

Segmentasi Citra Berbasis *Vektor Quantization*

Rosida Vivin Nahari

Fakultas Teknik, Universitas Trunojoyo Madura
Jl. Raya Telang PO BOX 2. Kamal. Bangkalan
rosida_vn@yahoo.com

Aery Rachmad, Haryanto, Riza Alfita, Mirza Pramudia

Fakultas Teknik, Universitas Trunojoyo Madura
Jl. Raya Telang PO BOX 2. Kamal. Bangkalan
aeri_r@yahoo.com, haryanto_utm@yahoo.com, yogya_001@yahoo.co.id,
mirza_ub@yahoo.com

Abstrak— Pada penelitian ini, Metode *vector quantization* yang dioptimasi menggunakan metode K-mean diimplementasikan untuk mengelompokkan region pada gambar 2 dimensi dengan ekstraksi fitur warna RGB dan fitur warna YcbCr. Fitur warna tersebut kemudian dijadikan sebagai vektor input (3 dimensi). Jumlah *cluster* yang diujicobakan adalah $N = 20$ dan $N=3$. Dari hasil pengujian dapat ditunjukkan bahwa semakin besar ukuran citra dan sebaran histogram yang semakin merata, waktu iterasi yang dibutuhkan semakin lama. Sedangkan semakin banyak jumlah *cluster*, akan menyebabkan semakin banyak jumlah iterasi dan semakin lama waktu iterasi yang dibutuhkan. Namun demikian terdapat beberapa pengujian yang mengalami perbedaan antara satu percobaan dengan percobaan yang lain pada citra dengan parameter-parameter yang sama. Hal ini disebabkan oleh peletakan nilai RGB k yang pertama secara acak dari nilai nol sampai dengan dua. Dari lima kali pengujian terhadap citra dengan parameter - parameter yang sama, sebagian besar diperoleh tingkat perubahan jumlah iterasi yang kurang dari 15%

Kata Kunci—Segmentasi Citra, *Vector Quantization*, *cluster*.

I. PENDAHULUAN

Perkembangan sistem visi komputer saat ini telah banyak dimanfaatkan dalam membantu manusia dalam proses pengenalan atau deteksi obyek. Proses pengenalan suatu obyek merupakan pekerjaan yang cukup sulit sehingga dalam proses tersebut disarankan untuk mengimplementasikan teknologi

computer vision untuk mengambil peranan dalam mengenali obyek 2 dimensi.

Dalam proses pengenalan obyek atau deteksi obyek diperlukan suatu pemisahan bagian atau segmen tertentu dalam citra yang akurat, proses pemisahan tersebut dikenal sebagai proses segmentasi. Beberapa metode telah dikembangkan diantaranya adalah *vector quantization* untuk pengenalan suara [4][5] dan beberapa pengenalan pola lainnya [6][7]. Proses pengenalan segmen merupakan salah satu kunci dalam mendapatkan suatu hasil pengenalan atau deteksi yang akurat. Segmentasi membagi suatu citra menjadi bagian-bagian atau segmen yang lebih sederhana dan bermakna sehingga dapat dilakukan analisis lebih lanjut. Secara umum pendekatan segmentasi citra yang sering digunakan adalah melalui pendekatan intensitas, pendekatan warna dan pendekatan bentuk.

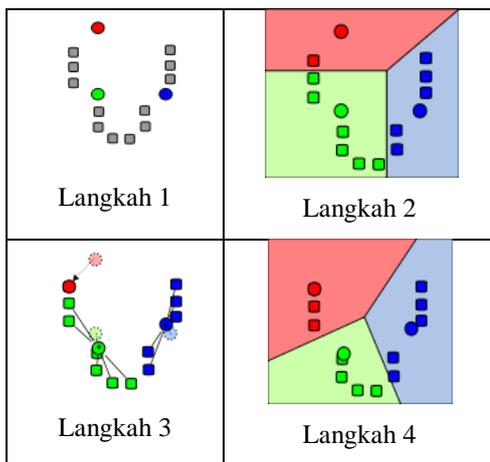
Segmentasi warna merupakan pemisahan segmen dalam suatu citra berdasarkan ekstraksi fitur warna yang terkandung dalam citra. Metode yang bisa dilakukan untuk segmentasi warna adalah metode clustering dengan *vector quantization* yang dioptimasi dengan menggunakan algoritma K-Means[8][9].

Metode *vector quatization* sudah banyak dimanfaatkan dalam berbagai pengolahan video dan kompresi data diantaranya: *bin* *video* [1], *Daala* [2], Microsoft Video, game dan masih banyak yang lainnya.

II. URAIAN PENELITIAN

A. K-Means

K-Means adalah teknik yang cukup sederhana dan cepat dalam pekerjaan pengelompokan data (clustering). Prinsip utama dari teknik ini adalah menyusun **k** buah prototipe/pusat massa (*centroid*)/rata-rata (*mean*) dari sekumpulan data berdimensi **n**. Teknik ini mensyaratkan nilai **k** sudah diketahui sebelumnya (*a priori*). Algoritma *k-means* dimulai dengan pembentukan prototipe cluster di awal kemudian secara iteratif prototipe cluster ini diperbaiki hingga konvergen (tidak terjadi perubahan yang signifikan pada prototipe cluster). Perubahan ini diukur menggunakan fungsi objektif yang umumnya didefinisikan sebagai jumlah atau rata-rata jarak tiap item data dengan pusat massa kelompoknya. Secara lebih detail algoritma *k-means* adalah seperti berikut :



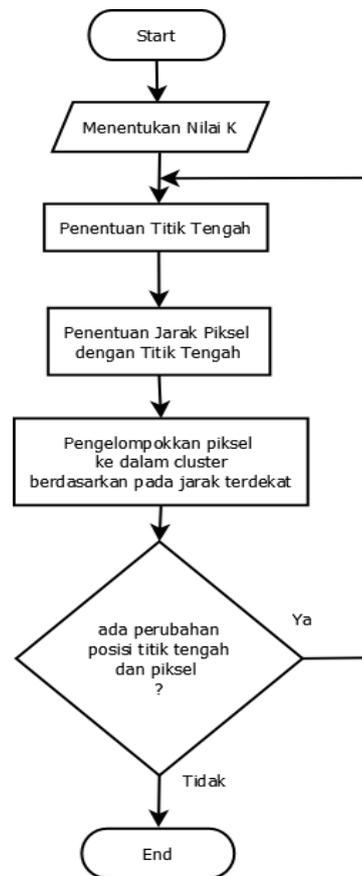
Gambar 1. Pemodelan algoritma K-Means

Langkah 1 : *k* insial dari “means” , pada kasus di atas menggunakan $k = 3$ diseleksi secara random dari dataset. (simbol berwarna).

Langkah 2 : Cluster *k* dihasilkan dari setiap percobaan dengan nilai rata-rata terdekat. Partisi tersebut merepresentasikan diagram voronoi hasil pembentukan dari nilai rata-rata.

Langkah 3 : Centroid dari masing-masing cluster *k* menjadi nilai *means* yang baru.

Langkah 4 : Langkah 2 dan 3 diulang sampai konvergensi tercapai.



Gambar 2. Flowchart *K-Means Clustering*.

B. Vector Quantization

Vector Quantization (VQ) pada awalnya merupakan teknik kompresi data lossy. Saat ini desain VQ terkendala pada implementasi multi-dimensi yang membutuhkan teknik berbasis *training sequence* untuk mengestimasi waktu komputasi ketika proses pelatihan pembentukan *codebook vector*.

VQ memetakan vektor *k-dimensional* di ruang vektor R^k ke dalam sekumpulan vektor $Y = \{y_i; i = 1, 2, \dots, N\}$. Masing-masing vektor y_i disebut *code vector* atau *codeword* dan kumpulan dari semua *codeword* dinamakan *codebook*. Penghubung setiap *codeword* y_i adalah region tetangganya yang disebut *voronoi region*. Elemen dari VQ tersebut bisa didefinisikan dalam bentuk formula sebagai berikut:

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|\}$$

$$\text{For all } j \neq i \} \tag{1}$$

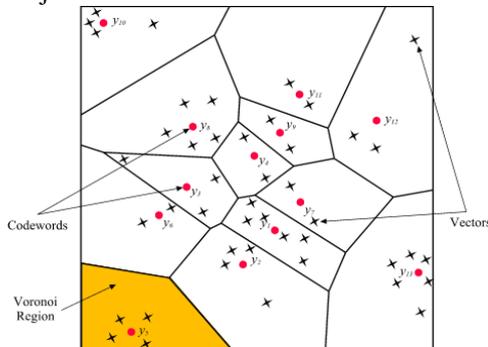
Himpunan partisi region Voronoi pada ruang R^k :

$$\bigcup_{i=1}^N V_i = R^k \tag{2}$$

dan

$$\bigcap_{i=1}^N V_i = \phi \text{ for all } i \neq j \tag{3}$$

Sebagai contoh kasus kita mengambil vektor di ruang 2 dimensi. Gambar 2 menunjukkan beberapa vektor dalam ruang. *Codeword* direpresentasikan sebagai hubungan dari setiap *cluster* vektor. Setiap *codeword* berada di wilayah voronoi-nya sendiri. Region diilustrasikan pada gambar 2 sebagai wilayah yang dipisahkan oleh garis imajiner.

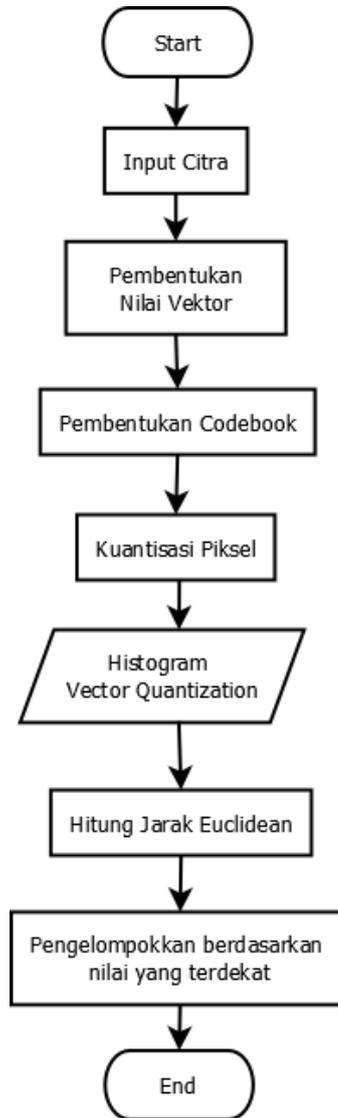


Gambar 3. Model VQ 2 - Dimensi

Representasi codeword ditentukan sebagai jarak euclidean terdekat dari vektor input.

$$d(x, y_i) = \sqrt{\sum_{j=1}^k (x_j - y_{ij})^2} \tag{4}$$

Dimana x_j adalah komponen vektor input ke j , dan y_{ij} adalah komponen codeword y_i ke j .



Gambar 4. Flowchart Vector Quantization with K-means Clustering

B.1. Jarak Euclidean

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini berkaitan dengan Teorema Phytagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

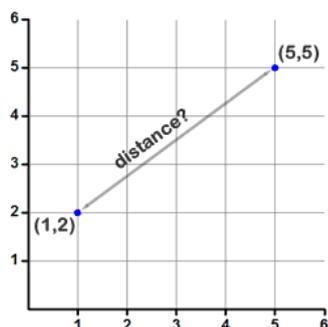
B.1.1. Pada 1 dimensi

Semisal ingin menghitung jarak Euclidean

1 dimensi. Titip pertama adalah 4, titik kedua adalah -10. Caranya adalah kurangkan -10 dengan 4. sehingga menghasilkan -14. Cari nilai absolut dari nilai -14 dengan cara memangkatkannya sehingga mendapat nilai 196. Kemudian diakarkan sehingga mendapatkan nilai 14. Sehingga jarak euclidean dari 2 titik tersebut adalah 14.

B.1.2. Pada 2 dimensi

Misalkan titik pertama mempunyai koordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, (5-1,5-2) sehingga menjadi (4,3). Kemudian pangkatkan masing-masing sehingga memperoleh (16,9). Kemudian tambahkan semuanya sehingga memperoleh nilai $16+9 = 25$. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak euclideannya adalah 5.



Gambar 5. Model perhitungan jarak euclidean (koordinat jarak)

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

B.2. Pembentukan *codebook*

Pemilihan *codeword* terbaik yang bisa mewakili vektor input sangat sulit dilakukan. Namun, ada beberapa algoritma yang dapat mengoptimalkan pembentukan *codebook* diantaranya algoritma LBG (Linde-Buzo-Gray) dan algoritma K-Means.

Secara garis besar algoritma pembentukan *codebook* adalah sebagai berikut:

1. Tentukan jumlah *codeword* N, atau ukuran dari *codebook*.
2. Pilih N *codeword* secara acak, dan digunakan sebagai inisial dari *codebook*. Inisialisasi *codeword* bisa didapatkan secara acak dari kumpulan vektor input.

3. Ukuran jarak *Euclidean* digunakan untuk mengelompokkan vektor disekitar *codeword* masing-masing yaitu dengan cara mengambil vektor input dan menemukan jarak *Euclidean* antara vektor input dan *codeword*. Vektor input menjadi anggota kelompok *codeword* dengan nilai jarak minimum.
4. Hitung himpunan *codeword* yang baru dengan cara memperoleh nilai rata-rata masing-masing *cluster*. Tambahkan komponen setiap vektor dan dibagi dengan jumlah vektor dalam *cluster*.

$$y_i = \frac{1}{m} \sum_{j=1}^m x_{ij} \quad (6)$$

Dimana i adalah komponen setiap vektor (x, y, z, ... arah), m adalah jumlah vektor dalam *cluster*.

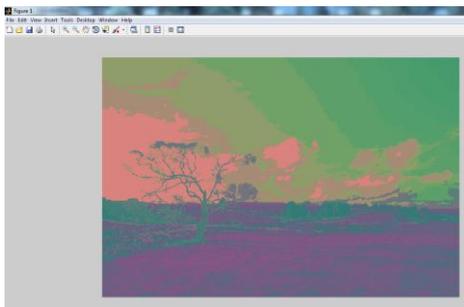
5. Ulangi langkah 2 dan 3 sampai *codeword* tidak berubah atau perubahan *codeword* kecil.

III. HASIL PENELITIAN

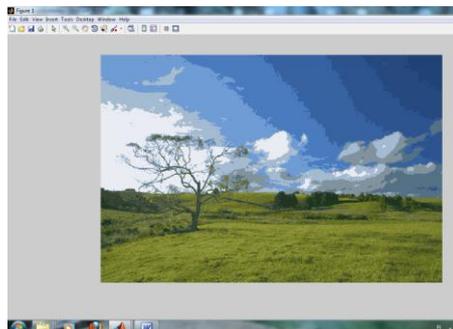
Pada penelitian ini, dilakukan uji coba dengan mengelompokkan region pada gambar 2 dimensi dengan ekstrasi fitur warna RGB dan fitur warna YcbCr. Fitur warna tersebut kemudian dijadikan sebagai vektor input (3 dimensi). Jumlah *cluster* yang diujicobakan adalah N = 20 dan N=3.



Gambar 6. Citra Asli



Gambar 7. Citra hasil *clustering YCbCr* dengan N=20

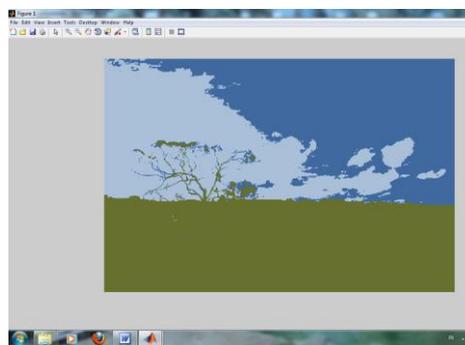


Gambar 10. Citra hasil *clustering RGB* dengan N=20

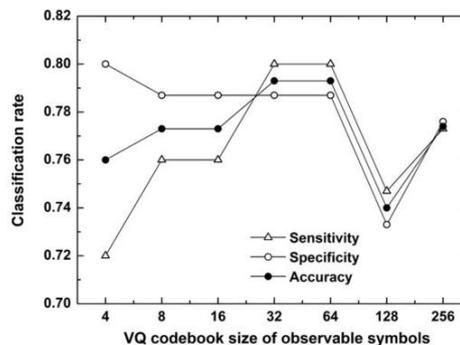
```

File Edit Debug Parallel Desktop Window
Shortcuts How to Add What's New
Workspace
New to MATLAB? Watch this Video, see Demos, or
codebook =
      R      G      B
92.4616 159.3951 105.8594
118.0570 115.0008 126.7394
141.8002 157.1539 105.6931
51.5651 122.5434 119.7153
88.1814 104.5276 127.4119
92.9676 125.8831 122.1790
113.6605 92.2753 130.5187
81.5774 159.4863 106.9934
112.2519 161.9930 103.0335
138.0066 86.9435 132.0631
126.7385 86.1219 132.2550
166.6965 140.4192 117.0025
77.9787 157.6332 108.6325
123.8033 93.0043 130.8163
217.5023 130.9656 125.6949
102.1004 97.8156 129.2654
73.1555 156.9267 108.7143
71.6561 112.6216 124.0202
66.8017 125.8551 120.8849
68.9399 154.8450 109.2188
    
```

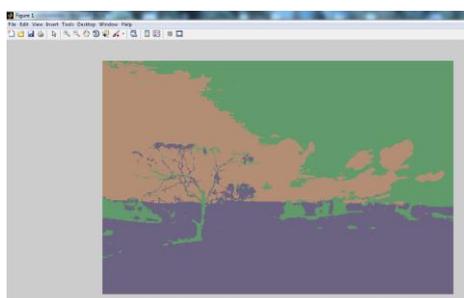
Gambar 8. Hasil codebook dari fitur warna RGB



Gambar 11. Citra hasil *clustering RGB* dengan N=3



Gambar 12. Perbandingan tingkat keberhasilan klasifikasi VQ berdasarkan jumlah *codebook*



Gambar 9. Citra hasil *clustering YCbCr* dengan N=3

IV. SIMPULAN

Dari hasil pengujian dapat ditunjukkan bahwa semakin besar ukuran citra dan sebaran histogram yang semakin merata, waktu iterasi yang dibutuhkan semakin lama. Sedangkan semakin banyak jumlah cluster, akan menyebabkan semakin banyak jumlah

iterasi dan semakin lama waktu iterasi yang dibutuhkan. Hasil penelitian menunjukkan bahwa pengelompokan piksel menggunakan vector quantization yang telah dioptimasi menggunakan metode k-means dapat membantu proses segmentasi citra.

DAFTAR PUSTAKA

- [1] "Bink video". Book of Wisdom. 2009-12-27. Retrieved 2013-03-16.
- [2] Valin, JM. (October 2012). Pyramid Vector Quantization for Video Coding. [IETF](#). I-D draft-valin-videocodec-pvq-00. Retrieved 2013-12-17.
- [3] "Vorbis I Specification". Xiph.org. 2007-03-09. Retrieved 2007-03-09.
- [4] Burton, D. K.; Shore, J. E.; Buck, J. T. (1983). "A generalization of isolated word recognition using vector quantization". IEEE International Conference on Acoustics Speech and Signal Processing ICASSP: 1021–1024. doi:10.1109/ICASSP.1983.1171915.
- [5] Soong, F.; A. Rosenberg; L. Rabiner; B. Juang (1985). "A vector Quantization approach to Speaker Recognition". IEEE Proceedings International Conference on Acoustics, Speech and Signal Processing ICASSP **1**: 387–390. doi:10.1109/ICASSP.1985.1168412.
- [6] H. Jegou; M. Douze; C. Schmid (2011). "Product Quantization for Nearest Neighbor Search". Transactions on Pattern Analysis and Machine Intelligence **33** (1): 117–128. doi:10.1109/TPAMI.2010.57.
- [7] Faundez-Zanuy, Marcos (2007). "On-line signature recognition based on VQ-DTW". Pattern Recognition **40** (3): 981–992. doi:10.1016/j.patcog.2006.06.007.
- [8] Faundez-Zanuy, Marcos; Juan Manuel Pascual-Gaspar (2011). "Efficient On-line signature recognition based on Multi-section VQ". Pattern Analysis and Applications **14** (1): 37–45. doi:10.1007/s10044-010-0176-8.
- [9] Gray, R.M. (1984). "Vector Quantization". IEEE ASSP Magazine **1** (2): 4–29. doi:10.1109/massp.1984.1162229