beberapa segmen yang memiliki fitur piksel yang sama atau mirip
- Metoda berbasis graf merupakan salah satu metoda untuk melakukan proses segmentasi citra.
- Pada proses pengolahan citra berbasis graf, ada beberapa cara dalam merepresentasikan graf citra dimana simpul graf bisa berupa piksel atau region (superpixel).
- Pada dasarnya permasalahan yang dilakukan dengan metoda berbasis graf adalah membagi suatu graf menjadi beberapa sub-graf sedemikian sehingga setiap bagian subgraf itu sesuai dengan yang diinginkan.
- Terdapat beberapa kategori metoda untuk segmentasi citra berbasis graf yaitu berdasarkan minimum spanning tree, berdasarkan graf cut, interaktif berdasarkan graf cut *markov random field* dan berdasarkan shortest path.

4

## Tahap Analisis Citra [Malmberg, 2015]



- ❑ Bagaimana melakukan klasifikasi dan pengenalan objek yang efektif dan efisien dengan menggunakan representasi graf dikaitkan dengan waktu pemrosesan (run time) yang cepat dan hasil yang akurat (berdasarkan pengukuran tertentu)
- ❑ Bagaimana mengekstraksi graph dari citra yang nantinya akan menjadi fitur representasi bagi proses klasifikasi dan pengenalan objek yang tepat

5

## Alasan Menggunakan graph

1. Representasi sederhana cocok untuk pengembangan metode yang efisien
2. Fleksibilitas dalam merepresentasikan berbagai macam tipe citra.
3. Algoritma dan teorema teori graph yang ada bisa dikembangkan kembali untuk digunakan dalam bidang analisis citra.
4. Adanya penambahan informasi *high level* (relasi) pada representasi *low level* (masalah *semantic gap*)

6



## Terminologi Graph

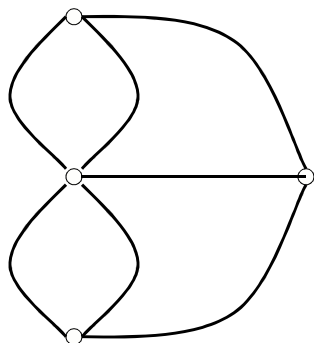
7

## Jembatan Konigsberg



Perjalanan Euler adalah : Perjalanan dari suatu simpul kembali ke simpul tersebut dengan melalui setiap sisi tepat satu kali.

- Perjalanan Euler akan terjadi, jika :
  - Graf terhubung.
  - Banyaknya sisi yang datang pada setiap simpul adalah genap.



8

8

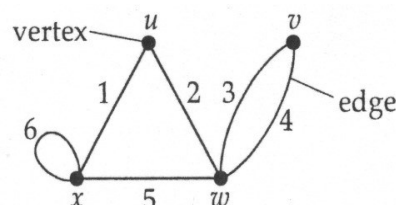
## Teori Graf Dasar



Graf  $G$ , adalah pasangan himpunan  $(V, E)$  ditulis dengan notasi  $G = (V, E)$

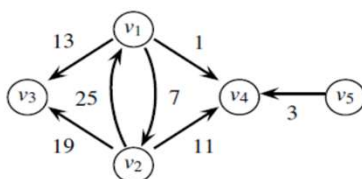
- (1) Himpunan  $V$  adalah himpunan tak kosong dari simpul (vertex atau node).
- (2) Himpunan  $E$  adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul

- Terminologi graf
  - Graf sederhana
  - Graf berarah dan tidak berarah
  - Adjacency (bertetangga)
  - Incidence (bersisian)
  - Subgraf
  - Graf berbobot
  - Graf Isomorfis
  - Graf planar

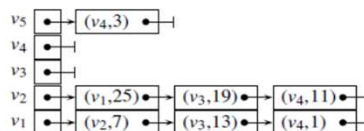


9

## Representasi Graf



	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
$v_1$	0	7	13	1	0
$v_2$	25	0	19	11	0
$v_3$	0	0	0	0	0
$v_4$	0	0	0	0	0
$v_5$	0	0	0	3	0



	$e_{12}$	$e_{13}$	$e_{14}$	$e_{21}$	$e_{23}$	$e_{24}$	$e_{54}$
$v_1$	+1	+1	+1	-1	0	0	0
$v_2$	-1	0	0	+1	+1	+1	0
$v_3$	0	-1	0	0	-1	0	0
$v_4$	0	0	-1	0	0	-1	-1
$v_5$	0	0	0	0	0	0	+1

- List Adjacency
- Matriks Adjacency
- Matriks incidence

10



## Path, Tree dan Keterhubungan

- Path adalah urutan dari simpul dimana setiap simpul bertetangga dengan simpul sebelumnya.
- Siklus adalah path dengan simpul awal sama dengan simpul akhir.
- Tree ( $T$ ) adalah suatu graph khusus dengan sifat yang dimilikinya sebagai berikut : Misalkan  $T$  adalah graf dengan  $n$  simpul. Maka
  - $T$  adalah terhubung dan tidak memiliki cycles.
  - $T$  memiliki  $n-1$  sisi dan tidak memiliki cycles.
  - $T$  adalah terhubung dan memiliki  $n-1$  sisi
  - $T$  adalah terhubung dan penghapusan satu sisi akan membuat  $T$  menjadi tidak terhubung
  - Setiap 2 simpul dalam  $T$  adalah terhubung dengan tepat memiliki satu path
  - $T$  tidak memiliki cycles, tetapi penambahan satu sisi akan membentuk sebuah cycle.
- **Suatu graf  $G$  disebut terhubung** jika untuk setiap dua buah simpul dari graf terdapat jalur yang menghubungkan kedua simpul tersebut

11



## Minimum Spanning Tree (MST)

12

## Minimum Spanning Tree (MST) Menggunakan Algoritma Kruskal (1956)



- MST adalah spanning tree yang jumlah bobotnya paling minimal diantara semua spanning tree suatu graf.

### Kruskal's algorithm

Set  $E_{new} = \emptyset$ .

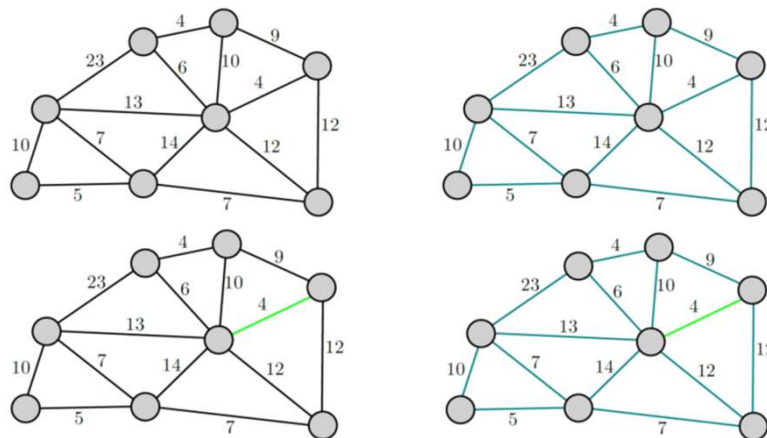
while there exists an edge  $e_{p,q}$  such that  $p \not\sim q$  do  
( $V, E_{new}$ )

| Choose such an edge with minimal weight and add it to  $E_{new}$ .

end

13

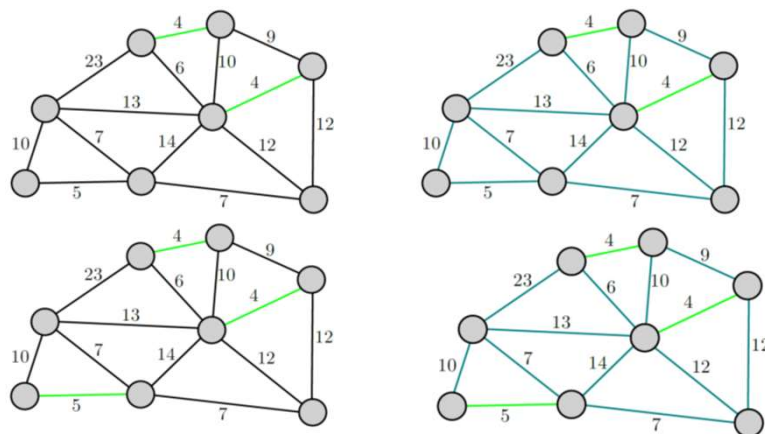
## Algoritma Kruskal(1)



- Biru – pilihan (pilih sisi terkecil yang tidak membentuk cycle)
- Hijau – yang dipilih (Tambahkan sisi ke dalam Tree)

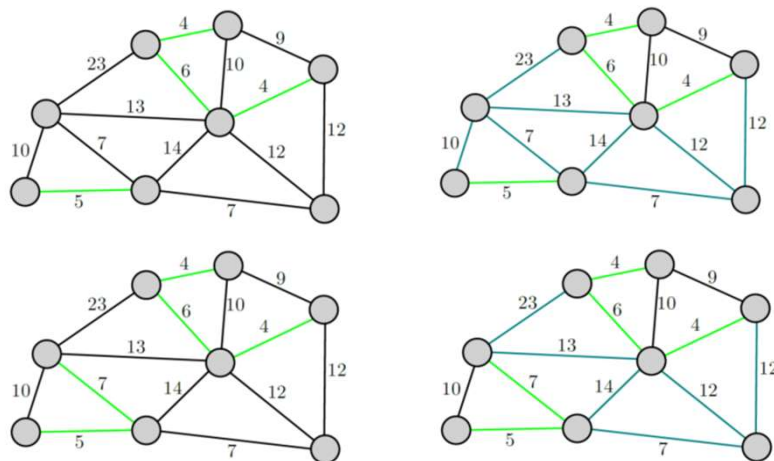
14

## Algoritma Kruskal(2)



15

## Algoritma Kruskal(3)

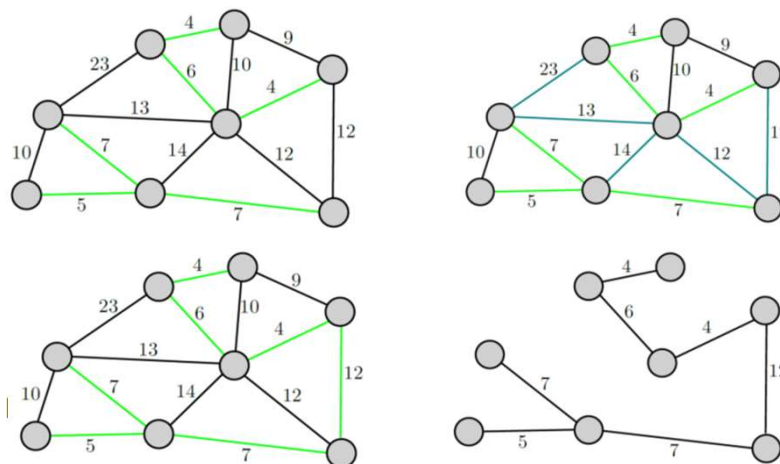


16





## Algoritma Kruskal(4)



□ Minimum Spanning Tree yang dihasilkan

17



## MST Menggunakan Algoritma Prim (1957)

### Prim's algorithm

Set  $V_{new} = \{v\}$ , where  $v$  is an arbitrary vertex in  $V$ .

Set  $E_{new} = \emptyset$ .

**while**  $V_{new} \neq V$  **do**

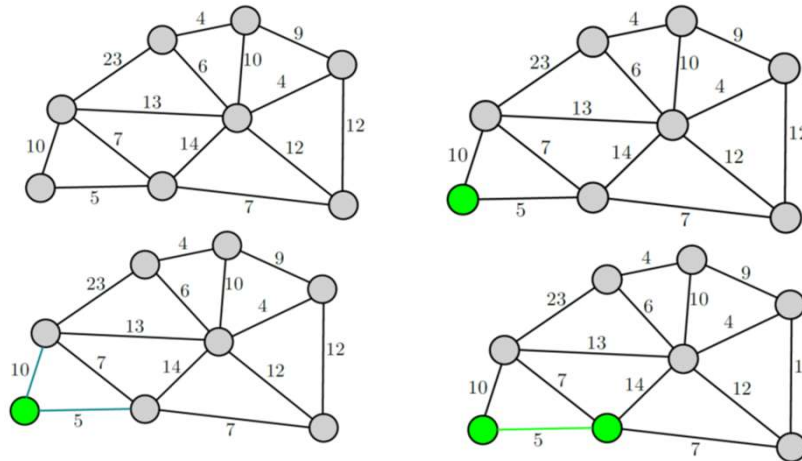
    Choose an edge  $e_{p,q}$  with minimal weight such that  $p$  is in  $V_{new}$  and  $q$  is not.

    Add  $q$  to  $V_{new}$  and  $e_{p,q}$  to  $E_{new}$ .

**end**

18

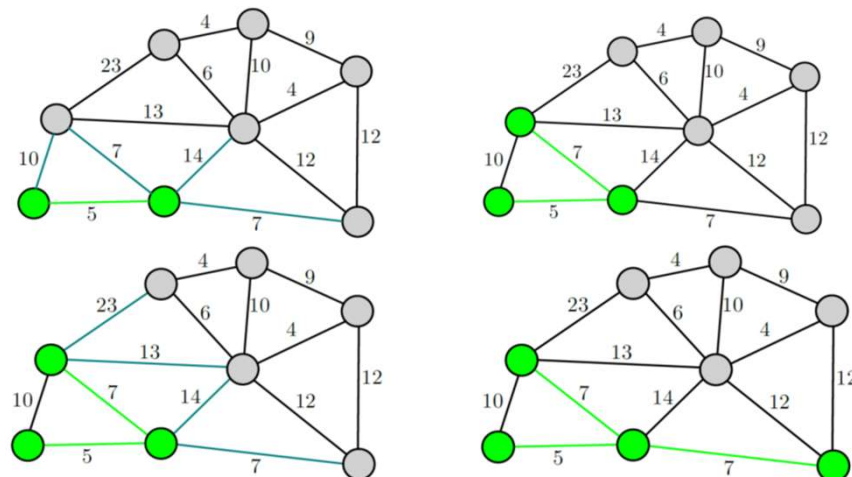
## Algoritma Prim(1)



- ▣ Pilih satu simpul sembarang dan tambahkan ke  $V^*$
- ▣ Pilih sisi tetangga yang bobotnya terkecil dan tambahkan simpul ke  $V^*$  dan tambahkan sisi ke  $E^*$

19

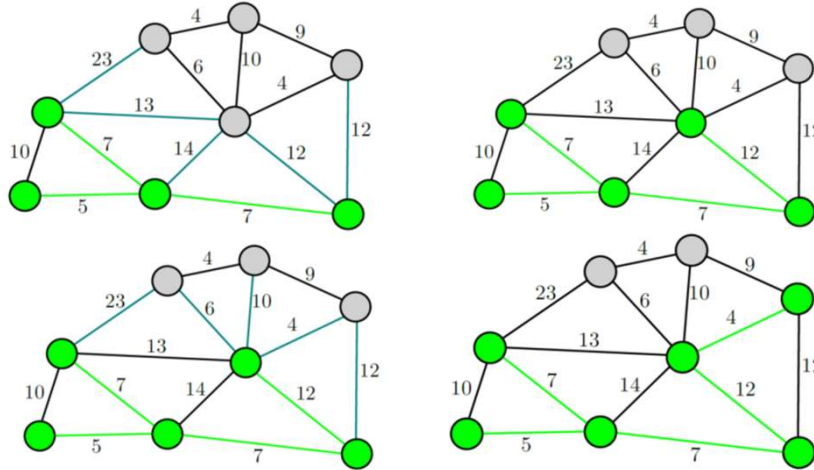
## Algoritma Prim(2)



-

20

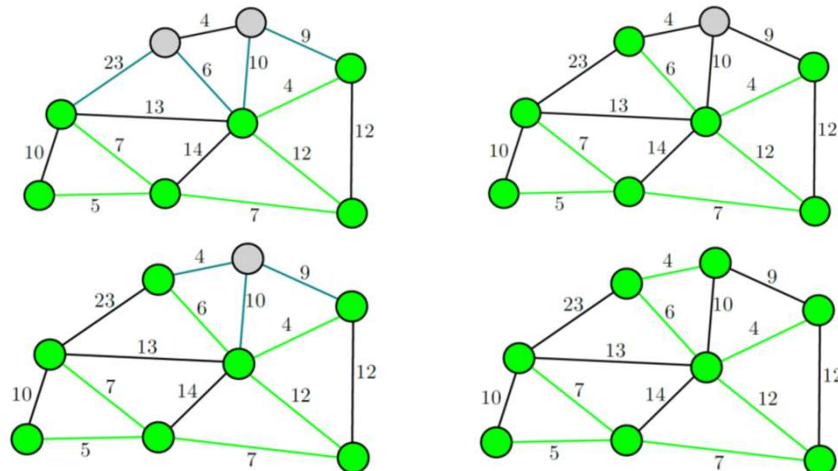
## Algoritma Prim(3)



□ -

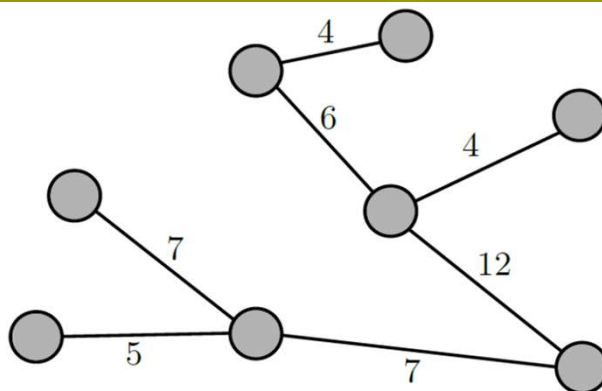
21

## Algoritma Prim(4)



22

## Algoritma Prim(5)



▣ Hasil Minimum Spanning Tree

23

## MST Menggunakan Algoritma Boruvka (1926) / Sollin (1965)



Input: A connected graph  $G$  whose edges have distinct weights

Initialize a forest  $T$  to be a set of one-vertex trees, one for each vertex of the graph.

While  $T$  has more than one component:

For each component  $C$  of  $T$ :

Begin with an empty set of edges  $S$

For each vertex  $v$  in  $C$ :

Find the cheapest edge from  $v$  to a vertex outside of  $C$ , and add it to  $S$

Add the cheapest edge in  $S$  to  $T$

Combine trees connected by edges to form bigger components

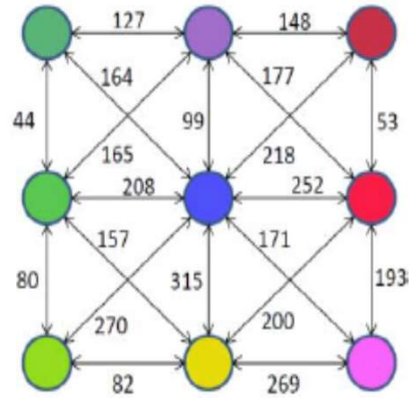
Output:  $T$  is the minimum spanning tree of  $G$ .

24

# Algoritma Boruvka



RGB (90, 180, 120)	RGB (160, 110, 200)	RGB (200, 50, 70)
RGB (90, 200, 80)	RGB (80, 80, 250)	RGB (250, 30, 70)
RGB (150, 220, 30)	RGB (230, 220, 10)	RGB (250, 100, 250)

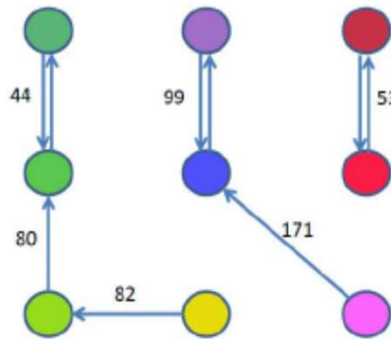


25

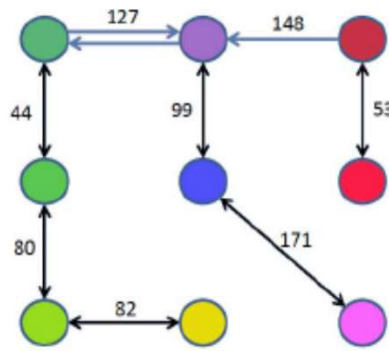
# Algoritma Boruvka



## Iteration 1



## Iteration 2



26



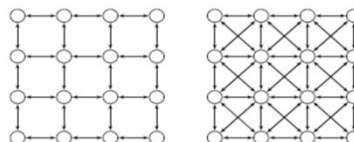
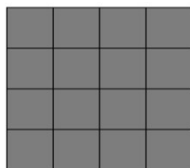
## Representasi Citra

27



## Representasi Citra Menggunakan Graf

- biasanya beroperasi pada graf piksel bertetangga, yaitu graf dengan himpunan simpulnya adalah himpunan dari piksel citra dan himpunan sisinya adalah relasi adjacency dari piksel citra



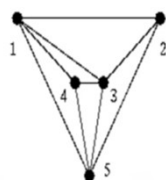
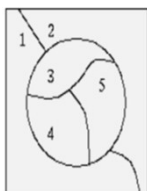
- Sebuah graf dibentuk dari piksel citra yang bisa direpresentasikan dengan 4 tetangga atau 8 tetangga. Untuk bobot dalam RGB bisa menggunakan jarak Euclidean antara 2 piksel yang bertetangga [Ali Saglam]:
- $$d(p, q) = \sqrt{(R_p - R_q)^2 + (G_p - G_q)^2 + (B_p - B_q)^2}$$
- Selanjutnya untuk merepresentasikan suatu graf bisa dilakukan dengan menggunakan list adjacency atau matriks adjacency

28



## Region Adjacency Graph.

- Menggunakan graf dual



- Region yang terdiri dari pixel-pixel yang memiliki kedekatan atau kemiripan dinamakan dengan superpixel.
- Beberapa algoritma untuk membangkitkan superpixel antara lain MST, Normalisasi Cut, Watershed dan Turbo Pixel.

29



## Segmentasi

30



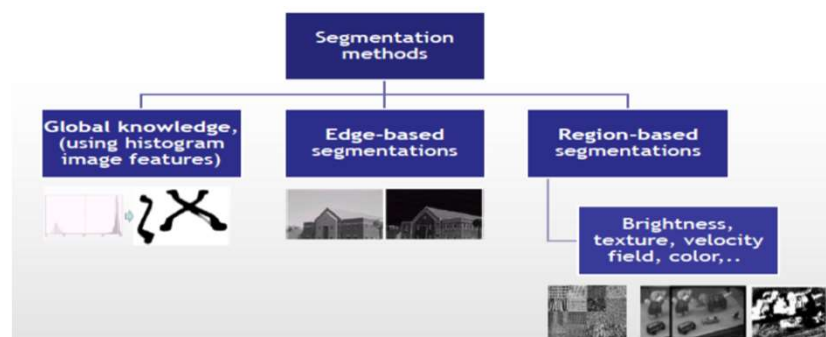
## Segmentasi citra

- ❑ Graph Cut adalah himpunan sisi yang jika dihapus akan membuat himpunan simpul dari graph terbagi atas dua himpunan yang saling lepas.
- ❑ Segmentasi Citra dengan graph dilakukan dengan menghapus sejumlah sisi penghubung antar dua segmen atau grup sehingga jumlah bobot yang dihapus minimal.
- ❑ Minimisasi jumlah bobot akan mengarahkan pada keadaan dimana piksel-piksel yang nilai fiturnya serupa akan berada pada segmen yang sama dan piksel-piksel yang nilai fiturnya tidak serupa akan berada pada segmen yang berbeda.

31



## Metoda Segmentasi Citra



- ❑ segmentasi berbasis non-graph hanya berdasarkan pada informasi batas **atau** region,
- ❑ segmentasi berbasis graph berdasarkan pada informasi batas **dan** region

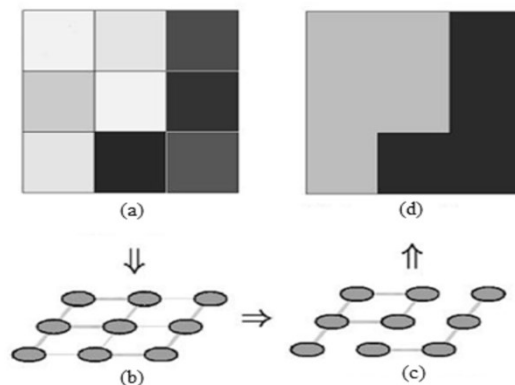
32



## Segmentasi citra Berbasis Graf



- Prinsip dasar dari metoda segmentasi berbasis graf adalah graf partisi.
- Dalam hal ini berarti mencari satu himpunan subgraph ( $SG_1, SG_2, \dots, SG_n$ ) dari graph  $G$ .



33

### 1. Metoda berdasarkan Minimum Spanning Tree (MST)



- MST dengan algoritma Boruvka
- MST dengan algoritma Kruskal
- MST dengan algoritma Prim

34

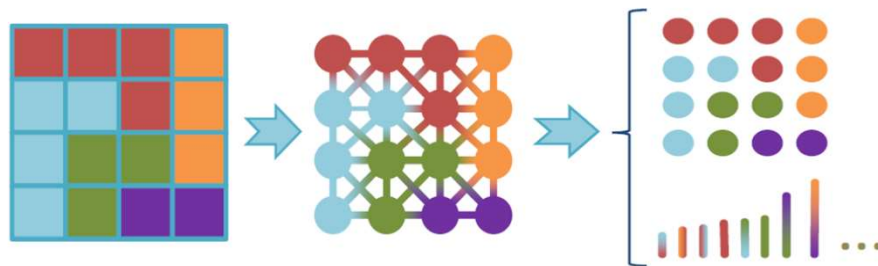


### 1. Metoda berdasarkan Minimum Spanning Tree (MST)

- ❑ Algoritma MST adalah algoritma greedy dan kompleksitas waktunya adalah polynomial.
- ❑ Metoda segmentasi berdasarkan MST pada dasarnya berhubungan dengan clustering berbasis graf, dimana data yang akan di cluster direpresentasikan sebagai graf tidak berarah.
- ❑ Untuk pengelompokan, sisi dengan bobot tertentu ditetapkan antara dua simpul jika keduanya bertetangga mengikuti system ketetanggaan yang sudah diberikan. Clustering didapatkan dengan membuang sisi dari graf untuk membentuk subgraph-subgraf eksklusif .
- ❑ Proses clustering biasanya menekankan pentingnya prinsip gestalt dari kemiripan (similarity) dan kedekatan (proximity).

35

### Representasi Citra Menggunakan Graf pixel 8-connected dan bobot jarak → MST



$$d(p, q) = \sqrt{(R_p - R_q)^2 + (G_p - G_q)^2 + (B_p - B_q)^2}$$

36

### Region Merging (Felzenszwalb & Huttenloche)

Kriteria :  $Dif f(C1, C2) < \min \left( Int(C1) + \frac{k}{|C1|}, Int(C2) + \frac{k}{|C2|} \right)$

```

For all edges from w_min to w_max
  if ( edge is internal to region )
    continue
  else
    compute Int(C1), Int(C2), D(C1, C2)
    if ( boundary exists between C1, C2 )
      continue
    else
      merge C1, C2 into new region
    
```

Int(C1)  
Int(C2)

Tidak Merge krn  
Dif f(C1,C2) false

Merge dg bobot baru  
Dif f(C1,C2) true

37

### RAG (Region Adjacency Graph)

- Berdasarkan region yang didapatkan melalui segmentasi maka dibentuk suatu graph dengan satu region diwakili dengan satu simpul yang diambil dari **centroid region** tersebut dan relasi ketetanggaan antar region dinotasikan dengan suatu sisi
- Model representasi ini disebut dengan **Region Adjacency graph (RAG)**.
- Dengan menggunakan Region adjacency Graph jumlah simpul yang bisa dihasilkan akan jauh berkurang karena ada reduksi dari jumlah piksel dalam satu region .

38

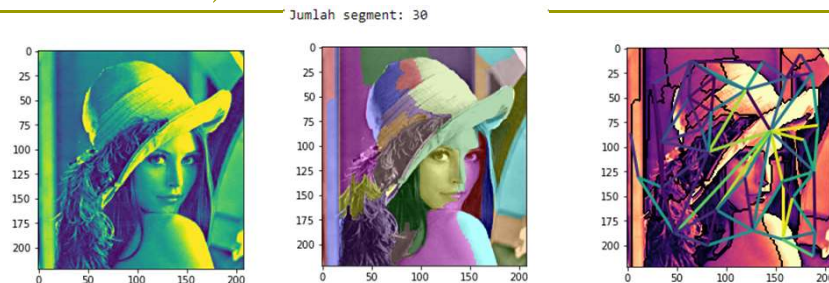
## Hasil ekstraksi graph

kamera → jumlah simpul 64 dan jumlah sisi 177,  
 lena → jumlah simpul 35 dan jumlah sisi 92



39

## Contoh uji coba lain



- **Nodes yang terbentuk, centroidnya, dan mean color :**  
 [(0, {'total color': array([244972., 244972., 244972.]), 'centroid': (86, 3), 'labels': [0], 'mean color': array([158.76344783, 158.76344783, 158.76344783]), 'pixel count': 1543}), .....
- **Edges yang terbentuk:**  
 [(0, 1), (1, 25), (1, 2), (2, 18), (2, 3), (2, 13), (2, 25), .....
- **weight dari edges :**  
 edge[0][1] = 57  
 edge[1][25] = 207

40



## Graph Matching

- Graph Matching adalah proses mengevaluasi kesamaan antara dua graf dengan membandingkannya untuk menemukan korespondensi yang bersesuaian antara simpul-simpul dan sisi-sisinya.
- Ada dua jenis pencocokan graph yang utama yaitu exact matching dan inexact matching

41

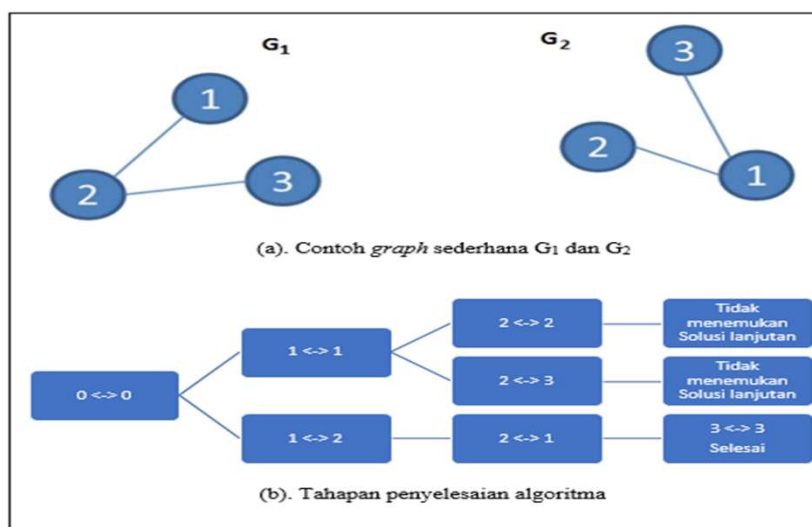


## Algoritma VF2 dalam Exact matching

- Metode exact matching adalah suatu metoda yang mencari korespondensi yang tepat antara dua graph jika ada.
- Secara struktural, memastikan bahwa pemetaan antara node dari dua graph harus menjaga sisi yang insiden dengan node tersebut yang berarti bahwa jika dua node dalam satu graph dihubungkan oleh sebuah sisi, maka ketika dipetakan pada dua node di graph lain, juga dihubungkan oleh sebuah sisi.

42

## VF2



43

### Graph Edit Distance dalam Inexact Matching



- Algoritma inexact matching tidak mencari korespondensi yang persis sama antara dua *graph* tetapi mencari kemiripan dari dua *graph* yang dibandingkan.
- Dalam hal ini, cost (atau jarak) dihitung dengan menghitung perbedaan perhitungan di antara atribut yang sesuai.
- Pencocokan akan mencari pemetaan yang meminimalkan cost tersebut.

44



## GED

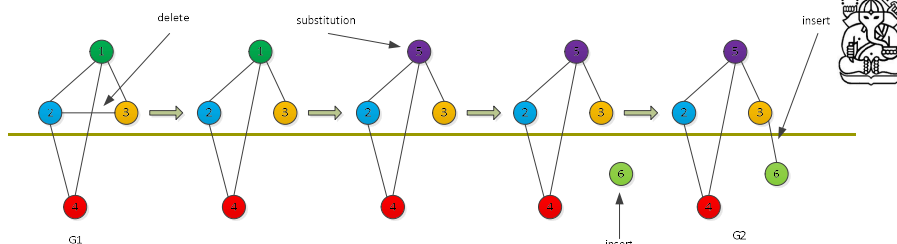
- Ide dasar dari graph edit distance adalah menentukan ukuran ketidaksamaan antara dua graf dengan jumlah minimum distorsi yang diperlukan untuk mengubah satu graf menjadi yang lain
- Untuk tujuan ini, sejumlah distorsi atau operasi edit, yang terdiri dari penyisipan, penghapusan, dan substitusi dari kedua simpul dan sisi didefinisikan

Definisi

Diberikan dua buah graph  $G_1 = (V_1; E_1)$ , dan  $G_2 = (V_2; E_2)$ . Graph Edit Distance dari  $G_1$  dan  $G_2$  adalah

$$d(G_1, G_2) = \min_{(ed_1, \dots, ed_k) \in P(G_1, G_2)} \sum_{i=1}^k c(ed_i)$$

45



- Untuk dua graf  $G_1 = (V_1; E_1)$  dan  $G_2 = (V_2; E_2)$  dan parameter nonnegative  $\alpha, \beta, \gamma, \theta \in R^+ \cup \{0\}$ , fungsi cost ini didefinisikan untuk semua node  $u \in V_1$ , dan  $v \in V_2$  untuk semua sisi  $p \in E_1$  dan  $q \in E_2$ , dengan

- $c(u \rightarrow \varepsilon) = \gamma$
- $c(\varepsilon \rightarrow v) = \gamma$
- $c(u \rightarrow v) = \alpha \cdot \|L(u) - L(v)\|$
- $c(p \rightarrow \varepsilon) = \theta$
- $c(\varepsilon \rightarrow p) = \theta$
- $c(p \rightarrow q) = \beta \cdot \|L(p) - L(q)\|$

Dengan  $L(\cdot)$  notasi untuk label dari elemen

46

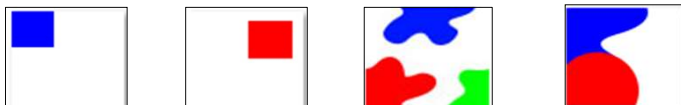
## Contoh Pemrosesan Citra berbasis Graph

47

## Data



### ▣ Citra Sintetis



### ▣ Citra Real (Motif inti batik : Grompol, Parang, Kawung)



48





## Pembangunan Region Adjacency Graph (RAG)

- Tahap Segmentasi (MST dan Kriteria Felzenszwalb)
- RAG
  - RAG dibangun dengan menghitung nilai *centroid* dari setiap label segmentasi,
  - menghitung nilai rata-rata RGB pada setiap region *centroid*,
  - kemudian menghubungkan antar *centroid* menggunakan *eucledian distance*.



Penggambaran proses ekstraksi RAG

49



## Output tahap RAG

- *Nodes*, simpul-simpul yang terbentuk
- *Total Color*, total nilai piksel RGB dari masing-masing region
- *Centroid*, nilai titik tengah dari masing-masing region
- *Mean Color*, nilai rata-rata piksel dalam RGB dari masing-masing region
- *Edges*, sisi-sisi yang terbentuk dari setiap pasangan simpul
- *Weight*, nilai *eucledian distance* dari masing-masing *edge*

50

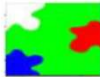


Nama File	Citra Asli	Citra Segmentasi	RAG
img030.jpg			
img094.jpg			
img101.jpg			
img156.jpg			
batik020.jpg			
batik114.jpg			
batik173.jpg			

51

## Tahap Uji Coba Inexact Graph Matching

Kasus 1 : Pengujian VF2 dengan menggunakan data citra sintetis

Data Uji Citra Artificial : lmg134.jpg


□ Hasil

Nama File	Citra Segmentasi	RAG	Citra Asal
lmg101.jpg			
lmg104.jpg			
lmg135.jpg			

52

Kasus 2 : Pengujian VF2 dengan menggunakan data citra real

Data Uji Citra Real : Batik082.jpg




■ Hasil

Nama File	Citra Segmentasi	RAG	Citra Asal
Batik083.jpg			
Batik084.jpg			
Batik085.jpg			

53

## Uji dengan GED

Data Uji Citra Real : Batik070.jpg



■ Hasil

Nama File	Citra Segmentasi	RAG	Citra Asal	Cost GED
Batik070.jpg				0
Batik074.jpg				7 (substitusi)
Batik082.jpg				4 (insert, delete, substitusi)
Batik092.jpg				6 (insert, delete, substitusi)

54



## Image Mining berbasis graph

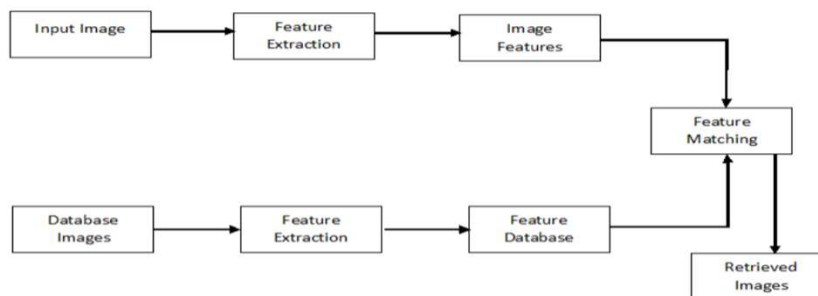
- ❑ Image mining bertujuan untuk menemukan relationship dan pola-pola yang tidak tersimpan secara eksplisit dalam database citra.
- ❑ Menambang dari koleksi citra dan dikombinasikan dengan data mining merupakan tema penting dari image mining.
- ❑ Ada lima teknik dalam image mining : pengenalan objek , image retrieval, image indexing, image classification dan clustering, dan Association rule mining

55



## Image Mining

### Diagram umum alur Image Retrieval



- ❑ Bagaimana memahami dan membahas lebih jauh pemodelan pengolahan citra dan analisis citra berbasis graph untuk dikaitkan dengan proses image mining (atau graph mining pada citra)?

56



## Pustaka Utama

---

- [1] Filip Malmberg, Image Processing using Graph, Centre for image analysis, Swedish University, 2015
- [2] Olivier Lezoray and Leo Grady, *Graph theory concepts and definitions used in image processing and analysis*, CRC Press, 2012
- [3] Saglam, Ali, and Nurdan Akhan Baykan. "An Efficient Object Extraction with Graph-Based Image Segmentation." *2015 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*. IEEE, 2015.

57



---

Terimakasih

58