Deteksi Kendaraan pada Citra Udara Beresolusi Sangat Tinggi di Area Perkotaan dengan Menggunakan Metode Ekstraksi *Oriented* FAST *And Rotated* BRIEF

Diagnosa Fenomena, Darlis Herumurti, Joko Lianto Buliali
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: darlis@if.its.ac.id, joko@cs.its.ac.id, d.fenomena@live.com

***Abstrak*—Data mengenai lalu lintas mempunyai peranan penting dalam perencanaan tata kota, seperti untuk perencanaan jalan dan perkiraan tingkat pencemaran udara yang disebabkan oleh lalu lintas kendaraan di kota. Berbagai upaya telah dilakukan untuk mendeteksi kepadatan lalu lintas, salah satunya deteksi kendaraan menggunakan CCTV.**

Namun, penggunaan CCTV hanya efektif untuk deteksi kendaraan pada ruas jalan yang relatif terbatas. Oleh karena itu, dibutuhkan suatu implementasi algoritma yang secara otomatis dapat mendeteksi kendaraan melalui citra udara.

Terdapat beberapa metode yang dapat digunakan untuk mendeteksi objek, beberapa diantaranya adalah *template matching*, klasifikasi, dan *feature matching*. Pada tugas akhir ini, menggunakan metode *Template Matching* dengan menggunakan persamaan korelasi, *Haar cascade classification* dan *Feature Matching* dengan menggunakan ekstraksi fitur ORB.

Nilai *recall* dan *precision* tertinggi dihasilkan oleh metode *Feature Matching* dengan MSER dan ORB masing-masing bernilai 100% dan 75%.

Kata Kunci—citra udara, deteksi mobil, ekstraksi fitur

I. PENDAHULUAN

KEMACETAN merupakan permasalahan yang umum terjadi dan banyak terjadi di kota-kota besar yang pada gilirannya mengakibatkan kota menjadi tidak efisien dan bisa mengakibatkan kerugian ekonomi yang tidak sedikit. Hal ini disebabkan oleh beberapa permasalahan, yaitu rasio infrastruktur transportasi dengan luas lahan, jaringan yang tidak memadai, tata ruang yang tak terkendali dan pertumbuhan kendaraan yang sangat tinggi. Untuk memecahkan masalah ini diperlukan langkah atas dasar kajian dan langkah yang pernah dilakukan kota lain.

Salah satu yang dapat digunakan sebagai bahan kajian yaitu data lalu lintas. Data mengenai lalu lintas mempunyai peranan penting dalam perencanaan tata kota dan tata ruang, seperti untuk perencanaan jalan dan perkiraan tingkat pencemaran udara yang disebabkan oleh lalu lintas kendaraan di kota. Berbagai upaya telah dilakukan, salah satunya adalah deteksi kendaraan dengan menggunakan kamera televisi sirkuit tertutup atau CCTV.

Namun, penggunaan CCTV hanya efektif untuk deteksi kendaraan pada ruas jalan yang relatif terbatas. Terutama kota yang memiliki arsitektur yang cukup kompleks, contohnya memiliki banyak tikungan. Hal ini dapat diatasi dengan memanfaatkan citra udara untuk mendapatkan data lalu lintas yang dibutuhkan yang diambil menggunakan balon udara maupun pesawat tanpa awak (UAV).

Karakteristik citra udara, yang menampilkan lanskap tampak atas, dapat membantu memperluas jangkauan pengawasan satu daerah. Penggunaan citra udara juga lebih sesuai dengan kondisi geografis Indonesia yang terdiri dari kepulauan. Selain itu, citra udara juga menghasilkan gambar yang lebih detail dan tidak terkena awan jika dibandingkan dengan citra satelit.

Oleh karena itu, dibutuhkan suatu implementasi algoritma yang secara otomatis dapat mendeteksi kendaraan melalui citra udara. Dari implementasi algoritma ini, diharapkan dapat secara efektif mendukung analisis dalam perencanaan perkotaan terkait dengan lalu lintas.

II. METODE PENELITIAN

A. Desain Sistem

Bagian ini merupakan desain sistem secara umum dari sistem sesuai deteksi mobil dengan menggunakan beberapa proses. Proses-proses yang terlibat di dalam implementasi sistem ini meliputi tahap *grayscale conversion*, ekstraksi fitur, pencocokan fitur (*feature matching*), pencocokan *template* dan klasifikasi. Masing-masing proses tersebut dapat dilakukan secara paralel.

Alur metode secara umum dapat dilihat pada Gambar 1.

B. Deteksi menggunakan Haar Cascade Classification

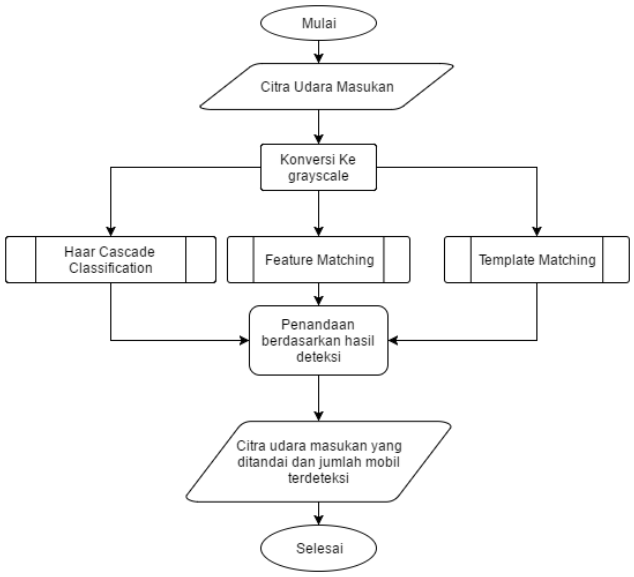
Proses *Haar-Cascade Classification* terdiri dari tiga tahapan, yaitu, tahap ekstraksi informasi sampel positif dan sampel negatif, tahap latihan (*training*), dan terakhir tahap klasifikasi.

Tahap *training* menggunakan metode *Adaboost* [1] untuk memilih fitur *Haar* dan melatih *classifier*. Pada Gambar 2, tampak 25 dari 527 sampel positif dan 25 dari 502 sampel negatif yang digunakan dalam metode ini.

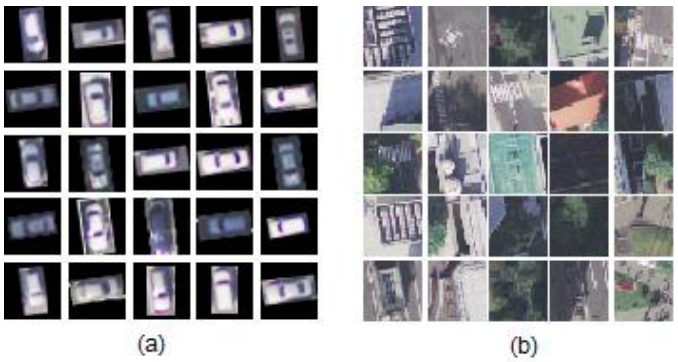
Untuk mempercepat perhitungan fitur *Haar*, gambar diubah terlebih dahulu menjadi citra integral. Kemudian, *Adaboost* akan memilih fitur *Haar*.

Selanjutnya, pada tahap klasifikasi dilakukan pelabelan yang ditentukan oleh jendela pada area tertentu. Jika label

menghasilkan nilai negatif, maka tahap klasifikasi dinyatakan selesai dan lokasi *window* dipindahkan ke lokasi berikutnya. Namun, jika label memberikan hasil positif, maka klasifikasi dilanjutkan ke tahap selanjutnya [2]



Gambar 1. Diagram alir sistem secara umum



Gambar 2. (a) Citra sampel positif. (b) Citra sampel negatif

C. Deteksi menggunakan Template Matching

Template matching merupakan metode untuk mencari area dalam gambar (*I*) yang sesuai atau mirip dengan gambar *template* (*T*). Metode ini digunakan ketika suatu gambar tidak memiliki fitur yang kuat. Untuk mencari area yang sesuai, dilakukan teknik *sliding window*, yaitu dengan menggeser *template T* sebanyak 1 piksel sampai menutupi seluruh gambar *I* [3].

Sambil melakukan pergeseran, nilai kecocokan area gambar masukan yang tertutup *window* dengan *template* akan dihitung menggunakan (1) sehingga menghasilkan matriks kecocokan ***R*.**

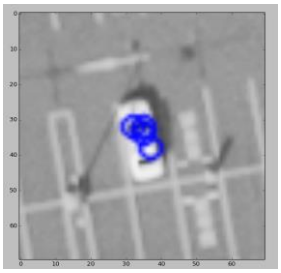
$$R(x,y) = \sum_{x',y'} T(x',y').I(x+x',y+y') \qquad (1)$$

Dimana *R(x,y)* merupakan nilai kecocokan matriks *R* pada indeks *(x,y)*. Sedangkan *T(x',y')* merupakan nilai keabuan pada gambar *template T* dan *I(x+x',y+y')* merupakan nilai keabuan pada gambar *I* Selanjutnya akan dilakukan *thresholding* pada matriks *R*. Jika nilai *R(x,y)* lebih besar daripada nilai *t*, maka *R(x,y)* dianggap cocok.

D. Deteksi menggunakan Feature Matching

Feature Matching atau pencocokan fitur digunakan untuk mencari titik fitur (*keypoint*) dalam gambar input yang sesuai dengan contoh gambar yang lebih kecil (gambar *template*). Metode pencocokan fitur ini dapat digunakan ketika gambar yang diproses mengalami tranformasi linear dan apabila gambar *template* memiliki fitur yang kuat

Langkah pertama dalam melakukan pencocokan fitur adalah dengan mencari titik fitur dari setiap gambar. Metode pencarian fitur dilakukan dengan memanfaatkan deteksi *keypoint* dari *Maximally Stable Extremal Region (MSER)* [4] dan *Oriented FAST and Rotated BRIEF(ORB)* [5]. Metode pencarian *keypoint* pada MSER dilakukan berdasarkan *extremal region*, sedangkan metode pencarian pada ORB dilakukan berdasarkan titik pojok atau disebut juga dengan *corner detection*.



Gambar 3. Salah satu hasil pencarian titik fitur

Pencarian titik fitur dilakukan pada *template* dan citra masukan seperti tampak pada Gambar 3. Berdasarkan titik fitur yang telah didapatkan, kemudian dilakukan ekstraksi fitur dengan menggunakan metode ekstraksi *Oriented FAST and Rotated BRIEF(ORB).* Fitur yang dihasilkan berupa *binary descriptor* BRIEF yang dapat menyimpan fitur intensitas keabuan dan juga orientasi dari *keypoint*.

Untuk menghitung nilai kecocokan fitur antara *template* dan citra masukan, digunakan *Hamming Distance*. Sedangkan, untuk mencocokkan fitur, digunakan metode klasifikasi KNN, yang dapat mengembalikan *k*-pasang fitur yang cocok. Dari pasangan fitur yang cocok, dicari pasangan fitur terbaik dengan menggunakan *ratio test* [6].

$$r < \frac{m}{n} \qquad (2)$$

Persamaan *ratio test* dapat dilihat pada (2). Dimana *r* merupakan *ratio test*, sedangkan m dan n merupakan 2 jarak dari sepasang fitur yang dihasilkan oleh KNN. Pasangan fitur dianggap *good match* jika memenuhi (2).

III. HASIL DAN DISKUSI

Pengujian sistem ini dilakukan pada sebuah *notebook* dengan spesifikasi prosesor Intel® Core™ i7-3517U CPU @ 1.90GHz (4 CPUs) , ~2.4GHz, RAM 4GB, dan berjalan pada sistem operasi Windows 10.

Pengujian dilakukan dengan menghitung nilai *recall*, *precision* dan lama proses. Nilai tersebut dihitung berdasarkan ketentuan berikut:

1. *True positive* : sebuah *tag* dengan objek mobil
2. *False negative* : sebuah mobil namun tanpa *tag*
3. *False positive* : sebuah *tag* namun tanpa objek mobil

Tag atau penanda pada citra udara hasil deteksi dihitung sebagai *true positive* jika posisi badan mobil yang terdeteksi berada di dalam penanda. Apabila terdapat lebih dari satu penanda pada satu mobil, maka penanda tersebut tidak dihitung sebagai *true positive*.

Terdapat beberapa skenario pengujian, yaitu deteksi dengan menggunakan *template matching*, *Haar cascade Classification* untuk mengetahui perbandingan kinerja deteksi masing-masing proses dengan *feature matching*. Skenario berikutnya yaitu deteksi dengan *feature matching* ORB, *feature matching* MSER, serta *feature matching* ORB dan MSER untuk mengetahui bagaimana pengaruh perbedaan metode deteksi *keypoint* dalam proses *feature matching*. Kelima skenario tersebut menggunakan 30 dataset citra udara dengan ukuran masing-masing 1000x1000 piksel. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1.
Perbandingan hasil deteksi setiap metode

Metode	*Precision*	*Recall*	Time
Template Matching	3.84%	14.06%	9.60 s
Haar cascade Classification	1.96%	15.19%	5.17 s
Feature Matching ORB	6.83%	24.17%	0.21 s
Feature Matching MSER	4.83%	8.55%	0.76 s
Feature Matching ORB dan MSER	10.71%	42.51%	0.88 s

Dari hasil pengujian tersebut tampak bahwa keseluruhan metode *feature matching* menunjukkan hasil yang lebih tinggi apabila dibandingkan dengan metode *template matching* dan *Haar cascade Classification*. Hal ini disebabkan oleh sifat *feature matching* yang tidak terpengaruh oleh rotasi atau perbedaan arah mobil pada citra dengan mobil pada *template*.

Sedangkan dari ketiga skenario *feature matching* dengan metode pencarian *keypoint* yang berbeda, metode ORB dan MSER menghasilkan kinerja yang paling tinggi. Hal ini disebabkan karena gabungan dari kedua metode tersebut dapat mendeteksi baik mobil berwarna cerah maupun mobil berwarna gelap.

Selain itu, dilakukan dua skenario pengujian dengan citra udara yang berbeda. Skenario pertama dilakukan dengan memotong citra menjadi ukuran yang lebih kecil untuk mengetahui pengaruh perbedaan ukuran citra udara terhadap hasil deteksi mobil pada setiap proses. Hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2.
Perbandingan citra udara berbeda ukuran

Ukuran	Metode	*Precision*	*Recall*
250x250 piksel	*Template Matching*	1.43%	4.00%
	Cascade Classification	12.50%	28.00%
	Feature Matching	13.10%	44.00%
1000x1000 piksel	*Template Matching*	0.00%	0.00%
	Cascade Classification	26.92%	28.00%
	Feature Matching	8.33%	16.00%

Pada skenario tersebut, terdapat peningkatan hasil deteksi pada *template matching* dan *feature matching* setelah dilakukan pemotongan citra udara. Hal ini disebabkan karena kemungkinan objek yang terdeteksi menjadi lebih sedikit. Sedangkan pada metode *cascade classification*, pemotongan citra udara justru meningkatkan *false alarm.*

Sedangkan, skenario kedua dilakukan dengan memotong citra udara pada area dengan objek yang terdiri dari jalan dan mobil saja. Pengujian ini dilakukan untuk mengetahui performa setiap metode untuk mendeteksi mobil pada citra udara yang tidak memiliki objek lain selain jalan dan mobil. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3.
Hasil deteksi citra jalan dan mobil

Metode	*Precision*	*Recall*
Template Matching	36.36%	33.33%
Cascade Classification	87.50%	53.85%
Feature Matching MSER ORB	100.00%	75.00%

Pada skenario kedua, terdapat peningkatan nilai *recall* dan *precision* apabila dibandingkan dengan rata-rata pada citra dengan banyak objek. Hal ini disebabkan karena tidak adanya objek gedung yang salah deteksi sebagai mobil.

IV. KESIMPULAN

Berdasarkan hasil pengamatan selama proses perancangan, implementasi dan seluruh pengujian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Metode *Template Matching* menghasilkan nilai rata-rata *recall* sebesar 3,84%. Sedangkan nilai rata-rata *precision* sebesar 14.06%. Rata-rata lama proses *template matching* untuk 30 citra udara adalah 9,6 detik.
2. Metode *Haar Cascade Classification* menghasilkan rata-rata *recall* sebesar 15,19%. Sedangkan nilai rata-rata *precision* yang dihasilkan paling rendah, yaitu 1,9%. Rata-rata lama proses *Haar Cascade Classification* untuk 30 citra udara adalah 5,17 detik.
3. Metode *Feature Matching* ORB menghasilkan nilai rata-rata *recall* sebesar 24.17%. Sedangkan nilai rata-rata *precision* adalah 6,83%. Rata-rata lama proses *feature matching* untuk 30 citra udara adalah 0,21 detik.
4. Metode *Feature Matching* MSER menghasilkan nilai rata-rata *recall* sebesar 8,55%. Sedangkan nilai rata-rata *precision* adalah 4,63%. Rata-rata lama proses *Feature Matching MSER* untuk 30 citra udara adalah 0,76 detik.
5. Metode *Feature Matching* MSER dan ORB menghasilkan nilai rata-rata *recall* 42.51%. Sedangkan nilai rata-rata *precision* adalah 10.71%. Rata-rata lama proses *Feature Matching* MSER dan ORB untuk 30 citra udara adalah 0,88 detik
6. Nilai *recall* tertinggi dihasilkan oleh metode *Feature Matching* dengan menggunakan ORB dan MSER. Hal Ini disebabkan karena kemampuan metode tersebut dalam mendeteksi mobil baik berwarna cerah maupun gelap. Metode ini juga dapat mendeteksi mobil meskipun adanya posisi serta ukuran mobil yang bervariasi.
7. Seluruh metode *Feature Matching* menghasilkan nilai rata-rata *recall* yang lebih tinggi apabila dibandingkan dengan metode *Template Matching* dan *Haar Cascade.* Selain itu juga lama proses *Feature Matching* lebih cepat dibandingkan dengan kedua metode yang lain.

UCAPAN TERIMA KASIH

Penulis, Diagnosa Fenomena, mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam menyelesaikan penelitian ini, diantaranya keluarga dan orang-orang terdekat yang selalu memberikan doa dan dukungannya serta dosen pembimbing dan pengajar di Jurusan Teknik Informatika atas saran dan bimbingannya.

DAFTAR PUSTAKA

[1] OpenCV, "OpenCV 3.0.0 Documentation," 3 Juni 2015. [Online]. Available: http://docs.opencv.org/3.0.0/d4/dc6/tutorial_py_*template_matching*.html. [Diakses 20 April 2016].

[2] L. Yu, Z. Yu dan Y. Gong, "An Improved ORB Algorithm of Extracting and *Matching Features*," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. VIII, no. 5, pp. 117-126, 2015.

[3] P. Viola dan M. Jones, "Rapid Object Detection using A Boosted *Cascade* of Simple *Features*," *Computer Vision and Pattern Recognition,* vol. I, pp. I-511, 2001.

[4] S. Soo, "Object detection using *Haar-cascade* Classifier," 2014.

[5] C. Schlosser, J. Reitberger dan S. Hinz, "Automatic car detection in high resolution urban scenes based on an adaptive 3D-model," dalam *Remote Sensing and Data Fusion over Urban Areas*, Oklahoma, 2003.

[6] E. Rublee, V. Rabaud, K. Konolige dan Bradski Gary, "ORB: an efficient alternative to SIFT or SURF," dalam *IEEE International Conference on Computer Vision*, California, 2011.