

Pencocokan Citra Berbasis *Scale Invariant Feature Transform* (SIFT) menggunakan *Arc Cosinus*

Agus Setiyawan, dan Ruri Suko Basuki, M.Kom
Teknik Informatika – S1, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Imam Bonjol No. 207, Semarang 50131 Indonesia

Tujuan penelitian ini mengekstrak fitur untuk keperluan *image matching*. Ekstraksi fitur tidak terpengaruh perubahan sudut pandang, skala maupun perubahan rotasi, menjadi hal yang riskan untuk mendapatkan fitur khusus yang menjadi pembeda dengan citra lain. Dalam algoritma SIFT, *keypoint* yang dihasilkan tidak terpengaruh oleh sudut pandang, skala maupun rotasi. Metode *Arc Cosine* yang merupakan *inverse trigonometry* digunakan untuk pencocokkan citra dengan berbagai parameter seperti perubahan sudut pandang, skala, dan rotasi, dengan memanfaatkan hasil *keypoint* dari SIFT. Pengujian juga dilakukan dengan 4 kompresi berbeda yaitu PNG, BMP, JPEG dan PGM. Hasil pengujian dari 100 citra training dan 1500 citra testing dengan beberapa parameter tersebut didapatkan hasil rata-rata citra cocok untuk perubahan sudut pandang 33,498 %, perubahan skala 60,236 % dan perubahan rotasi 82,779 %, untuk kompresi terbaik adalah PNG dengan 59,078%, diikuti BMP, PGM dan JPEG masing-masing 59,05 %, 58,12 % dan 58,51 %. Dari hasil tersebut, pencocokkan citra dengan *Arc Cosine* mempunyai keunggulan dalam mencocokkan perubahan rotasi pada citra.

Kata kunci- Ekstraksi fitur, pencocokan citra, SIFT, *Arc Cosine*, data training, data testing.

I. PENDAHULUAN

PENCOCOKAN citra merupakan hal yang mudah bagi manusia baik dalam keadaan cahaya yang cukup maupun dengan cahaya yang redup. Namun, dalam pandangan komputer, pekerjaan itu menjadi pekerjaan yang sulit, karena citra tidak memiliki informasi *semantic*. Sehingga perlu pendekatan tertentu agar pencocokan citra dapat dijadikan basis beberapa aplikasi dalam *computer vision* seperti pengenalan citra (*image recognition*).

Efektifitas pencocokkan dapat dipengaruhi beberapa parameter seperti, sudut pandang, perubahan skala atau ukuran, perubahan rotasi maupun perubahan intensitas cahaya pada citra, yang mana nilai efektifitasnya akan optimal jika berbanding lurus terhadap pengenalan citra. Tahapan – tahapan sebelum masuk ke dalam proses pencocokkan citra, terlebih dahulu citra harus terdeteksi, kemudian mencari ciri khusus yang membedakan dengan citra – citra lainnya. Pada proses deteksi, terkadang sistem tidak mengerti atau tidak bisa mendeteksi citra, ketika citra yang dimasukkan mengalami perubahan ukuran, pencahayaan, rotasi maupun sudut pengambilan citra.

Pendeteksian dan pencocokkan citra merupakan salah satu penelitian yang menarik, pencocokkan citra menjadi dasar dari pengembangan penelitian lanjut seperti pengenalan objek atau

recognition. Banyak informasi yang terdapat pada sebuah citra, seperti kedalaman warna, pencahayaan dan juga kerapatan titik-titik gambar atau pixel. Dengan memanfaatkan informasi – informasi tersebut, computer dapat mengenali melalui ekstraksi fitur masing-masing informasi.

Pada tahun 1999, David Lowe pertama kali mengenalkan metode ekstraksi fitur yang disebut *Scale Invariant Feature Transform* (SIFT). Metode ini dapat memberikan fitur yang tidak dipengaruhi oleh perubahan ukuran objek, adanya translasi atau rotasi pada objek, serta sedikit terpengaruh terhadap perbedaan intensitas cahaya pada objek yang dikenali. Pada penelitian kali ini, SIFT akan digunakan dalam menemukan titik utama (*keypoint*) pada citra yang selanjutnya akan dicocokkan. *Arc Cosine* adalah salah satu fungsi dari trigonometri inverse (*Invers Trigonometri*), atau sering disebut *Invers Cosinus* (*Invers Cosine*).

Adanya beberapa parameter seperti pencahayaan, rotasi, sudut pengambilan citra, dan perubahan skala pada citra, mengakibatkan citra menjadi sulit untuk dikenali sistem. Sehingga diperlukan pendekatan tertentu untuk meningkatkan kemampuan pengenalan citra. Dengan algoritma *Scale Invariant Feature Transform* (SIFT), akan didapatkan *keypoint* *keypoint* yang unik antara satu dengan yang lain dari sebuah citra. Namun objek dari citra tersebut belum dapat dicocokkan..

II. TINJAUAN PUSTAKA

A. Citra

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut [18], hal tersebut diilustrasikan pada gambar 2.1, teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (Red, Green, Blue - RGB).

B. SIFT

Scale Invariant Feature Transform (SIFT) adalah sebuah algoritma dalam *computer vision* untuk mendeteksi dan mendeskripsikan fitur lokal dalam gambar. Algoritma ini dipublikasikan oleh David Lowe pada tahun 1999. Dengan menggunakan SIFT ini, suatu citra akan di ubah menjadi vector fitur lokal yang kemudian digunakan sebagai pendekatan dalam mendeteksi maupun mengenali objek yang dimaksud melalui titik-titik point atau *keypoint*

Algoritma yang digunakan pada metode SIFT antara lain :

1. *Scale Space Extreme Detection*

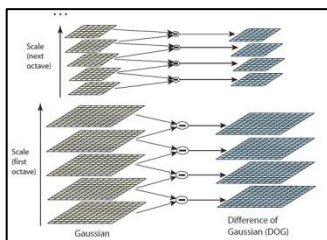
Dimisalkan ada sebuah pohon dan ingin dipisahkan antara daun, batang, ranting, dll dengan sengaja. Untuk memisahkan tersebut harus diberikan rincian rincian masing masing agar tidak tertukar. Pada hal ini digunakan Gaussian Blur (dibuktikan secara matematik, dan didukung beberapa alasan yang menguatkan).

SIFT membutuhkan ruang skala ke tingkat berikutnya. Dimisalkan diambil sebuah gambar asli dan menghasilkan semakin kabur keluar gambar, kemudian ukurannya diubah menjadi setengah dari ukuran semula dan terus berulang sampai 4 octave atau 4 tahap.

Fungsi yang digunakan adalah gaussian blur :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y).....(1)$$

Dengan menggunakan ruang skala, dihitung perbedaan antar 2 ruang skala berturut – turut. Kemudian menggunakan *different of gaussian* untuk mengeliminasi.



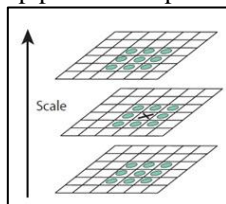
Gambar 1 Eliminasi menggunakan *Different of Gaussian*

Pada gambar tersebut hanya diterapkan untuk satu octave. Hal yang sama dilakukan untuk semua octave dan akan menghasilkan gambar DoG dari berbagai ukuran. Konsep pendekatannya dua gambar berturut – turut dalam satu octave dipetik dan dikurangi dari yang lain. Kemudian pasangan selanjutnya berturut – turut diambil, dan proses berulang terus. Hal ini dilakukan untuk semua octave.

2. *Keypoint localization*

Setelah menghasilkan ruang skala kemudian digunakan untuk menghitung *different of Gaussian* dan dilanjutkan menghitung *laplacian of Gaussian* dengan hasilnya adalah menghasilkan titik titik atau *point*. [3][6][9]

Selanjutnya adalah mencari *maxima / minima* dalam gambar *Dog (Different of Gaussian)*. Langkahnya adalah menemukan titik *maxima / minima* yang kasar dulu. Caranya dengan melewati setiap pixel dan diperiksa tetangganya.



Gambar 2 : Menentukan Titik *maxima* dan *minima* untuk *keypoint*

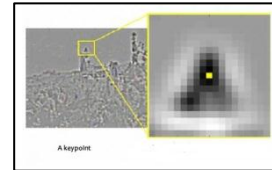
Pada gambar diatas x menandai pixel, dan lingkaran hijau menandai tetangga. Dengan cara ini, total 26 cek tetangga yang dibuat. X adalah sebagai “*key point*” jika itu adalah yang

terbesar atau paling tidak dari 26 tetangga sekitar titik tersebut. Sedangkan untuk yang putih atau kosong tidak cukup diperbandingkan jadi hanya dilewatkan saja, secara matematis, dapat dirumuskan seperti ini :

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x.....(2)$$

3. *Orientation Assigment*

Setelah mendapatkan *keypoint* hasil seleksi, Hal berikutnya adalah menetapkan orientasi kepada setiap *keypoint*. Langkahnya adalah dengan mengumpulkan arah gradient dan besaran sekitar setiap *keypoint*. Kemudian diketahui orientasi yang paling menonjol di wilayah *keypoint* tersebut. [6]



Gambar 3 : *Keypoint* dan Orientasinya

Untuk mendapatkan orientasi tersebut digunakan fungsi berikut :

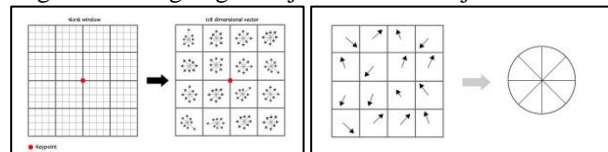
$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))) \quad (3)$$

4. *Keypoint Descriptor*

Tahap ini menjadi langkah akhir dari algoritma SIFT setelah memiliki orientasi dan *keypoint*, langkah selanjutnya membuat *descriptor* atau sidik jari untuk setiap *keypoint*. [6][9] yang berfungsi untuk mengidentifikasi setiap *keypoint*.

Untuk mempermudah perhitungan harus dihasilkan *keypoint* yang sangat unik. Untuk melakukan ini, ambil jendela area 16 atau 4 x 4 di sekitar *keypoint* tersebut. Area 16 ini masing masing kotak dibagi lagi menjadi enam belas jendela 4 x 4.



Gambar 4 : Penentuan *Descriptor Keypoint*

Setiap orientasi gradient di kisaran 0-44 derajat menambah bin pertama. 45-89 menambah bin berikutnya. Dan jumlahnya akan ditambahkan ke bin tergantung pada besarnya gradient. Gradient yang jauh dari *keypoint* akan menambah nilai yang lebih kecil untuk histogram. Fungsinya adalah :

$$f(\theta, x, y) = |J(x, y)| \delta(\theta - J(x, y)).....(4)$$

C. *Arc Cosinus*

Invers Cosinus mempunyai fungsi :
Y=arc cos X.

Dengan mengembalikan invers cosinus tiap nilai derajat X. Jika dalam deretan(*array*) inverse cosinus dapat di proses dengan fungsi berikut :

$$\cos^{-1}(z) = -i \log[z + i(1 - z^2)^{1/2}] \dots \dots \dots (5)$$

III. METODE PENELITIAN

A. Tahapan Penelitian



Gambar 6 : Tahapan Penelitian

Data Training & Data Testing

Data Training dan Data Testing disini adalah citra yang akan diproses sistem. Data ini berjumlah 126 data dari 9 objek benda, masing masing benda diambil dari 4 sudut pengambilan yang berbeda. Diambil secara langsung dengan jarak pengambilan 30 – 70 cm dengan menggunakan Kamera Telepon Genggam (*handphone*) LG Optimus Black P970. Data dari citra disini sudah mengalami proses sedemikian rupa untuk menguji kemampuan sistem, dimulai dari pengambilan sudut gambar, perubahan skala ukuran ,dan juga perubahan rotasi dengan perputaran tertentu.

Konversi RGB ke Grayscale

Pada tahap ini data training maupun data testing akan dirubah dari RGB(*Red Green Blue*) menjadi *Grayscale*. Tujuan dari Konversi ini adalah untuk mereduksi citra dari 24 bit menjadi 8 bit untuk mempermudah diolah oleh sistem.

Ekstraksi Fitur dengan SIFT

Ekstraksi Fitur dengan SIFT dilakukan proses mencari dan menentukan *keypoint*, kemudia menentukan *descriptor* atau sidk jari dari masing masing *keypoint* tersebut. *Descriptor* inilah yang akan menjadi dasar dalam pencocokan citra dalam penelitian ini.

Pencocokan Dengan ArcCos

Disini *keypoint* yang didapatkan dari data training maupun data testing dihitung menggunakan fungsi *Arc Cosine*, kemudian di cocokkan satu per satu untuk mendapatkan brapa *keypoint* yang sama antara data training dan data testing.

Hasil Pencocokan

Hasil akhir dari proses ini adalah brapa *keypoint* yang sama antara data training dengan data testing dan brapa persen kemiripan kecocokan antar keduanya.

B. Evaluasi

Evaluasi disini dilihat melalui statistika hasil dari pengujian pencocokan 2 citra dengan beberapa parameter seperti sudut pengambilan (*point of view*), perubahan ukuran(skala) dan perubahan rotasi. Tujuannya untuk mengukur seberapa besar kemampuan *Arc Cosine* dalam mencocokkan citra dengan

beberapa paramater tersebut dan juga beberapa jenis compresi image.

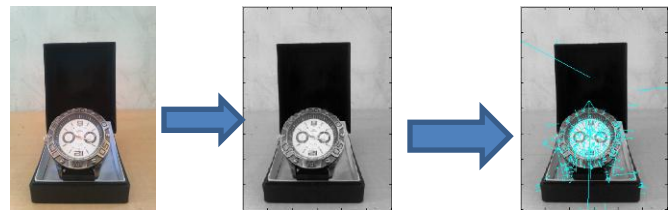
IV. HASIL PENELITIAN DAN PEMBAHASAN

A. Akuisisi Data

Data yang disiapkan pada penelitian kali ini adalah data training dan data testing. Pengambilan data training maupun data testing menggunakan telephone genggam (*handphone*) Lg Optimus Black P970. Jarak pengambilan data citra sekitar 30 – 60 cm. Data citra yang diambil berupa JPEG dan masuk kategori RGB

B. Pre Processing

Setelah citra training dikumpulkan, kemudian mulai dilakukan pemrosesan pada citra. Citra yang masih RGB di rubah menjadi Grayscale. Kemudian dilakukan pemrosesan SIFT untuk dicari dan ditentukan *keypoint* – *keypoint* yang terdapat pada citra tersebut.



Gambar 7 Proses Pre Processing sampai mendapatkan Keypoint

Hasil dari ekstraksi SIFT adalah didapatkan *keypoint location*, *keypoint descriptor*, *image*. *keypoint location* yaitu yang menjelaskan lokasi dimana *keypoint* tersebut berada, mempunyai 4 variabel yaitu *row*(baris), *coloum*(kolom), *scale*(ukuran, panjang *keypoint*), *orientation*(arah *keypoint* tersebut dalam radian). *Keypoint descriptor* yaitu sidik jari atau ciri khusus dari masing masing *keypoint* yang membedakan dengan *keypoint* lainnya. *Descriptor* terdiri dari 128 ciri khusus yang diambil dari kotak 4x4 sekitar *keypoint* tersebut. *Image* adalah citra yang di proses.

Tabel 1 : contoh *Keypoint Location*

No	Baris	Kolom	Skala	Orientasi
1	194,52	55,04	15,23	0,372
2	194,52	55,04	15,23	-0,837
3	34,06	47,73	11,9	2,192
4	138,44	21,09	9,93	-0,682
...

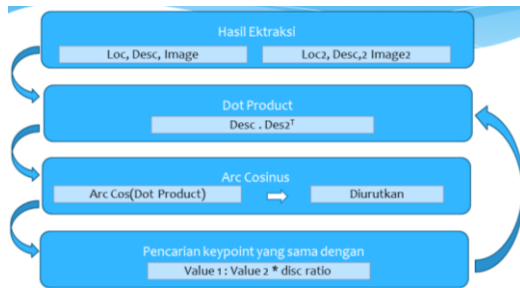
Tabel 2 : Contoh *Keypoint Descriptor*

1	0	0	0	0	...
2	0,003933	0,001966	0,021631	0,26744	...
3	0	0	0	0	...
4	0,076606	0	0	0,009821	...

5	0	0	0,233425	0,170655	...
...

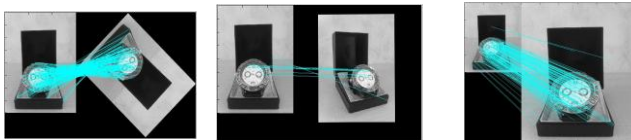
C. Arc Cosinus

Setelah didapatkan *keypoint location*, *keypoint descriptor*, dan *image* untuk citra training maupun testing, selanjutnya di cocokkan dengan *Arc Cosine*, langkahnya adalah :



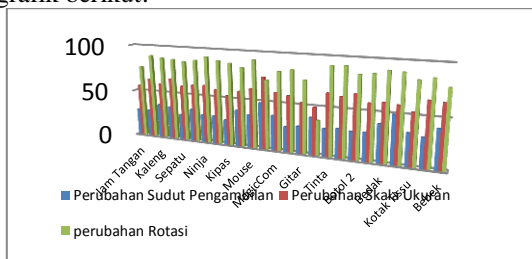
Gambar 8 : Tahapan Pencocokan

Setelah melalui tahap pada diagram di atas jika didapatkan *keypoint* yang sama kemudia digambarkan garis untuk melihat letak kesamaan *keypoint* tersebut.



Gambar 9 : Hasil Pencocokan

Setelah dilakukan pengujian terhadap semua data training maupun data testing maka didapatkan hasil seperti disajikan pada grafik berikut.



Gambar 10 : Grafik Hasil Pencocokan

dalam pengujian beberapa compresi citra mendapatkan hasil sebagai berikut :

Tabel 3 : Hasil Pengujian Compresi

No	Compresi	Sudut Pengambilan	Skala	Rotasi	Rata - rata
1	PNG	33,396	60,293	83,546	59,07833
2	BMP	33,381	60,191	83,592	59,05467
3	PGM	33,761	60,294	82,067	58,70733
4	JPEG	33,454	60,168	81,913	58,51167
Rata - rata		33,498	60,2365	82,7795	

V. KESIMPULAN/RINGKASAN

A. Kesimpulan

Pencocokan Citra Berbasis Scale Invariant Feature Transform (SIFT) menggunakan *Arc Cosinus*, menghasilkan rata rata kecocokan citra berdasarkan parameter Perubahan Sudut pengambilan Citra 33,498 %, Perubahan Ukuran (*Scala*) Citra 60,237 %, Perubahan Rotasi Citra 82,78 %. Dengan data hasil pengujian tersebut *Arc Cosinus* dengan tingkat baik dalam mencocokkan citra yang terkena perubahan rotasi, sedangkan untuk Perubahan Ukuran (*Scala*) dan Sudut Pengambilan Citra masih dalam tingkat kurang baik. Untuk compresi terbaik adalah compresi PNG dengan 59,078 %, diikuti BMP, PGM dan JPEG masing masing 59,05 %, 58,12 % dan 58,51 %

Terdapat beberapa faktor yang mempengaruhi keberhasilan sistem dalam mencocokkan citra, seperti latar belakang, kerapatan citra, kualitas citra dari masing masing citra masukan. Faktor ketelitian sistem dalam mendeteksi dan mencocokkan *keypoint* juga berpengaruh, sedikit perubahan *keypoint* bisa dianggap tidak cocok oleh sistem.

B. Saran

Untuk menghasilkan tingkat akurasi pencocokan yang lebih baik dalam pengambilan citra masukan diambil dengan tingkat kerapatan citra yang tinggi, tempat pengambilan dengan cahaya yang stabil.

Untuk parameter pengujian yang masih kurang baik seperti Sudut Pengambilan Citra, maupun Perubahan Skala pada citra bisa dicoba dengan metode lain, seperti Arc Sinus, *Euclidean Distance* maupun metode lainnya.

Penelitian ini juga bisa dikembangkan untuk ke tahap Pengenalan Citra (*Recognition*)

DAFTAR PUSTAKA

- [1] Sepritahara. 2012. Pembuatan Perangkat Lunak Pengenalan Wajah Menggunakan Principal Component Analysis, Depok.
- [2] Grosso, Enrico. Face Identification by SIFT-based Complete Graph Topology. Sassari, Italy
- [3] Lowe, D.G. 1999. Object Recognition from Local Scale-Invariant Features, Canada, pp. 1,2,3
- [4] <http://blog.uin-malang.ac.id/alan/2010/10/18/sidik-jari-fingerprint-dan-pengertian-biometric/> diakses tanggal 6 april 2013
- [5] Rakhman A, Rommy. 2010. Analisis Penggunaan Scale Invariant Feature Transform Sebagai Metode Ekstraksi Fitur Pada Pengenalan Jenis Kendaraan, Indonesia, pp. 5,8,9,10
- [6] <http://www.globalsecurity.org/security/systems/biometrics.htm>
- [7] <http://blog.uin-malang.ac.id/alan/2010/10/18/sidik-jari-fingerprint-dan-pengertian-biometric/>
- [8] Muja, Marius and David G. Lowe Fast Matching of Binary Features Laboratory for Computational Intelligence
- [9] Srivastava, Vikram Goyal, Prashant An Efficient Image Identification Algorithm using Scale Invariant Feature Detection
- [10] David G. Lowe, 2004. "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*.
- [11] Seo, Naotashi and David A. Schug. *Image Matching Using Scale Invariant Feature Transform (SIFT)*.
- [12] <http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/>
- [13] <http://note.sonots.com/SciSoftware/SIFT.html#x01604e7>
- [14] http://en.wikipedia.org/wiki/Inverse_trigonometric_functions
- [15] <http://www.scribd.com/doc/26241515/Pengolahan-Citra-Digital>
- [16] Baumberg, Adam. "Reliable Feature Matching Across Widely Separated Views"
- [17] <http://www.cs.cmu.edu/~schneide/tut5/node12.html>
- [18] <http://www.vlfeat.org/api/sift.html>