

# Perbandingan Penggunaan Mean Lokal, Median Lokal dan Invarians Statistik Koefisien DCT dalam Perancangan *Image Hashing*

Kuntadi Widiyoko<sup>1</sup>, Iwan Setyawan<sup>2</sup>

Program Studi Teknik Elektro,  
Fakultas Teknik Elektronika dan Komputer,  
Universitas Kristen SatyaWacana, Salatiga  
<sup>1</sup>612007009@student.uksw.edu, <sup>2</sup>iwan.setyawan@ieee.org

## Ringkasan

Pada tulisan ini dirancang sistem *image hashing* dengan berbagai tipe. Tipe 1 merupakan sistem *image hashing* yang cara kerjanya membandingkan nilai *mean* citra dengan *mean* tiap blok dalam citra tersebut. Tipe 2 adalah sistem *image hashing* yang membandingkan nilai median citra dengan median tiap blok. Tipe 3 merupakan sistem *image hashing* yang membandingkan nilai *mean* sebuah blok dengan blok selanjutnya dalam 1 citra. Tipe 4 adalah pengembangan tipe 3, yaitu dengan terlebih dahulu mengacak blok-blok citra. Tipe 5 adalah *image hashing* yang membandingkan nilai-nilai koefisien DCT tiap blok. Kinerja kelima tipe tersebut diuji dari sisi ketahanannya dalam menghadapi gangguan seperti kompresi JPEG, *sharpening*, *histogram equalization* dan *low pass filtering*.

**Kata kunci:** *image hashing*, DCT, median, *mean*, citra digital

## 1. Pendahuluan

Perkembangan teknologi informasi saat ini membuat banyak data multimedia yang disimpan dalam bentuk digital untuk mempermudah pengaksesannya. Namun, hal ini juga membuat merebaknya aplikasi-aplikasi untuk mempermudah manipulasi data digital [1]. Diperlukan suatu proses otentikasi untuk melindungi data-data tersebut. Proses otentikasi akan memilah apakah sebuah data multimedia mengalami manipulasi atau tidak. Salah satu bentuk otentikasi adalah dengan cara membandingkan nilai *hash* data yang akan diuji keasliannya dengan data aslinya. Untuk menghasilkan *hash* tersebut banyak digunakan algoritma-algoritma seperti MD5 atau SHA1. *Hash* ini akan berubah bila terdapat perubahan sekecil apa pun pada data multimedia tersebut. Perubahan tersebut bisa terjadi karena proses yang sederhana seperti kompresi atau pemfilteran [2]. Tetapi dalam beberapa aplikasi seperti *watermarking* dan pengindeksan data multimedia, *hash* yang dihasilkan pada proses *image hashing* yang dilakukan harus memiliki ketahanan (*robustness*) terhadap gangguan sederhana [3].

Tulisan ini berisi hal-hal sebagai berikut. Pada bagian 2 dibahas tentang *mean* dan median, pada bagian 3 dibahas tentang *discrete cosine transform*, bagian 4 berisi tentang cara kerja sistem yang dirancang, pada bagian 5 berisi tentang percobaan dan hasilnya sedangkan pada bagian 6 berisi tentang kesimpulan dan saran.

## 2. Mean dan Median

Mean atau rata-rata adalah ukuran pemusatan lokasi yang banyak digunakan, dan paling mudah untuk dihitung [4]. Median adalah rerata posisi, karena median adalah nilai bilangan di tengah pada sekumpulan data. Median ditentukan dengan cara menyusun deretan bilangan dalam suatu data, baik menaik maupun menurun, lalu menentukan bilangan yang ada di tengah [5].

Secara matematis *mean* ( $\mu$ ) dan median (M) dari sekumpulan data dengan jumlah elemen sebanyak  $n$ ,  $X = \{x_1, x_2, \dots, x_n\}$  dicari menggunakan rumus:

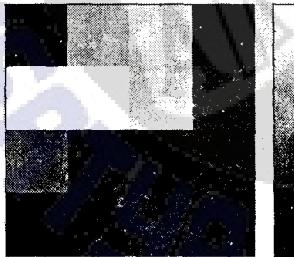
$$\mu = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad (1)$$

$$M = \begin{cases} \frac{x_n + 1}{2} & \text{untuk } n \text{ ganjil} \\ \frac{x_{n/2} + x_{(n/2)+1}}{2} & \text{untuk } n \text{ genap} \end{cases} \quad (2)$$

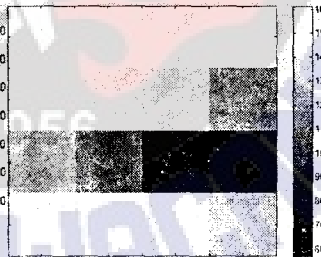


(a) "Barbara"

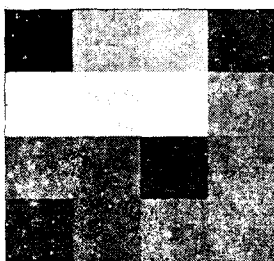
(b) "Boat"



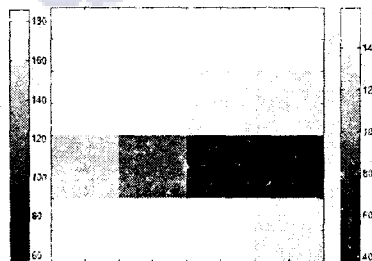
(c) Mean lokal "Barbara"



(d) Mean lokal "Boat"



(e) Median lokal "Barbara"



(f) Median lokal "Boat"

Gambar 1. Perbandingan *mean* dan median lokal dua buah citra

Setiap citra *grayscale* memiliki nilai *mean* dan median lokal yang berbeda tergantung dari konten lokal citra tersebut. Nilai *mean* dan median lokal yang berbeda ini bisa dimanfaatkan untuk menghasilkan deretan nilai *hash* yang mempunyai sifat unik dan berbeda antara satu citra dengan citra yang lain.

### 3. Discrete Cosine Transform

*Discrete Cosine Transform* (DCT) adalah sebuah transformasi linear yang sering dilakukan pada pengolahan sinyal. Selain untuk pengolahan sinyal, DCT juga dimanfaatkan untuk pengolahan citra pada ranah frekuensi seperti untuk kompresi citra pada JPEG dan pengenalan pola [6]. Transformasi ini akan mengubah citra dari domain spasial menjadi citra pada domain frekuensi [7].

Pada tulisan ini transformasi DCT 2 dimensi terhadap sebuah blok citra dengan ukuran  $M \times N$  dilakukan menurut persamaan berikut.

$$B(p, q) = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A(m, n) \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad (3)$$

untuk  $0 \leq p \leq M-1$  dan  $0 \leq q \leq N-1$ , dan:

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & p \text{ lainnya} \end{cases}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & q \text{ lainnya} \end{cases}$$

Pada persamaan (3),  $A(m, n)$  adalah citra masukan sedangkan  $B(p, q)$  merupakan koefisien DCT ke  $(p, q)$ .

### 4. Cara Kerja Sistem

Pada tulisan ini cara kerja sistem *image hashing* yang dirancang dibagi menjadi 2 bagian. Bagian pertama adalah perancangan pada ranah spasial menggunakan *mean* dan median lokal. Bagian kedua adalah perancangan pada ranah frekuensi menggunakan *statical invariance* koefisien DCT.

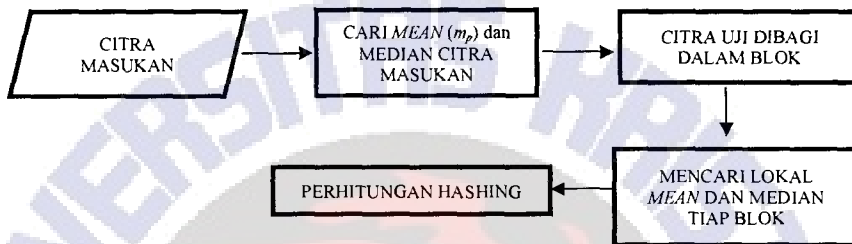
#### 4.1. Lokal Mean dan Median

Pertama-tama dicari nilai *mean* ( $m_p$ ) dan nilai median dari citra masukan. Selanjutnya citra masukan yang berukuran  $M \times N$  akan dibagi menjadi  $P$  blok dengan ukuran yang sama. Pada tiap blok akan dicari nilai *mean* atau median.

Pada *image hashing* tipe 1 *hash* citra diperoleh dengan membandingkan nilai *mean* tiap blok ini dengan nilai  $m_p$ . Bila nilai *mean* sebuah blok lebih besar daripada nilai  $m_p$  akan diperoleh nilai bit *hash* sama dengan 1, sedangkan bila nilai *mean* sebuah blok lebih kecil daripada nilai  $m_p$  akan diperoleh bit *hash* sama dengan 0.

Pada *image hashing* tipe 2 *hash* citra diperoleh dengan cara membandingkan nilai median tiap blok citra dengan nilai median citra. Bila nilai median suatu blok lebih besar daripada nilai median citra akan diperoleh bit *hash* sama dengan 1, sedangkan bila nilai median suatu blok lebih kecil dari pada nilai median citra akan diperoleh bit *hash* sama dengan 0.

Pada *image hashing* tipe 3 *hash* citra diperoleh dengan membandingkan nilai *mean* sebuah blok dengan nilai *mean* blok selanjutnya. Bila nilai *mean* blok tersebut lebih besar daripada nilai *mean* blok selanjutnya, akan diperoleh bit *hash* sama dengan 1, sedangkan bila nilai *mean* tersebut lebih kecil akan diperoleh bit *hash* sama dengan 0. *Image hashing* tipe 4 memiliki prinsip seperti *image hashing* tipe 3, akan tetapi pada tipe ini blok-blok citra terlebih dahulu diacak sebelum dilakukan proses perbandingan. Pada tulisan ini, keempat tipe yang sudah dijelaskan diatas akan menghasilkan nilai *hash* biner sepanjang  $P$  bit.



Gambar 2. Cara kerja *image Hashing* dengan lokal *mean* dan median.

4.2. Sifat *Statistical Invariance Koefisien DCT*

*Image hashing* tipe 5 yang dibahas pada tulisan ini memanfaatkan koefisien DCT pada citra. Citra masukan berukuran  $M \times N$  yang berupa *grayscale* selanjutnya dibagi menjadi blok-blok  $b_i$  dengan  $i = 1, 2, 3, \dots, P$ . Selanjutnya transformasi DCT diterapkan pada tiap blok tersebut. Dari masing-masing blok, 9 koefisien AC pertama (diambil menggunakan aturan zig-zag seperti pada Gambar 2) dimanfaatkan untuk menyusun *hash*. Koefisien DCT yang diambil dari blok ke- $i$  disimbolkan dengan  $s_{ij}$ , dengan  $j = 1, 2, \dots, 9$ . Kumpulan koefisien DCT tersebut disimpan dalam matriks  $x = (s_{1j}, s_{2j}, \dots, s_{pj})$ .

1	2	3	4	5	6	7
9	10	11	12	13	14	15
17	18	19	20	21	22	23
25	26	27	28	29	30	31
33	34	35	36	37	38	39
41	42	43	44	45	46	47

Gambar 3. Aturan pengambilan koefisien DCT dengan zig-zag order.

Langkah selanjutnya adalah menerapkan Persamaan *Maximum Likelihood* (ML) [8] terhadap  $x_j$ , yaitu elemen-elemen matriks  $x$ , sebagai berikut.

$$f_{ML} = \frac{\varphi(1+\frac{1}{v}) + \log_{10}(v)}{v^2} + \frac{1}{v^2} \log\left(\frac{1}{n} \sum_{j=1}^n |x_j|\right) - \frac{\sum_{j=1}^n |x_j|^v \log(x_j)^v}{v \sum_{j=1}^n |x_j|^v} \tag{4}$$

dengan :

$$\varphi(\tau) = -\gamma + \int_0^1 (1 - t^{\tau-1})(1 - t)^{-1} dt \tag{5}$$

Pada Persamaan (4),  $\gamma = 0.577$  dan  $v = 1$ . Dari hasil perhitungan ini akan didapat 9 nilai *statistical invariance*  $v_j$ . Masing-masing nilai  $v_j$  akan dibandingkan satu dengan yang lain sehingga mendapatkan *hash* berupa angka biner sepanjang 36 bit.



Gambar 4. Cara kerja *image hashing* dengan DCT

## 5. Percobaan dan Hasil

Pada percobaan ini digunakan citra *grayscale* dengan ukuran  $M \times N = 256 \times 256$  pixel. Pada percobaan tipe 1-4 citra masukan dibagi menjadi  $P = 1024$  blok dengan tiap blok berukuran  $8 \times 8$  pixel. Untuk tipe 5 citra masukan dibagi menjadi  $P = 256$  blok, masing-masing berukuran  $16 \times 16$  pixel. Kinerja sistem *image hashing* yang dinilai dengan menghitung rata-rata *hamming distance* antara dua buah *hash*. Perhitungan *hamming distance*  $H_d$  antara dua buah *hash*  $h$  dan  $h'$  dilakukan sebagai berikut.

$$H_d = \frac{1}{N} \sum_{i=1}^N \delta(h_i, h'_i) \quad (6)$$

dengan

$$\delta(h_i, h'_i) = \begin{cases} 1, & h_i \neq h'_i \\ 0, & h_i = h'_i \end{cases} \quad (7)$$

Hasil percobaan untuk menguji ketahanan *hash* terhadap gangguan ditampilkan pada Tabel 1-4. Pada gambar 5, ditampilkan grafik perbandingan hasil *hamming distance* antar *hash* citra yang berlainan dengan menggunakan kelima tipe *image hashing* yang dirancang. Hasil yang ditunjukkan pada gambar ini menunjukkan seberapa unik *hash* yang dihasilkan.

Tabel 1 menunjukkan hasil rata-rata *hamming distance* kelima tipe *image hashing* yang digunakan ketika mendapat gangguan berupa *histogram equalization* dan *sharpening*. Terlihat tipe 4 lebih tahan menghadapi *histogram equalization* dengan rata-rata kesalahan sebesar 0,022, sedangkan untuk gangguan berupa *sharpening* tipe 1 lebih unggul dengan rata-rata kesalahan sebesar 0,025.

Tabel 1. Rata-rata *hamming distance* tiap tipe dengan gangguan *histogram equalization* dan *sharpening*

TIPE	HISTEQ	SHARPENING
1	0,071	0,025
2	0,024	0,034
3	0,027	0,065
4	0,022	0,033
5	0,111	0,191

Tabel 2 merupakan hasil rata-rata *hamming distance* kelima tipe *image hashing* yang digunakan ketika mendapat gangguan berupa filter Gaussian dengan variasi sigma antara 0.05-0.5. Dari percobaan tersebut dapat disimpulkan bahwa Tipe 1 memiliki ketahanan yang paling baik terhadap gangguan berupa filter Gaussian.

Tabel 2. Rata-rata *hamming distance* tiap tipe dengan gangguan berupa filter Gaussian

Tipe	SIGMA									
	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
1	0	0	0	0	0	0,00006	0,00039	0,00115	0,00195	0,00287
2	0	0	0	0	0	0,00020	0,00061	0,00131	0,00291	0,00398
3	0	0	0	0	0	0,00057	0,00201	0,00490	0,00932	0,01287
4	0	0	0	0	0	0,00014	0,00065	0,00195	0,00330	0,00469
5	0	0	0	0	0	0,00167	0,00556	0,00167	0,00556	0,02389

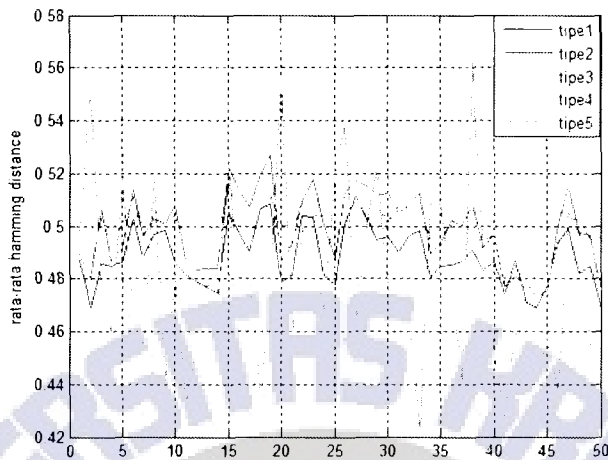
Tabel 3 merupakan hasil rata-rata *hamming distance* kelima tipe *image hashing* ketika mendapatkan gangguan berupa kompresi JPEG dengan nilai *Q-factor* antara 25-90. Nilai *Q-factor* makin kecil menunjukkan tingkat kompresi yang makin tinggi. Hasil percobaan menunjukkan bahwa tipe 1 memiliki ketahanan paling tinggi terhadap kompresi JPEG.

Tabel 3. Rata-rata *hamming distance* tiap tipe dengan gangguan berupa kompresi JPEG

Tipe \ <i>Q-factor</i>	1	2	3	4	5
25	0,008	0,021	0,098	0,019	0,051
30	0,005	0,017	0,086	0,016	0,038
35	0,004	0,016	0,075	0,014	0,039
40	0,005	0,014	0,068	0,013	0,033
45	0,004	0,013	0,062	0,011	0,025
50	0,004	0,014	0,057	0,009	0,022
55	0,003	0,012	0,053	0,009	0,019
60	0,003	0,012	0,049	0,008	0,021
65	0,002	0,011	0,043	0,007	0,019
70	0,003	0,011	0,039	0,007	0,016
75	0,001	0,010	0,033	0,005	0,013
80	0,001	0,009	0,025	0,004	0,013
85	0,001	0,008	0,021	0,003	0,010
90	0,001	0,007	0,014	0,002	0,009

Gambar 5 merupakan perbandingan rata-rata *hamming distance* bila nilai *hash* dua buah citra uji yang berbeda dibandingkan nilai. Suatu sistem *hash* yang baik seharusnya menghasilkan *hamming distance* sekitar 0.5, yang menunjukkan bahwa kurang lebih 50% bit *hash* memiliki perbedaan. Dari Gambar 5 dapat dilihat bahwa hal ini dipenuhi oleh tipe 1, 2, 3 dan 4. Sementara itu, tipe 5 menghasilkan *hamming distance* yang kecil jika terdapat kemiripan antara dua citra yang dibandingkan.

Perlu diperhatikan pada tulisan ini antara *image hashing* tipe 1-4 dengan tipe 5 memiliki perbedaan jumlah bit *hash* yang dihasilkan. Perbedaan ini akan berpengaruh pada hasil rata-rata *hamming distance*.



Gambar 5. Perbandingan hasil rata-rata *hamming distance* antar citra yang berbeda

## 5. Kesimpulan dan Saran

*Image hashing* dapat disusun dengan memanfaatkan *mean* dan median lokal suatu citra. Pemanfaatan *mean* dan median lokal memberikan tingkat ketahanan yang tinggi dalam menghadapi gangguan berupa operasi *histogram equalization*, *sharpening*, filter Gaussian dan kompresi JPEG. Pada percobaan didapat hasil tipe 1 memiliki ketahanan tertinggi terhadap *sharpening*, kompresi JPEG dan filter Gaussian, sedangkan tipe 4 memiliki ketahanan tertinggi terhadap operasi *histogram equalization*.

Selain memanfaatkan lokal mean dan median, pada tulisan ini dilakukan juga pembuatan *image hashing* dalam ranah frekuensi dengan memanfaatkan sifat invarians statistik koefisien DCT. Sistem *image hashing* dengan memanfaatkan koefisien DCT memiliki kinerja yang cukup baik terhadap serangan, meskipun bukan merupakan yang memiliki kinerja terbaik. Sistem ini peka terhadap konten citra, yaitu dua citra dengan konten yang mirip akan menghasilkan *hash* dengan perbedaan yang lebih kecil dibandingkan dengan citra dengan konten yang berbeda.

Perlu dilakukan penelitian dan pengembangan lebih lanjut terhadap sistem *image hashing* tipe 5 (menggunakan koefisien DCT). Penelitian perlu difokuskan pada efek penambahan jumlah koefisien DCT yang digunakan dalam *image hashing* terhadap kinerja sistem dalam menghadapi gangguan.

## 6. Daftar Pustaka

- [1] M. Wu dan B. Liu, *Multimedia Data Hiding*, Springer-Verlag, 2002.
- [2] F. Liu, H.Y. Leung, L.M. Cheng dan X.Y. Ji, "Content-based image hashing using wave atoms," *Chinese Physical Society and IOP*, 2012.
- [3] A. Swaminathan, Y. Mao, dan M. Wu, *Image Hashing Resilient To Geometric And Filtering Operations*, ECE Department, University of Maryland, College Park, U.S.A.
- [4] P. Waxmann, *Business Mathematics and Statistics*, 3rd edition, Prentice Hall, Victoria, 1993.
- [5] Wijanarto, "Image Retrieval Berdasarkan Properti Statistik Histogram," *Jurnal Techno Science*, 2009.

- [6] S. Syarif, N. Harun, M. Tola, M.W. Tjaronge, Z.B. Hasanuddin, Z. Jamid, dan R.Z. Asgar, "Sistem Cerdas Deteksi Citra Dengan Metode Discrete Cosine Transform," *PROSIDING*, 2012.
- [7] E.Y. Hidayat dan F.N. Adnan, "Hybrid DWT-DCT Watermarking Citra Digital Menggunakan Metode Comparison-Based Correlation Embedding," *Semantik*, Semarang, 2012.
- [8] F.X. Yu, Z.M. Lu, Y.Q. Lei dan Y.G. Wang, "Robust Image Hashing Based on Statistical Invariance of DCT Coefficients," *Journal of Information Hiding and Multimedia Signal Processing*, 2010.

