

Aplikasi Penghitung Kendaraan Pada Jalur Pantura Menggunakan Blob Deteksi Dan Kalman Filter

A Sumarudin^{1*}, Darsih^{2*}, Iryanto^{3*}, Adi Suheryadi^{4*}

* Jurusan Teknik Informatika, Politeknik Negeri Indramayu

shumaru@polindra.ac.id¹

Article Info

Article history:

Received 07-12-2019

Revised 21-03-2019

Accepted 01-01-2019

Keyword:

Deteksi kendaraan,
Blob tracker,
Kalman filter,
Penghitung kendaraan,
Jalan raya.

ABSTRACT

Jalur pantura jawa merupakan jalan raya wilayah utara jawa membentang dari merak sampai ke banyuwangi. Jalur pantura ini merupakan jalur padat yang dilalui oleh kendaraan baik roda 4 maupun lebih. Beban jalan pantura per tahun mencapai 1 juta ton. Selain itu juga dengan beban ini merupakan penyumbang penggunaan BBM (bahan bakar minyak) yang lebih sebesar 42 ribu kiloliter dan mengakibatkan pemberi emisi udara cukup banyak sebesar 350 ton per hari. Dengan demikian semakin banyak kendaraan yang melintas jalur pantura, sehingga perlu diketahui tentang jumlah kendaraan yang melintas dijalur pantura secara otomatis oleh sistem. Data ini dapat dijadikan rujukan beban jalan pantura secara tepat. Pada penelitian ini, menggunakan beberapa algoritma dari blob detection dengan menggunakan library openCV dan kalman filter. Hasil dari deteksi dengan menggunakan blob deteksi masih terdapat beberapa error yang cukup besar dan menggunakan kalman filter didapatkan 78.81 %.

Copyright © 2019 Journal of Applied Informatics and Computing.
All rights reserved.

I. PENDAHULUAN

Jalur pantura jawa merupakan jalan raya wilayah utara jawa membentang dari merak sampai ke banyuwangi. Jalur pantura ini merupakan jalur padat yang dilalui oleh kendaraan baik roda 4 maupun lebih. Sehingga beban jalan sangat padat. Jalur pantura merupakan jalan raya utama dengan akses pengiriman logistic antar propinsi di pulau jawa. Beban yang sangat tinggi ini perlu dilakukan monitoring beban kendaraan menggunakan teknologi informasi.



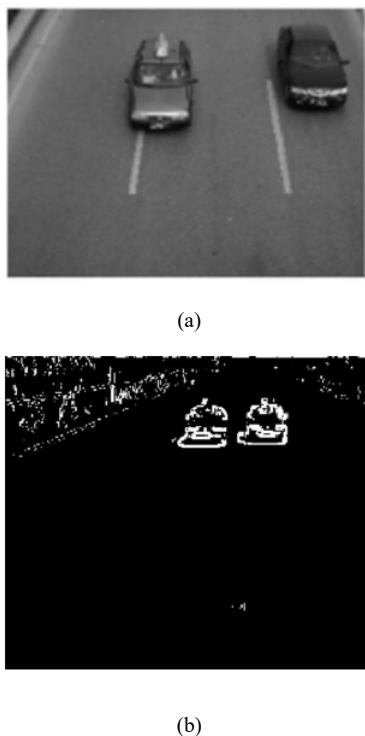
Beban jalan pantura ini per tahun mencapai 1 juta ton [10]. Selain itu juga dengan beban ini merupakan penyumbang penggunaan BBM (bahan bakar minyak) yang lebih sebesar 42 ribu kiloliter dan mengakibatkan pemberi emisi udara cukup banyak sebesar 350 ton per hari [10].

Dengan demikian semakin banyak kendaraan yang melintas jalur pantura, sehingga perlu diketahui tentang jumlah kendaraan yang melintas dijalur pantura secara otomatis oleh sistem. Data ini dapat dijadikan rujukan beban jalan pantura secara tepat. Pada penelitian ini, kita gunakan mencoba beberapa algoritma dari *blob detection* dan *kalman filter* yang diterapkan dalam sistem penghitung kendaraan



Gambar 1. Jalur Pantura Jawa

teknologi pengolahan citra digital dapat berbasis video image. Dengan menggunakan perbedaan antara image background dengan image detection. Metode yang digunakan biasa deteksi edge [1]. Algoritma yang digunakan menggunakan *multi-frame average method*, *Gaussian distribution model method*, *random update method*, *selection-update method*. Hasil deteksi adalah sebagai berikut:



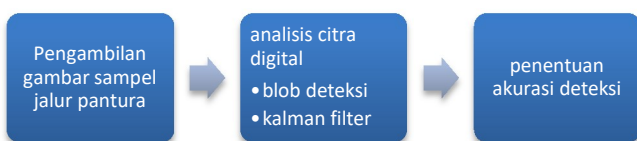
Gambar 2. (a) Input *Image* (b) Hasil deteksi menggunakan algoritma perbedaan *background* [1]

Selain algoritma deteksi tepi, deteksi kendaraan dapat pula menggunakan algoritma yang digunakan menggunakan openCL dengan metoda 2 tahap: [2]
 a) Hypothesis generation (HS)
 b) Hypothesis verification (HV)

Untuk penanaman sistem bisa menggunakan FPGA [3], OPenCL [2], stereo camera [3], static dan dynamic kamera [4], teknik deteksi malam hari [5]. Hal ini memungkinkan penggunaan FPGA, MCU, miniPc, PC or laptop. Dalam penelitian ini menggunakan miniPc berbasis raspberry pi, tujuan dari penggunaan miniPc aplikasi dapat *mobile* dan lebih kecil serta hemat daya untuk implementasi di Jalur Pantura Jawa.

II. METODE PENELITIAN

Dalam penelitian ini, kita menggunakan beberapa tahap penelitian yaitu:



Gambar 3. Rancangan penelitian

Dalam pengambilan gambar kita menggunakan kamera yang dipasang di jalan pantura. Video diambil dengan durasi 1 menit dan 2 menit.



Gambar 4. Video lalu lintas pantura

Setelah dilakukan pengambilan video lalu lintas yang berada di pantura kemudian kita lakukan analisis menggunakan OpenCV dan matlab. Dilakukan analisis error dari system yang dibuat.

III. DESAIN SISTEM PENDETEKSI KENDARAAN

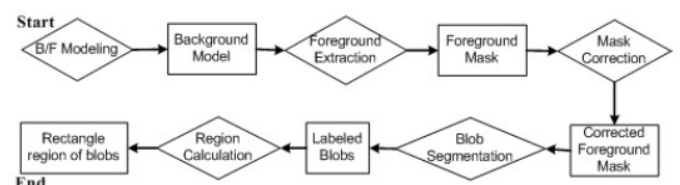
Desain pendeteksi kendaraan yang direncanakan dalam penelitian ini menggunakan dua metode, yaitu:

- a) Blob Detection
- b) Kallman Filter

System deteksi terdiri dari kamera yang mengarah ke jalan dan akan dilakukan pendeteksian di perangkat yang terpasang di jalan, hasil dari pendeteksian akan di kirim ke server menggunakan media wireless. Blob deteksi adalah hal penting dalam proses beberapa aplikasi teknik komputasi *vision* seperti intelegent pengawasan [6]. Intelegent pengawasan menggunakan analisis menekankan pada aktifitas dari obyek. [6]

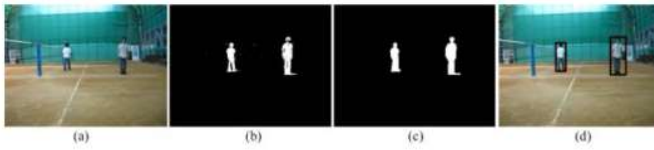
Blob deteksi algoritma untuk *motion* analisis terdapat langkah-langkah, sebagai berikut:[6]

1. Foreground mask extracting
2. Foreground mask correction
3. Blob segmentation trough connected component labeling
4. Regulation calculation



Gambar 6. Langkah kerja umum algoritma *blob detection* [6]

Hasil dari deteksi blob detection, seperti berikut:



Gambar 7. (a) Image asli (b) Foreground binary (c) foreground image setelah koreksi (d) Pelabelan Region foreground[6]

Penggunaan blob deteksi dapat juga digunakan untuk deteksi medis [7], penggunaan sensor optic [8], atau digunakan untuk penggunaan autonomous kendaraan menggunakan image processing berbasis blob deteksi [9]. Sedangkan untuk proses kalman filter adalah sebagai berikut. Secara umum kalman filter mencakup empat fase [11], sebagai berikut:

1. Process equations

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$p(w) \sim N(0, Q)$$

2. Measurement equation

$$z_k = Hx_k + v_k$$

$$p(v) \sim N(0, R)$$

3. Time update equation

$$\hat{x}_k^- = Ax_{k-1} + w_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

4. Measurement update equation

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

$$P_k = (1 - K_k H)P_k^-$$

Proses pengambilan video dilakukan dengan menggunakan kamera digital, berikut hasil pengambilan video di jalan pantura.

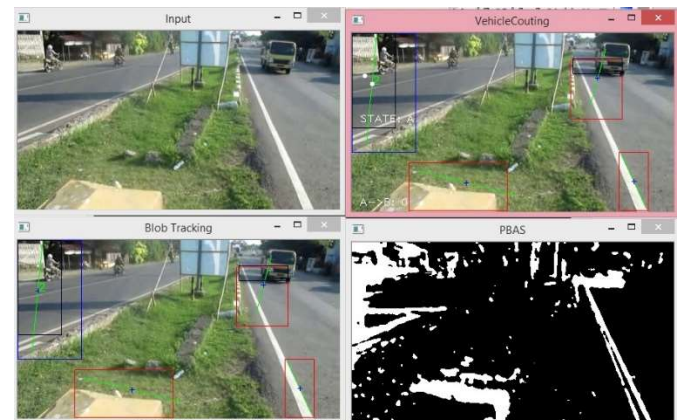


Gambar 8. Video Lalu Lintas Pantura

Algoritma yang digunakan dalam pendeteksi jumlah kendaraan menggunakan blob deteksi adalah sebagai berikut:

<p>Algorithm 1 Algoritma Counting menggunakan Deteksi Blob</p> <ol style="list-style-type: none"> 1: Input data (video) 2: Tentukan Counting Line 3: repeat 4: Background subtraction algorithm 5: Blob tracker algorithm 6: Counting vehicle algorithm 7: until End of video

Kemudian kami ambil video lalu lintas pantura seperti pada gambar 5.3 sehingga didapat hasil sebagai berikut:



Gambar 9. Hasil penghitung kendaraan di jalur pantura



Gambar 10. Hasil penghitung kendaraan di jalur pantura

Dari hasil deteksi menggunakan blob, masih terdapat beberapa error yang sangat besar dikarenakan posisi background dan foreground tidak stabil. Sedangkan untuk deteksi kendaraan menggunakan kalman filter dengan algoritma sebagai berikut:

Algorithm 1 Algoritma Counting menggunakan kalman filter

- 1: Input Video
- 2: Convert Video to Frame
- 3: Repeat
- 4: Background Substraction Algorithm
- 5: Prediction of Object Location
- 6: Blob Detection Algorithm
- 7: Update and Counting Vehicle Algorithm
- 8: until End of video

Berikut hasil menggunakan kalman filter

1. Masking original



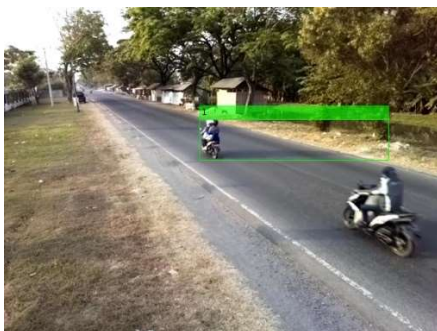
Gambar 11. Masking original

2. Result



Gambar 12. Hasil Masking

3. Result Masking



Gambar 13. Hasil vehicle counting

Dari hasil 150 Frame yang dideteksi (*true positif*) menggunakan kalman filter, terhitung 78.81% tingkat deteksi.

Hal ini dikarenakan masih ada bayangan dari obyek deteksi dan jarak antar obyek yang berdekatan. Hal ini bisa dilakukan dengan menambahkan filter untuk memisahkan bayangan dengan interest obyek.

V. KESIMPULAN

Dari hasil penelitian menggunakan blob deteksi yang didapat masih terdapat error dalam pendeteksian lalu lintas kendaraan pantura. Resolusi gambar yang sangat besar dan pengambilan gambar yang tidak stabil pada *blob detection*, sehingga mengakibatkan deteksi *background* dan *foreground* yang berubah-ubah. Sedangkan menggunakan kalman filter didapat deteksi (*true positif*) 78.81 %. Untuk mendapatkan tingkat error yang mendekati nol menggunakan filter untuk memisahkan bayangan kendaraan. Dari hasil penelitian ini, dapat disimpulkan penggunaan kalman filter memiliki nilai deteksi 78.81 % sehingga dapat diimplementasikan dalam sistem pendeteksian jumlah kendaraan di jalur pantura.

UCAPAN TERIMA KASIH

Terimakasih kepada Dirjen Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi dan Perguruan Tinggi atas dana penelitian dosen pemula yang sudah mendanai penelitian ini.

DAFTAR PUSTAKA

- [1] Z. Yuanyuan, Z. Kaiwen, and M. Yuming, "Research on Vehicle Detection Method Based on Video Image," *2012 Int. Conf. Ind. Control Electron. Eng.*, pp. 987–990, 2012.
- [2] K. Cheng, C. Lin, Y. Chen, T. Su, S. Lai, and J. Lee, "Design of vehicle detection methods with OpenCL programming on multi-core systems," *11th IEEE Symp. Embed. Syst. Real-time Multimed.*, pp. 88–95, 2013.
- [3] Seunghun Jin, Junguk Cho, Xuan Dai Pham, Kyoung Mu Lee, Sung-kee Park, Munsang Kim, and Jae Wook Jeon, "FPGA Design and Implementation of a Real-Time Stereo Vision System," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 15–26, 2010.
- [4] L. C. Leon and R. Hirata, "Vehicle detection using mixture of deformable parts models: Static and dynamic camera," *Brazilian Symp. Comput. Graph. Image Process.*, pp. 237–244, 2012.
- [5] S. Kim, S. Oh, J. Kang, K. Kim, S. Park, and K. Park, "Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion," *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 2173–2178, 2005.
- [6] B. N. Thanh and T. C. Sun, "An improved real-time blob detection for visual surveillance," *Proc. 2009 2nd Int. Congr. Image Signal Process. CISP'09*, pp. 0–4, 2009.
- [7] J. Liu, J. M. White, and R. M. Summers, "Automated detection of blob structures by hessian analysis and object scale," *Proc. - Int. Conf. Image Process. ICIP*, pp. 841–844, 2010.
- [8] M. Harmouchi, "Computer blob detection and tracking for highly repeatable optical fiber sensor."
- [9] D. Kiran and A. I. Rasheed, "FPGA Implementation of Blob Detection Algorithm for Object Detection in Visual Navigation."
- [10] <http://www.tempo.co/read/news/2014/03/29/090566333/Rel-Ganda-Kurangi-30-Persen-Beban-Jalur-Pantura> diakses 6 April 2015
- [11] Welch, G. B. (2001). *An Introduction to the Kalman Filter*. SIGGRAPH.