

# SEGMENTASI CITRA MENGGUNAKAN PENGHAPUSAN TEKSTUR DAN K-MEANS DENGAN BATASAN RUANG

Gama Wisnu Fajarianto<sup>1</sup>, Handayani Tjandrasa<sup>2</sup>  
Jurusan Teknik Informatika, Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Email: gamawisnu@gmail.com

## ABSTRAK

*Segmentasi citra merupakan salah satu topik penting dalam pengolahan citra digital. Banyak metode yang dapat digunakan untuk proses segmentasi salah satu metode yang populer digunakan yaitu dengan segmentasi berbasis clustering K-means.*

*Langkah pertama segmentasi gambar menggunakan penghapusan tekstur dan K-Means dengan batasan ruang adalah berdasarkan pada prosedur de-texturing yang bertujuan mengubah input alami gambar berwarna menjadi gambar berwarna tanpa tekstur. Setelah gambar de-tekstur ini diestimasi, segmentasi akhir didapat dengan segmentasi K-means dengan batasan ruang. Batasan ruang ini membantu proses perulangan pelabelan dalam K-means agar berhasil menemukan segmentasi yang akurat.*

*Dari hasil keseluruhan skenario percobaan yang dilakukan didapatkan performa tertinggi nilai akurasi PRI 74.70% yaitu dengan nilai threshold, subdivision, size window, size fuse dan K secara berurutan adalah 0.37, 5000, 5, 400 dan 9.*

**Kata kunci:** segmentasi tekstur, K-Means clustering, penghapusan tekstur.

## 1. PENDAHULUAN

Segmentasi merupakan proses yang penting dalam pengolahan citra. Oleh karena itu Segmentasi telah secara luas dipelajari dan digunakan juga di dalam bidang perindustrian dalam sepuluh tahun terakhir ini.

Clustering adalah salah satu teknik yang digunakan dalam segmentasi. Clustering itu sendiri merupakan proses pengelompokan data dalam kelas-kelas atau cluster-cluster sehingga data-data dalam suatu cluster memiliki tingkat persamaan yang tinggi satu dengan yang lainnya tetapi sangat berbeda dengan data pada cluster lain. Dalam kaitannya dengan segmentasi, data pada clustering adalah piksel pada citra.

Metode K-Means adalah teknik yang paling umum digunakan dalam segmentasi berbasis clustering. Setelah dilakukan pemilihan dan ekstraksi fitur pada citra, fitur sampel, yang

berupa vektor dikelompokkan bersama sebagai cluster-cluster yang berkorespondensi terhadap kelas dari citra. Sekumpulan *connected piksel* masuk ke setiap kelas yang di estimasi sehingga menampilkan *region-region* yang berbeda.

Pada makalah ini akan dibahas mengenai salah satu metode segmentasi citra dengan menggunakan penghapusan tekstur dan K-Means dengan batasan ruang.

## 2. ANALISIS

Pada bagian ini akan dibahas mengenai proses penghapusan tekstur yang terdiri dari perhitungan bin deskriptor, clustering K-Means dengan bin deskriptor sebagai fitur, proses subdivisi, pergantian nilai piksel daerah dan rata-rata nilai piksel antar ruang warna. Kemudian dibahas proses segmentasi K-Means dengan batasan ruang yang terdiri dari

perhitungan edge map, perhitungan edge map homogen, pengambilan fitur pada hasil penghapusan tekstur dan segmentasi K-Means dengan batasan ruang.

## 2.1 Penghapusan Tekstur

Penghapusan tekstur bertujuan untuk mengurangi kompleksitas atau dimensi tinggi fitur deskriptor yang merupakan karakteristik pada warna tekstur.

Proses penghapusan tekstur secara umum membutuhkan perhitungan bin deskriptor yang dihitung pada setiap ruang warna. Ruang warna yang digunakan ada 10 yaitu RGB, HSV, YIQ, XYZ, LAB, LUV,  $I_1I_2I_3$ ,  $H_1H_2H_3$ ,  $YC_bC_r$ , TSL. Hasil perhitungan bin deskriptor digunakan sebagai fitur untuk proses clustering dengan menggunakan K-Means. Hasil clustering setiap ruang warna ini menjadi dasar untuk proses selanjutnya. Skema penghapusan tekstur dapat dilihat pada Gambar 1.

Proses penghapusan tekstur yang skema kerjanya dapat dilihat pada Gambar 1, diawali dengan menghitung bin deskriptor dan clustering K-Means. Setelah itu melakukan subdivisi dan mengganti nilai piksel dengan rata-rata nilai piksel clusternya. Terakhir dilakukan perhitungan rata-rata nilai piksel antar ruang warna.

### 2.1.1 Menghitung Bin Deskriptor

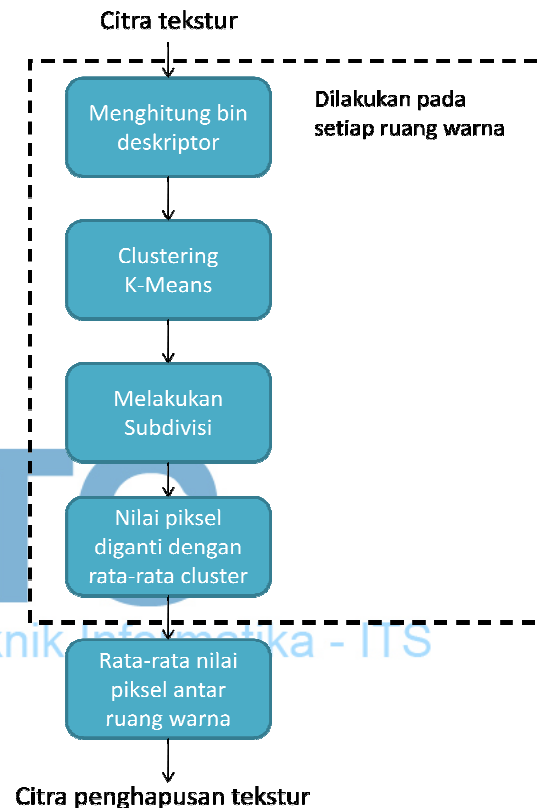
Bin deskriptor dibutuhkan pada proses clustering K-Means sebagai fitur vektor. Perhitungannya yaitu: untuk setiap piksel  $x \in N_x$  dengan nilai warna  $R_x, G_x, B_x$  dihitung

$$k = q^2 \cdot \left\lfloor \frac{q \cdot R_x}{256} \right\rfloor + q \cdot \left\lfloor \frac{q \cdot G_x}{256} \right\rfloor + \left\lfloor \frac{q \cdot B_x}{256} \right\rfloor \quad (1)$$

$$h[k] = h[k] + \frac{1}{N_0^2}$$

Dimana

- $N_x$  : Sekumpulan lokasi piksel  $x$  di dalam  $N_0 \times N_0$  neighborhood region yang memiliki pusat  $x$
- $h[]$  : Bin deskriptor : array bertipe float dari  $N_b$  ( $h[0], h[1], \dots, h[N_b - 1]$ )
- $N_b = q^3$  (panjang bin deskriptor dengan  $q$  adalah bilangan bulat)



Gambar 1 Skema Penghapusan Tekstur

### 2.1.2 Clustering K-Means

Clustering K-Means mengelompokkan piksel ke dalam cluster-cluster sehingga piksel dalam cluster yang sama memiliki tingkat kesamaan yang tinggi tetapi berbeda terhadap piksel pada cluster lain, dengan sebelumnya menentukan sejumlah K cluster terlebih dahulu. Bin deskriptor yang digunakan sebagai fitur vektor pada aplikasi dihitung menggunakan persamaan jarak *euclidean*

$$d(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_{p=1}^d (X_{i,p} - X_{j,p})^2} \quad (2)$$

$$= \|\vec{X}_i - \vec{X}_j\|$$

### 2.1.3 Melakukan Subdivisi

Subdivisi adalah prosedur *splitting* dengan membagi semua *region* yang memiliki ukuran piksel lebih dari bilangan tertentu ( $W_0$ ) menjadi potongan terpisah (*region* baru) dengan batasan jumlah  $W_0$  piksel. Pada aplikasi, prosedur ini dibantu dengan menggunakan *region growing* dan pola tertentu yang acak yakni

1. Pola 1: (r-1,c), (r+1,c), (r,c-1), (r,c+1)
2. Pola 2: (r,c-1), (r,c+1), (r-1,c), (r+1,c)
3. Pola 3: (r-1,c), (r,c-1), (r+1,c), (r,c+1)
4. Pola 4: (r,c-1), (r-1,c), (r,c+1), (r+1,c)

Dimana r,c adalah *seed* yang menjadi indeks piksel awal pada *region growing*.

### 2.1.4 Nilai Piksel diganti dengan Nilai Rata-rata Cluster

Nilai piksel diganti dengan nilai rata-rata cluster adalah mengganti nilai piksel dengan rata-rata nilai piksel pada cluster tersebut.

Setiap piksel memiliki keanggotaan cluster mana piksel tersebut berada. Pada setiap cluster dilakukan perhitungan rata-rata nilai piksel pada setiap anggota cluster, kemudian hasilnya menggantikan nilai piksel setiap anggota cluster tersebut. Langkah ini dilakukan pada setiap ruang warna dengan dimensi masing-masing ruang warna adalah memiliki 3 dimensi

### 2.1.5 Rata-rata Nilai Piksel antar Ruang Warna

Rata-rata nilai piksel antar ruang warna adalah melakukan perhitungan rata-rata nilai piksel antar ruang warna.

Hasil dari nilai piksel diganti dengan rata-rata cluster pada aplikasi, adalah sejumlah 10 ruang warna. Setiap ruang warna memiliki ukuran dan dimensi yang sama. Perhitungan rata-rata nilai piksel dilakukan antar ruang warna dengan dimensinya yang sama. Nilai piksel pada dimensi pertama dirata-rata dengan nilai piksel pada dimensi pertama antar ruang warna yang berbeda, begitu juga dengan dimensi kedua dan ketiga. Hasil dari langkah ini adalah citra hasil penghapusan tekstur.

## 2.2 K-Means dengan Batasan Ruang

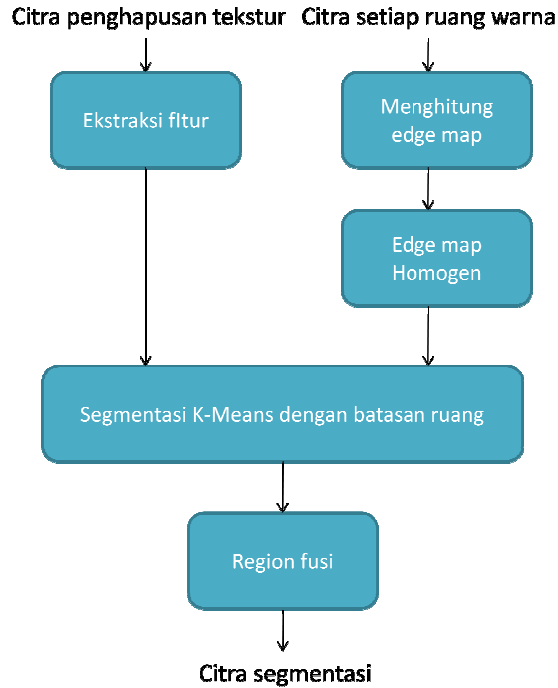
Segmentasi K-Means dengan batasan ruang dilakukan setelah proses penghapusan tekstur selesai. K-Means dengan batasan ruang menambahkan hubungan ruang pada proses segmentasi K-Means. Tujuannya untuk membantu proses pelabelan pada iterasi K-Means sehingga berhasil dalam menemukan solusi optimal yakni peta segmentasi yang akurat.

Proses segmentasi K-Means dengan batasan ruang, dapat dilihat skemanya pada Gambar 2, terdiri dari beberapa tahapan.

Pada awalnya melakukan ekstraksi fitur dari citra hasil penghapusan tekstur dan menghitung edge map dari citra setiap ruang warna.

Setelah itu dilakukan perhitungan untuk mendapatkan edge map yang homogen dengan berdasarkan pada nilai threshold tertentu.

Setelah proses ekstraksi fitur dan edge map homogen selesai, dilakukan proses segmentasi K-Means dengan batasan ruang. Terakhir dilakukan proses *region* fusi. *Region* fusi bertujuan untuk menggabungkan *region* yang kecil dengan *region* tetangganya.



**Gambar 2 Skema Segmentasi K-Means dengan Batasan Ruang**

### 2.2.1 Menghitung Edge Map

Menghitung edge map bertujuan untuk mendapatkan edge map dari suatu citra yang memiliki ruang warna berbeda-beda. Untuk menghitung edge map dilakukan dengan rumus berikut

$$D(h_{i-(\frac{N_0}{2}),j}, h_{i+(\frac{N_0}{2}),j}) + D(h_{i,j-(\frac{N_0}{2})}, h_{i,j+(\frac{N_0}{2})}) \quad (3)$$

Dimana

$h$  : Bin deskriptor : array bertipe float dari  $N_b$  ( $h[0], h[1], \dots, h[N_b - 1]$ )

$D$  : jarak L1 norm antar vektor bin deskriptor yang dihitung pada  $N_0 \times N_0$  overlapping window

Kemudian hasil dari edge map pada masing-masing ruang warna dirata-rata dan dinormalisasi (0-1).

### 2.2.2 Edge Map Homogen

Pada edge map diidentifikasi sekumpulan *connected piksel* yang edge potential dibawah threshold. Hasil identifikasi ini adalah Edge Map yang homogen dalam konteks tekstural.

Dapat dihitung dengan nilai piksel maksimal dikali threshold kemudian dibandingkan dengan nilai piksel edge map.

Sekumpulan piksel yang edge potensialnya dibawah threshold direpresentasikan dengan region warna. Region warna menunjukkan homogen dalam konteks tekstural. Region yang putih berkorespondensi terhadap sekumpulan piksel yang edge potensialnya di atas threshold. Region yang putih menunjukkan inhomogen dalam konteks tekstural

### 2.2.3 Ekstraksi Fitur

Ekstraksi fitur bertujuan untuk mendapatkan fitur vektor pada gambar hasil penghapusan tekstur, yang dibutuhkan pada segmentasi K-Means dengan batasan ruang.

Fitur deskriptor vektor didapat dengan memakai semua nilai warna dari setiap ruang warna. Hasilnya pada ekstraksi fitur ini didapatkan  $[n_1 \times n_1 \times 3 \times 10]$  dimensi fitur vektor dengan  $n_1 \times n_1$  adalah banyaknya piksel pada overlapping window

### 2.2.4 Segmentasi K-Means dengan Batasan Ruang

Segmentasi K-Means dengan batasan ruang menambahkan hubungan perbandingan edge map pada proses iterasi K-Means. Fitur yang digunakan adalah fitur deskriptor hasil ekstraksi fitur pada citra hasil penghapusan tekstur.

Berikut adalah langkah-langkah yang dilakukan untuk melakukan segmentasi K-Means dengan batasan ruang

1. Secara acak pilih  $K_1$  inialisasi pusat cluster  $c_1^{[1]}, \dots, c_{K_1}^{[1]}$

2. Pada langkah ke k tetapkan sampel  $x_m$  ke dalam cluster dengan pusat terdekat, yakni ke cluster  $i$  jika  $\|x_m - c_i^{[k]}\| < \|x_m - c_j^{[k]}\|, \forall j \neq i$
3.
  - a. Untuk setiap region  $R_i$  dari edge map
  - b. Cari cluster mayor dari sampel  $x_m \in R_i$ , yakni cluster  $i$  jika:
 
$$c_l^{[k]} = \arg \max_{x_m \in R_i} \sum_{1 \leq j \leq K, j \neq l} I \{ \|x_m - c_j^{[k]}\| < \|x_m - c_l^{[k]}\| \}$$
 Dimana  $I$  adalah fungsi indicator
4.  $c_i^{[k]}$  menunjukkan cluster ke  $i$  dengan sampel  $n_i$  setelah langkah 2 tentukan cluster baru dengan rata-rata sampel dalam cluster
5. Ulangi sampai konvergensi terpenuhi
6. Fusi region yang kecil (yakni sekumpulan *connected pixel* dari cluster yang ukurannya di bawah 300 piksel)

3	100098_320.ppm
4	101085_320.ppm
5	101087_320.ppm
6	102061_320.ppm
7	103041_320.ppm
8	103070_320.ppm
9	104022_320.ppm
10	105019_320.ppm
11	105025_320.ppm
12	105053_320.ppm
13	106020_320.ppm
14	106024_320.ppm
15	106025_320.ppm
16	108005_320.ppm
17	108041_320.ppm
18	108070_320.ppm
19	108073_320.ppm
20	108082_320.ppm
21	109034_320.ppm
22	109053_320.ppm
23	112082_320.ppm
24	113009_320.ppm
25	113016_320.ppm
26	113044_320.ppm
27	117054_320.ppm
28	118020_320.ppm
29	118035_320.ppm
30	119082_320.ppm

### 3. HASIL DAN PEMBAHASAN

Perangkat lunak Segmentasi Citra berhasil dibangun untuk mengimplementasikan metode Segmentasi Citra Menggunakan Penghapusan Tekstur dan K-Means dengan Batasan Ruang.

#### 3.1 Dataset, Skenario Uji Coba dan Perhitungan Akurasi

Dilakukan uji coba terhadap dataset citra tekstur Berkeley sejumlah 30 citra.

Data pengujian dapat dilihat pada tabel 1

**Tabel 1. Dataset Uji Coba**

No	Nama
1	100075_320.ppm
2	100080_320.ppm

Pengujian dilakukan dengan lima skenario. Uji coba pada skenario pertama dilakukan dengan mengubah nilai parameter nilai threshold pada edge map. Uji coba pada skenario kedua dilakukan dengan mengubah nilai jumlah kelas  $K$  untuk segmentasi K-Means dengan batasan ruang. Uji coba pada skenario ketiga dilakukan dengan mengubah jumlah *size window* untuk pengambilan fitur. Uji coba pada skenario keempat dilakukan dengan mengubah nilai *subdivision* dan uji coba pada skenario kelima dilakukan dengan mengubah nilai *size fuse*. Detail parameter dapat dilihat pada tabel 2, 3, 4, 5 dan 6

**Tabel 2. Parameter Uji Coba Skenario 1**

Parameter	Nilai					
	Percobaan Ke					
	1	2	3	4	5	6
Subdivision	5000	5000	5000	5000	5000	5000
Size Window	5	5	5	5	5	5
Size Fuse	600	600	600	600	600	600
K	9	9	9	9	9	9
Threshold	0.2	0.25	0.27	0.3	0.35	0.37

**Tabel 3. Parameter Uji Coba Skenario 2**

Parameter	Nilai					
	Percobaan Ke					
	1	2	3	4	5	6
Subdivision	5000	5000	5000	5000	5000	5000
Size Window	5	5	5	5	5	5
Size Fuse	600	600	600	600	600	600
K	4	5	6	7	8	10
Threshold	0.37	0.37	0.37	0.37	0.37	0.37

**Tabel 4. Parameter Uji Coba Skenario 3**

Parameter	Nilai			
	Percobaan Ke			
	1	2	3	4
Subdivision	5000	5000	5000	5000
Size Window	1	2	3	4
Size Fuse	600	600	600	600
K	9	9	9	9
Threshold	0.37	0.37	0.37	0.37

**Tabel 5. Parameter Uji Coba Skenario 4**

Parameter	Nilai		
	Percobaan Ke		
	1	2	3
Subdivision	6000	5000	4000
Size Window	5	5	5
Size Fuse	600	600	600
K	9	9	9
Threshold	0.37	0.37	0.37

**Tabel 6. Parameter Uji Coba Skenario 5**

Parameter	Nilai		
	Percobaan Ke		
	1	2	3
Subdivision	5000	5000	5000
Size Window	5	5	5
Size Fuse	600	500	400
K	9	9	9
Threshold	0.37	0.37	0.37

Evaluasi dilakukan dengan menggunakan perhitungan akurasi Probabilistic Rand Index.

Probabilistic Rand Index dihitung dengan menggunakan rumus

$$PRI = \frac{1}{k} \cdot \sum_{k=1}^k RI(\text{hasil Segmentasi}, \text{benchmark}(k)) \quad (4)$$

Dengan rumus untuk menghitung RI dapat ditulis sebagai berikut

$$RI = 1 - \frac{\left[ \frac{1}{2} (\sum_u n_u^2 + \sum_v n_v^2) - \sum_{u,v} n_{uv}^2 \right]}{N(N-1)/2} \quad (5)$$

### 3.2 Hasil Uji Coba

Hasil uji coba pada 30 gambar masukan pada skenario 1, 2, 3, 4 dan 5 dapat dilihat pada tabel 7, 8, 9, 10 dan 11 dengan nilai akurasi rata untuk masing-masing percobaan

**Tabel 7. Hasil Uji Coba Skenario 1**

No	Nilai Akurasi (%)					
	Percobaan Ke					
	1	2	3	4	5	6
	73.34	74.21	74.13	73.79	72.93	72.69

**Tabel 8. Hasil Uji Coba Skenario 2**

No	Nilai Akurasi (%)					
	Percobaan Ke					
	1	2	3	4	5	6
	71.78	72.16	72.81	72.42	72.32	73.62

**Tabel 9. Hasil Uji Coba Skenario 3**

No	Nilai Akurasi (%)			
	Percobaan Ke			
	1	2	3	4
	69.91	69.85	68.17	71.96

**Tabel 10. Hasil Uji Coba Skenario 4**

No	Nilai Akurasi (%)		
	Percobaan Ke		
	1	2	3
	72.81	72.69	73.71



**Tabel 11. Hasil Uji Coba Skenario 5**

No	Nilai Akurasi (%)		
	Percobaan Ke		
	1	2	3
	72.69	72.02	74.70

Hasil uji coba pada skenario ke 1 menunjukkan performa terbaik pada percobaan ke dua pada nilai threshold 0.25 dengan nilai rata-rata akurasi PRI adalah 74.21 %.

Kemudian pada skenario ke 2 menunjukkan performa terbaik pada percobaan ke 6 yaitu nilai K = 10 dengan nilai rata-rata akurasi PRI adalah 73.62 %.

Pada skenario ke 3 menunjukkan performa terbaik pada percobaan ke 4 yaitu jumlah *size window* bernilai 4 dengan nilai rata-rata akurasi PRI adalah 71.96 %

Pada skenario ke 4 menunjukkan performa terbaik pada percobaan ke 3 yaitu pada saat jumlah *subdivision* bernilai 4000 dengan nilai rata-rata akurasi PRI adalah 73.71 %

Pada skenario ke 5 menunjukkan performa terbaik pada percobaan ke 3 yaitu pada saat jumlah *size fuse* bernilai 400 dengan nilai rata-rata akurasi PRI adalah 74.70 %

Dari keseluruhan uji coba di dapat performa terbaik dengan nilai threshold 0.37, *subdivision* 5000, *size window* 5, *size fuse* 400 dan nilai K 9 dengan nilai rata-rata akurasi PRI adalah 74.70 %.

#### 4. KESIMPULAN

Beberapa kesimpulan yang dapat diambil adalah sebagai berikut:

1. Metode segmentasi citra menggunakan Penghapusan Tekstur dan K-Means dengan Batasan Ruang cukup baik untuk melakukan segmentasi tekstur dari suatu citra.
2. Nilai bin deskriptor dari citra tekstur berwarna dapat digunakan sebagai representasi fitur citra untuk penghapusan tekstur.
3. Nilai edge map yang *homogen* dari suatu citra dapat digunakan sebagai batasan ruang

dengan membantu segmentasi citra pada K-Means.

4. Performa segmentasi citra tekstur ini mampu mencapai 74,21% pada skenario 1 dengan menggunakan nilai *subdivision*, *size window*, *size fuse*, K dan threshold secara berturut-turut adalah 5000, 5, 600, 9, dan 0.2

#### 5. DAFTAR REFERENSI

- [1] Mignotte, M., 2007. Image denoising by averaging of piecewise constant simulations of image partitions. IEEE Trans. Image Process. 16(2), 523-533.
- [2] Mignotte, M., 2008. Segmentation by fusion of histogram-based k-means clusters in different color space. IEEE Trans. Image Process. 17(5), 780-787.
- [3] Mignotte, M., 2010. A de-texturing and spatially constrained K-means approach for image segmentation. Pattern Recognition Letters. 32, 352-367.
- [4] Gonzalez R.C., Woods R.E. 2008. Digital Image Processing, Third Edition, Prentice Hall.
- [5] R. Unnikrishnan, M. Hebert. 2005. Measures of Similarity. Seventh IEEE Workshop on Applications of Computer Vision, pp. 394-400.
- [6] Y. B. Chen. 2011. A robust fully automatic scheme for general image segmentation. Digital Signal Processing: A Review Journal 21 (1), pp. 87-99.
- [7] Verma, O.P., Hanmandlu, M Susan, S., Kulkarni, M., Jain, P.K.. 2011, "A simple single seeded region growing algorithm for color image segmentation using adaptive thresholding". Communication Systems and Network Technologies (CSNT), 2011 International Conference on, pp. 500-503.
- [8] Terrillon, Jean-Christophe, Akamatsu, Shigeru. 2000. Comparative Performance of different chrominance spaces for color segmentation. Proc. of the 12th Conf. on Vision Interface (VI '99, pp 180-187.
- [9] Y. B. Chen, Oscar T.-C. Chen. 2002. Semi-automatic image segmentation using dynamic direction prediction. Dept. of

Electrical Engineering, National Chug Cheng University, Chia-Yi, Taiwan, R.O.C.

- [10] K. Karsch, Q. He, Y. Duan. 2009. A fast, semi-automatic brain structure segmentation algorithm for magnetic resonance imaging. In Proc. of IEEE BIBM, November 2009, pp. 297–302
- [11] Y.B. Chen, O.T.-C. Chen. 2009. Image segmentation method using thresholds automatically determined from picture contents. Eurasip Journal on Image and Video Processing 2009, art. no. 140492
- [12] Y.B. Chen, O.T.-C. Chen. 2009. High-accuracy moving object extraction using background subtraction. ICIC Express Letters 3 (4), pp. 933-938
- [13] Haris, K., Efstratiadis, S.N., Maglaveras, N., Katsaggelos, A.K.. 1998. Hybrid image segmentation using watersheds and fast region merging. IEEE Trans. Image Process. 7(12), 1684–1699

# RBTC

Ruang Baca Jurusan Teknik Informatika - ITS