

## PENDETEKSI TINGKAT KEPADATAN JALAN MELALUI CANNY EDGE DETECTION

**Budi Hartanto<sup>1</sup>, Sholeh Hadi Setiawan<sup>2</sup>, Chandra Budi<sup>3</sup>**

Teknik Informatika, Universitas Surabaya

Jl. Raya Kalirungkt Surabaya - 60292.

Email: [budi@ubaya.ac.id](mailto:budi@ubaya.ac.id)<sup>1</sup>

[sholeh@ubaya.ac.id](mailto:sholeh@ubaya.ac.id)<sup>2</sup>

### *Abstrak*

*Informasi tentang tingkat kepadatan suatu jalan sudah menjadi kebutuhan yang mendesak, terutama di kota-kota besar di Indonesia seperti Jakarta atau Surabaya. Jika saja informasi ini bisa diperoleh maka si pengguna jalan dapat menghindari lokasi-lokasi tertentu yang sedang padat dan memilih jalan alternatif yang lebih longgar. Dengan demikian volume kendaraan dapat lebih tersebar dan kepadatan di suatu titik tertentu dapat dihindari atau dapat segera dikurangi.*

*Canny edge detection adalah salah satu metode pengolahan citra yang bertujuan untuk mencari tepian gambar. Pada penelitian ini, kondisi suatu jalan tertentu dengan berbagai kendaraan yang ada di dalamnya diambil citranya dan dilakukan pemrosesan canny edge detection. Sebagai hasilnya akan didapat sebuah citra yang berisi tepian-tepian dari berbagai kendaraan yang ada. Gambar tepian-tepian kendaraan ini kemudian dihitung dan digunakan untuk merepresentasikan tingkat kepadatan suatu jalan. Meskipun tidak ada jaminan semua tepian objek akan terdeteksi oleh Canny edge detection, tapi penggabungan metode ini dengan metode lain yang bertujuan untuk menghitung kepadatan suatu area akan menyebabkan tingkat kepercayaan hasil pengenalan kepadatan jalan tetap tinggi.*

*Pada penelitian ini citra digital yang menunjukkan kondisi suatu jalan telah berhasil diubah menjadi informasi yang menyatakan kepadatan jalan tersebut. Selama gambar yang diperoleh masih dapat dilihat secara baik secara visual (tidak peduli apakah cuaca mendung/tidak, siang/malam, dan sebagainya) maka tingkat kepadatan jalan tersebut masih dapat dideteksi dengan baik. Pada pengembangannya informasi ini dapat disebarluaskan ke berbagai pengguna jalan yang memerlukannya melalui berbagai media yang ada*

**Kata kunci:** *kepadatan jalan, canny edge detection, pengolahan citra.*

### **I. PENGANTAR**

Di kota-kota besar seperti Jakarta atau Surabaya, kebutuhan untuk mengetahui kepadatan jalan yang akan dilewati dapat dikatakan sudah menjadi kebutuhan tersendiri dalam berkendara. Jika saja pengendara jalan dapat mengetahui kondisi suatu jalan tertentu maka pengendara dapat menghindari jalan-jalan yang sedang dalam kondisi padat dan mencari jalan alternatif yang lebih longgar.

Kebutuhan akan informasi tingkat kepadatan jalan ini dapat difasilitasi jika saja kepadatan suatu jalan dapat diketahui secara otomatis melalui suatu perangkat tertentu dan dikirimkan kepada mereka yang membutuhkannya. Proses untuk mendeteksi kepadatan jalan secara otomatis inilah yang hendak dicapai melalui penelitian ini. Jalan yang hendak diketahui tingkat kepadatannya akan diambil gambarnya untuk kemudian dilakukan pemrosesan terhadapnya hingga didapat informasi yang menyatakan

tingkat kepadatan jalan tersebut.

Salah satu metode yang terlibat dalam pengubahan gambar jalan menjadi informasi adalah *canny edge detection*. Sebagaimana yang tercermin dari nama metode ini, *canny edge detection* akan mendeteksi tepian yang ada di suatu gambar dan menghilangkan berbagai bagian lain yang tidak diperlukan. Setelah tepian gambar diperoleh maka proses selanjutnya yang harus dilakukan adalah menganalisa kepadatan tepian-tepian objek yang tersisa dalam gambar tersebut. Kepadatan tepian-tepian ini dapat disetarakan dengan kepadatan objek yang ada dalam gambar.

Pada penelitian ini, kepadatan suatu jalan digolongkan menjadi 5 kategori dengan istilah: sepi, cukup sepi, sedang, cukup padat, dan padat. Dari hasil uji coba dapat diketahui bahwa informasi yang dihasilkan melalui penelitian ini telah mampu mencerminkan tingkat-kepadatan suatu jalan.

## 2. PENDETEKSI KEPADATAN

Kepadatan suatu jalan dapat diketahui secara otomatis melalui beberapa tahap pemrosesan. Pertama-tama jalan yang hendak diketahui kepadatannya harus diambil gambarnya. Suatu data citra biasanya mempunyai ukuran yang cukup besar sehingga kurang layak untuk didistribusikan secara langsung kepada mereka yang membutuhkannya. Untuk itu gambar yang didapat harus diolah sedemikian rupa hingga didapat data lain yang lebih sederhana seperti data teks misalnya.

Pengubahan data gambar menjadi data teks biasanya harus dilakukan melalui beberapa tahap. Pertama-tama data gambar yang ada harus diperbaiki terlebih dahulu untuk menghilangkan berbagai *noise* yang mungkin ada serta memperbaiki feature objek yang mungkin tertangkap secara tidak sempurna oleh kamera. Setelah itu gambar yang diperoleh harus

disederhanakan lagi sehingga hanya bagian-bagian yang hendak dilibatkan dalam proses sajalah yang seharusnya masih tertinggal dalam gambar. Kedua proses ini dapat dilakukan melalui metode *canny edge detection* yang akan menghilangkan berbagai data gambar yang tidak diperlukan hingga hanya tertinggal tepian objek saja.

Setelah tepian gambar diperoleh maka proses selanjutnya yang harus dilakukan adalah melakukan penghilangan gambar latar belakang. Pada penelitian ini gambar latar belakang yang ada akan dihilangkan dengan mengurangi gambar target terhadap gambar master. Yang dimaksud dengan gambar target adalah gambar yang akan dihilangkan latar belakangnya, sedang yang dimaksud dengan gambar master adalah gambar latar belakang itu sendiri. Untuk itu gambar latar belakang diharapkan benar-benar hanya terisi gambar latar belakang dari jalan yang akan dikenali kepadatannya. Melalui proses ini diharapkan gambar target hanya akan berisi gambar kendaraan-kendaraan yang ada di jalan tersebut.

Proses terakhir yang harus dilakukan tentu saja adalah menerjemahkan tepian-tepian gambar kendaraan yang ada pada gambar target menjadi informasi yang menunjukkan kepadatan suatu jalan. Metode yang digunakan untuk mengubah tepian objek menjadi informasi kepadatan jalan adalah metode penelusuran titik. Melalui metode ini luasan daerah yang tidak berisi objek akan dihitung dan akan digunakan untuk menentukan padat - tidaknya kondisi suatu jalan. Beberapa sub bab berikut akan memberi penjelasan secara rinci terhadap proses-proses yang dilakukan dalam tiap langkah.

### 2.1 Canny Edge Detection

*Canny edge detection* merupakan metode pencari tepian gambar yang paling optimum (Green, 2002). Tidak semata-

mata hanya mencari tepian gambar, metode Canny juga melakukan beberapa proses lain agar tepian yang dihasilkan dari suatu gambar menjadi lebih baik dan jelas. Beberapa proses tersebut antara lain adalah proses penghilangan noise, penghapusan tepian yang terlalu tebal, hingga proses penggabungan kembali tepian yang mungkin terputus.

Secara umum proses yang dilakukan dalam *Canny edge detection* terbagi menjadi beberapa proses sebagai berikut (Moenslund, 2009; Nixon dan Aquado 2008):

- Penghilangan noise
- Pencarian gradien
- Penghilangan tepian yang tidak perlu
- Pengubahan menjadi gambar biner
- Penyambungan tepian

Proses pertama yang ada dalam metode *Canny edge detection* adalah proses penghilangan noise. Langkah yang dilakukan dalam proses ini adalah dengan menghaluskan (smoothing) gambar yang ada. Sebuah operator konvolusi Gaussian Smoothing dapat diterapkan untuk melakukan hal ini (Fisher et al., 2003). Salah satu contoh operator Gaussian Smoothing dapat dilihat pada persamaan (1).

Setelah noise pada gambar dihilangkan dengan melakukan penghalusan gambar, proses selanjutnya yang harus dilakukan adalah pencarian gradien pada gambar. Gradien pada suatu gambar bisa diperoleh dengan melakukan konvolusi terhadap operator Sobel (Nixon dan Aquado, 2008).

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (1)$$

Persamaan (2) dan (3) menunjukkan

operator Sobel yang dapat digunakan dalam proses pencarian gradien suatu titik:

$$g_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2)$$

$$g_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3)$$

Gradien suatu titik dapat diperkirakan melalui persamaan (4) atau untuk mengurangi proses komputasi yang harus dilakukan, dapat juga disederhanakan menjadi persamaan (5):

$$g = \sqrt{g_y^2 + g_x^2} \quad (4)$$

$$g = |g_y| + |g_x| \quad (5)$$

Penebalan titik-titik pada gradien garis akan menyebabkan tepian objek sudah cukup terlihat dengan jelas. Hanya saja tepian objek ini masih terlihat cukup tebal dan belum dapat menunjukkan tepian objek yang sebenarnya (Mousland, 2009).

Proses selanjutnya yang harus dilakukan adalah mengurangi ketebalan tepian objek. Pengurangan ketebalan tepian objek dapat dimulai dengan mencari arah suatu garis. Persamaan (6) dapat digunakan untuk mencari arah dari suatu garis.

$$\theta = \arctan\left(\frac{|g_y|}{|g_x|}\right) \quad (6)$$

Karena titik-titik yang ada akan tersimpan secara diskrit maka arah garis yang diperoleh dapat disederhanakan menjadi sudut dengan kelipatan  $45^\circ$ . Dengan demikian, nilai sudut  $0^\circ \leq \theta \leq 45^\circ$  dapat diubah menjadi  $45^\circ$ , nilai  $45^\circ < \theta \leq 90^\circ$  dapat diubah menjadi  $90^\circ$  dan seterusnya. Melalui proses ini akan didapat 8 sudut yang merepresentasikan

arah mata angin.

Proses pengurangan ketebalan garis dapat dilakukan dengan menghilangkan tepian garis yang tidak perlu. Penghilangan ini dapat dilakukan dengan membandingkan nilai intensitas suatu titik dengan titik lain yang berada "di depan" dan "di belakang"-nya. Titik B disebut berada "di depan" titik A bila titik B terletak satu piksel jauhnya dan searah dengan titik A. Sementara itu titik C akan disebut berada "di belakang" titik A bila titik C terletak satu piksel jauhnya dan berlawanan arah dengan titik A.

Sebagai contoh, bila titik A mempunyai arah garis sebesar  $90^{\circ}$  (arah atas) maka titik yang berada "di depan" titik A adalah titik lain yang terletak di atasnya. Sementara itu titik yang terletak "di belakang" titik A adalah titik lain yang terletak di bawahnya. Garis yang diwakili oleh titik A dapat dihapus bila ternyata nilai intensitas pada titik A tidak lebih besar dari nilai intensitas titik lain yang berada "di depannya" atau "di belakangnya".

Setelah tepian gambar diperkecil, proses selanjutnya yang harus dilakukan adalah mengubah gambar jalan dari gambar greyscale menjadi biner. Bila pada umumnya perubahan gambar ini dilakukan melalui satu nilai batas (*threshold*) maka *Canny edge detection* menggunakan dua nilai batas untuk proses ini. Sebut saja nilai batas pertama adalah nilai batas atas, dan nilai batas kedua adalah nilai batas bawah.

Karena *Canny edge detection* menggunakan dua nilai batas maka perubahan gambar greyscale menjadi biner akan dilakukan melalui dua tahap. Pada tahap pertama, gambar greyscale akan diubah terlebih dahulu menjadi gambar dengan tiga nilai intensitas. Selanjutnya gambar dengan tiga nilai intensitas ini akan diubah lagi menjadi gambar dengan dua nilai intensitas (biner) (proses ini akan dilakukan pada tahap akhir proses *Canny edge detection*)

Misalkan saja tiga nilai intensitas yang akan dihasilkan oleh proses ini kita sebut dengan nama hitam, abu-abu, dan putih. Perubahan gambar greyscale menjadi gambar dengan tiga nilai intensitas dapat dilakukan dengan aturan sebagai berikut:

- Nilai intensitas yang lebih besar atau sama dengan nilai batas atas akan diubah menjadi warna putih
- Nilai intensitas yang lebih kecil atau sama dengan nilai batas bawah akan diubah menjadi warna hitam
- Nilai intensitas yang berada diantara nilai batas atas dan bawah akan diubah menjadi warna abu-abu.

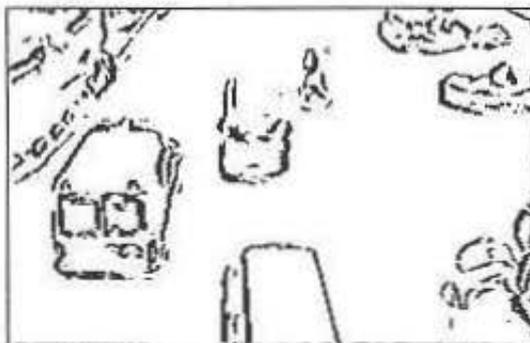
Langkah terakhir dari *Canny edge detection* disebut dengan langkah penyambungan tepian. Bila disampaikan dengan istilah yang lebih sederhana, proses yang dilakukan pada langkah ini sebenarnya adalah menentukan apakah titik dengan warna abu-abu akan diubah menjadi warna hitam atau putih.

Penentuan apakah sebuah titik berwarna abu-abu akan diubah menjadi hitam atau putih didasarkan dari kedekatan lokasi titik tersebut dengan titik lain yang berwarna hitam (tepi). Bila titik yang berwarna abu-abu berada di lokasi yang tepat bersebelahan dengan titik lain yang berwarna hitam maka warna titik tersebut diubah menjadi hitam. Sebaliknya warna titik tersebut akan diubah menjadi putih. Melalui proses ini tepian-tepi objek yang didapat dari *Canny edge detection* akan menjadi lebih terlihat, sementara garis-garis lain yang sebenarnya bukan tepian objek akan hilang.

Gambar 1 menunjukkan contoh suatu gambar tertentu yang akan diproses dengan *Canny edge detection*. Hasil pemrosesan terhadap gambar ini dapat dilihat pada Gambar 2.



Gambar 1. Gambar yang akan dikenali



Gambar 2. Hasil *Canny edge detection* terhadap gambar 1

## 2.2 Penghapusan Latar Belakang

Proses selanjutnya yang harus dilakukan dalam pengenalan kepadatan jalan ini adalah proses penghapusan gambar latar belakang. Melalui proses ini diharapkan objek yang tersisa pada gambar hanyalah objek yang memang perlu dihitung saja.

Untuk keperluan penghapusan latar belakang ini, sebuah gambar jalan tanpa kendaraan harus diambil untuk digunakan sebagai pengurang terhadap gambar jalan yang berisi kendaraan. Dalam penelitian ini, gambar latar belakang tersebut diberi istilah gambar master sedangkan gambar yang akan dikenali kepadatannya disebut dengan istilah gambar target.

Setelah objek yang ada pada gambar target dan gambar master diubah menjadi tepian-tepiannya saja maka proses penghilangan latar belakang objek pada gambar target dilakukan dengan mengurangkan titik-titik yang bersesuaian dari gambar target dengan gambar master. Untuk keperluan ini, sangatlah penting

untuk mendapatkan gambar master dari posisi yang sama persis dengan gambar target.

Penghapusan gambar latar belakang ini juga dioptimalkan dengan meminta user untuk memilih lokasi jalan yang tidak akan dilewati kendaraan. Bagian ini akan dihapus dan tidak akan diikuti dalam proses yang lebih lanjut. Dalam penelitian ini, lokasi jalan yang tidak dilewati kendaraan dibatasi akan selalu berada di bagian kiri dan kanan jalan. Gambar 3 menunjukkan contoh hasil proses penghapusan latar belakang gambar target.



Gambar 3. Hasil penghapusan latar belakang

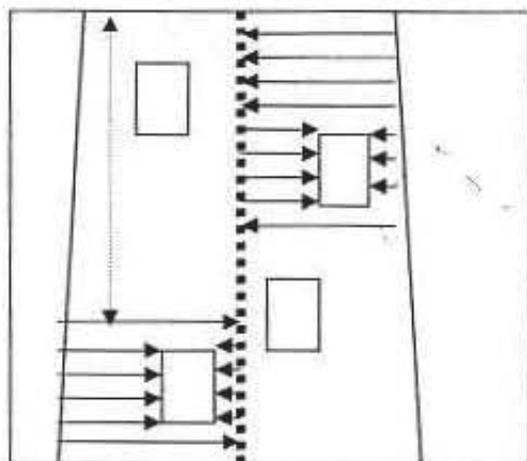
## 2.3 Luasan Daerah Kosong

Yang dimaksud dengan tingkat kepadatan jalan dalam penelitian ini adalah seberapa banyak luasan jalan yang digunakan untuk kendaraan dibanding dengan luasan jalan secara keseluruhan. Untuk keperluan ini, tingkat kepadatan jalan akan dihitung dalam satuan persen dan hasilnya akan digolongkan dalam 5 kategori yaitu sepi, agak sepi, sedang, agak padat, dan padat. Prosentase kepadatan untuk masing-masing kategori beserta nama kategori dapat diubah-ubah sesuai dengan keinginan user dan dapat disesuaikan dengan kondisi untuk suatu jalan tertentu.

Untuk itu, langkah terakhir yang harus dilakukan adalah menentukan tingkat kepadatan jalan dari gambar tepian kendaraan yang telah diproses dari berbagai langkah sebelumnya. Dalam

penelitian ini proses yang dilakukan untuk menentukan kepadatan jalan dilakukan dengan menghitung luasan jalan yang tidak berisi kendaraan (luasan daerah kosong) dibanding dengan luasan jalan secara keseluruhan. Selanjutnya persentase kepadatan jalan dapat dihitung dengan mengurangkan angka 100% dengan persentase luasan daerah yang kosong.

Proses penghitungan persentase luasan daerah yang kosong dapat dihitung dengan melakukan penelusuran titik yang berada dalam bagian jalan yang dilewati kendaraan. Gambar 4 menunjukkan ilustrasi proses penelusuran titik yang dilakukan untuk mendapatkan persentase luasan daerah kosong.



Gambar 4. Ilustrasi proses perhitungan luasan daerah kosong

Pertama-tama akan dilakukan penelusuran dari kiri ke kanan, mulai dari lokasi kiri atas bagian jalan yang dilewati kendaraan. Bila tidak dijumpai tepian objek (kendaraan) pada gambar tersebut maka penelusuran akan dilakukan terus hingga ke bagian tengah gambar. Sebaliknya bila pada saat penelusuran didapatkan informasi akan tepian suatu objek maka proses penelusuran akan dilanjutkan dari bagian tengah gambar ke arah kiri.

Proses yang serupa akan dilakukan dengan melakukan penelusuran dari kanan ke kiri, mulai dari bagian kanan atas

gambar jalan yang dilewati kendaraan. Penelusuran juga akan dilakukan hingga ke bagian tengah gambar selama tidak dijumpai adanya tepian objek.

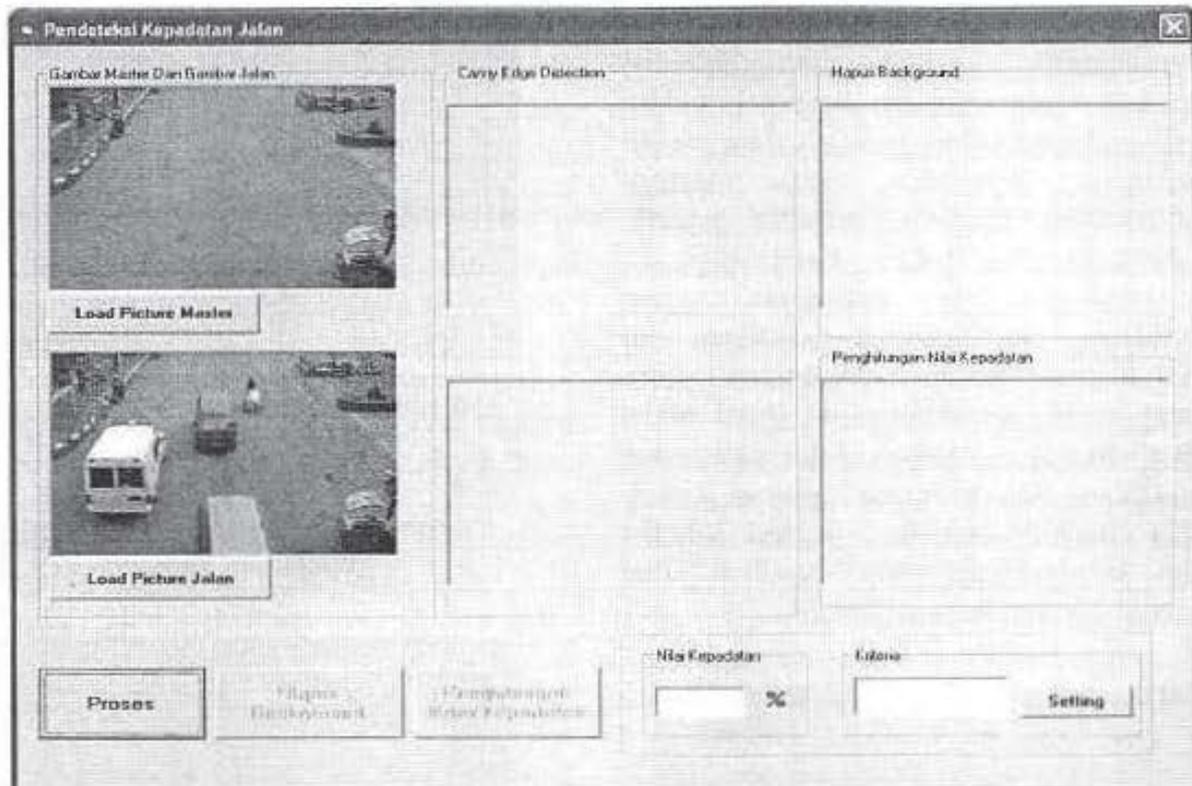
Selama proses penelusuran, akan dihitung berapa banyaknya titik yang berhasil dilewati. Jumlah titik ini akan digunakan untuk menentukan luasan daerah kosong dari jalan tersebut. Sementara itu luasan jalan secara keseluruhan dapat dihitung melalui perkalian dimensi gambar dikurangi luasan daerah yang tidak dapat dilewati kendaraan.

Hasil perhitungan luasan daerah kosong ini kemudian diubah dalam persen dan dilakukan penghitungan tingkat kepadatan jalan tersebut. Hasil perhitungan ini kemudian disesuaikan dengan tabel kepadatan jalan yang dapat ditentukan sendiri oleh user untuk mengetahui tingkat kepadatan jalan tersebut dalam bentuk enumerasi.

### 3. UJI COBA DAN EVALUASI

Hasil implementasi pengenalan tingkat kepadatan jalan dapat dilihat pada gambar 5. Pertama-tama user harus memilih dua buah gambar, yang disebut dengan istilah gambar master dan gambar target. Gambar master harus diisi dengan gambar jalan saat sedang dalam kondisi kosong atau sebagian besar tidak berisi kendaraan. Pada saat gambar master dipilih, user juga akan diminta untuk menentukan daerah-daerah gambar yang bukan merupakan jalan raya. Dalam penelitian ini, daerah yang bukan merupakan jalan raya diasumsikan selalu berada di sisi kiri dan kanan jalan.

Sementara itu, gambar target dapat diisi dengan gambar jalan yang harus diambil dari lokasi yang sama persis dengan gambar master. Gambar target ini dapat berisi kendaraan untuk dianalisa tingkat kepadatannya.



Gambar 5. Program pendeteksi tingkat kepadatan jalan

Tingkat kepadatan jalan yang diperoleh melalui program aplikasi ini sebenarnya dinyatakan dalam persentase dari luasan jalan yang berisi kendaraan dibandingkan dengan luasan jalan secara keseluruhan. Hanya saja dari sudut pandang pengguna, informasi kepadatan jalan secara kuantitatif ini mungkin akan terasa tidak umum dan tidak cukup mudah untuk dimengerti. Karena itu informasi kuantitatif yang diperoleh program aplikasi akan diubah menjadi informasi kualitatif yang lebih umum digunakan di masyarakat.

Untuk keperluan tersebut, tingkat kepadatan jalan yang diperoleh dari hasil perhitungan akan dikelompokkan menjadi 5 kelompok. User dapat mengubah nama serta daerah (*range*) nilai untuk masing-masing kelompok tersebut. Pada kondisi awal, program aplikasi akan menentukan nama kelompok beserta daerah nilainya dengan ketentuan sebagai berikut:

- Sepi : kepadatan  $\leq 20\%$
- Cukup Sepi :  $20\% < \text{kepadatan} \leq 40\%$

- Sedang :  $40\% < \text{kepadatan} \leq 60\%$
- Cukup Padat:  $60\% < \text{kepadatan} \leq 80\%$
- Padat : kepadatan  $> 80\%$

### 3.1 Uji Coba Tingkat Kepadatan Jalan

Uji coba pertama yang dilakukan adalah membandingkan informasi tingkat kepadatan jalan yang dihasilkan program dengan pengamatan secara visual. Untuk itu beberapa gambar jalan dengan tingkat kepadatan yang berbeda-beda telah dipersiapkan sebagaimana terlihat di gambar 6 hingga 10. Dari hasil uji coba, dapat dilihat bahwa informasi yang dihasilkan program aplikasi cukup dapat merepresentasikan tingkat kepadatan suatu jalan.

Pada gambar 6, saat secara visual jalan terlihat dalam kondisi sepi tingkat kepadatan jalan yang dihasilkan oleh program aplikasi adalah sebesar 7.059%. Sebagaimana batasan yang diberlakukan, tingkat kepadatan jalan dengan persentase di bawah 20% akan dinyatakan sebagai kondisi sepi.

Gambar 7, 8, 9, dan 10 secara berturut-turut menyatakan kondisi kepadatan jalan dengan kategori: cukup sepi, sedang, cukup padat, dan padat. Persentase kepadatan untuk masing-masing kondisi jalan tersebut adalah: 30.59%, 55.47%, 76.12%, dan 81.10%.

Jika tingkat kepadatan yang dihasilkan oleh program aplikasi ini dibandingkan dengan pengamatan secara visual untuk masing-masing jalan maka dapat dikatakan bahwa informasi yang dihasilkan oleh program aplikasi cukup dapat merepresentasikan tingkat kondisi jalan sebagaimana jika kondisi jalan tersebut diamati secara manual.



Nilai Kepadatan	Kriteria
55.47 %	Sedang

Gambar 8. Jalan dengan tingkat kepadatan: sedang



Nilai Kepadatan	Kriteria
7.059 %	Sepi

Gambar 6. Jalan dengan tingkat kepadatan: sepi



Nilai Kepadatan	Kriteria
76.12 %	Cukup Padat

Gambar 9. Jalan dengan tingkat kepadatan: cukup padat



Nilai Kepadatan	Kriteria
30.59 %	Cukup Sepi

Gambar 7. Jalan dengan tingkat kepadatan: cukup sepi



Nilai Kepadatan	Kriteria
81.10 %	Padat

Gambar 10. Jalan dengan tingkat kepadatan: padat

### 3.2 Uji Coba di Malam Hari

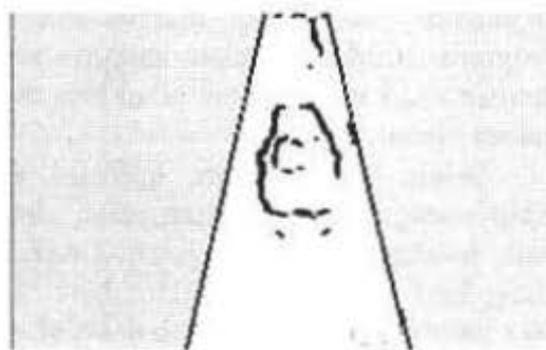
Seluruh gambar yang digunakan pada uji coba sebelumnya, diambil pada siang hari sehingga semua jalan mendapat penerangan yang cukup. Uji coba selanjutnya akan melihat apakah tingkat kepadatan jalan masih dapat dinyatakan dengan baik jika gambar jalan diambil di malam hari.

Gambar 11 menunjukkan contoh pengambilan gambar suatu jalan di malam hari. Meskipun malam hari, lokasi jalan dan kendaraan yang melintas di atasnya masih terlihat dengan cukup jelas berkat lampu penerangan jalan yang ada. Pada kondisi jalan semacam ini, *Canny edge detection* masih dapat menunjukkan tepian objek dengan baik (lihat gambar 12). Sebagai hasilnya, tentu saja tingkat kepadatan jalan jalan ini masih dapat dikenali program aplikasi dengan baik, meskipun gambar jalan diambil di malam hari.



Nilai Kepadatan	Kriteria
6.083 %	Sepi

Gambar 11. Suatu jalan di malam hari



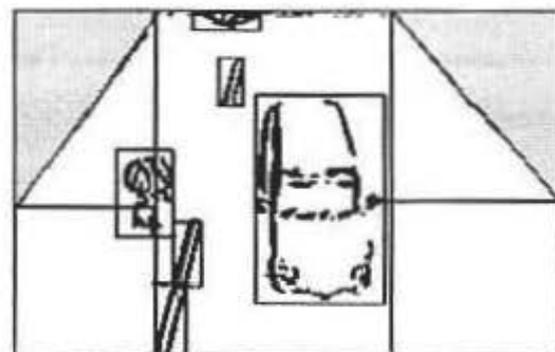
Gambar 12. Hasil deteksi tepian

### 3.3 Verifikasi Tingkat Kepadatan Jalan

Jika diperhatikan pada berbagai gambar yang telah ada, hasil *Canny edge detection* tidak selalu memberikan seluruh tepian yang ada pada suatu objek. Beberapa variabel yang terlibat dalam proses *Canny edge detection* beserta kualitas gambar yang diinputkan akan menentukan seberapa baik tepian objek yang dihasilkan.

Untuk itu pada bagian ini ingin diuji seberapa besar perbedaan yang akan muncul bila tepian objek dapat diketahui secara keseluruhan dibandingkan dengan kondisi yang dihasilkan program pada saat ini. Untuk itu akan diambil beberapa gambar sebagai contoh kasus dan program aplikasi akan digunakan untuk mengetahui tingkat kepadatan dari masing-masing jalan tersebut. Setelah itu tingkat kepadatan jalan pada gambar yang sama akan dihitung secara manual.

Penghitungan tingkat kepadatan jalan secara manual dapat dilakukan dengan membagi-bagi gambar jalan dan objek menjadi kumpulan persegi panjang atau segitiga (lihat contoh pembagian ini di gambar 13). Panjang dan lebar masing-masing persegi panjang atau segitiga ini kemudian diukur secara manual dan dilakukan penghitungan luas terhadap masing-masing objek yang ada.



Gambar 13. Contoh pembagian jalan menjadi persegi panjang dan segitiga

Dari hasil pemotongan berbagai objek yang ada di gambar 13, dapat dilakukan berbagai perhitungan sebagai berikut:

- Luas jalan:**  
 yang dimaksud dengan luas jalan adalah seluruh daerah berwarna putih yang ada pada gambar. Perhitungan luas jalan dapat dilakukan dengan mudah dengan membagi jalan di gambar 13 menjadi 5 bagian, yaitu tiga buah persegi panjang, dan dua buah segitiga. Luas jalan yang didapat melalui cara ini adalah:  $2,8 + 4,64 + 16,83 + 3,11 + 5,29 = 32,67$
- Luas seluruh objek yang ada di jalan:**  
 penghitungan luas seluruh objek yang ada di jalan dapat dilakukan dengan membuat persegi panjang yang melingkupi masing-masing objek yang ada. Setelah luasan masing-masing objek dihitung maka luasan jalan yang berisi objek dapat dihitung dengan menjumlahkan luasan seluruh objek yang ada. Dari hasil perhitungan, luasan seluruh objek yang ada di gambar 13 adalah:  $0,24 + 0,21 + 0,84 + 0,5 + 0,4 + 5,4 = 7,59$
- Persentase kepadatan jalan:**  
 persentase kepadatan jalan dari hasil perhitungan secara manual ini adalah 23,23% (didapat dari  $7,59 / 32,67 \times 100$  %). Sementara itu, tingkat kepadatan jalan yang dihasilkan oleh program aplikasi untuk gambar yang sama adalah: 22,89%. Dengan demikian perbedaan tingkat kepadatan jalan yang dihasilkan oleh program aplikasi untuk gambar jalan seperti yang terlihat di gambar 13 adalah 0,34%.

Berbagai perbedaan hasil perhitungan kepadatan jalan yang dilakukan secara manual dengan hasil pemrosesan program aplikasi dapat dilihat pada tabel 1.

Tabel 1. Perbandingan kepadatan jalan

Gambar ke	Hasil Program	Hasil Manual	Selisih
1	22,89 %	23,23 %	0,34 %
2	17,85 %	19,83 %	1,98 %
3	6,08 %	10,24 %	4,16 %
4	5,86 %	6,83 %	0,97 %

Empat contoh gambar akan digunakan sebagai contoh kasus perbandingan tingkat kepadatan jalan. Dari empat contoh kasus tersebut perbedaan terbesar terjadi pada gambar 3 dengan tingkat perbedaan sebesar 4,16% dan terkecil pada gambar 1 dengan tingkat perbedaan sebesar 0,34%. Rata-rata tingkat perbedaan untuk seluruh gambar tersebut adalah 1,86%. Angka ini dapat dikatakan cukup rendah sehingga dapat disimpulkan bahwa meskipun tepian objek yang didapat dari Canny edge detection tidak selalu sempurna, tapi hasil perhitungan kepadatan jalan tidak akan terlalu berpengaruh dibanding jika seluruh tepian objek bisa diperoleh secara sempurna.

#### 4. KESIMPULAN DAN SARAN

Penelitian ini telah menunjukkan bahwa tingkat kepadatan suatu jalan dapat dideteksi dengan menerapkan *Canny edge detection* dalam salah satu langkah pengenalannya. Dalam beberapa kasus yang diuji coba, informasi tingkat kepadatan jalan yang dikeluarkan oleh program aplikasi sudah cukup serupa dengan tingkat kepadatan jalan bila dilihat secara visual.

Selain itu, program aplikasi dapat tetap mengenali kepadatan jalan dengan baik meskipun gambar jalan diambil di siang hari maupun di malam hari. Tentu saja gambar jalan dan gambar objek yang berada di jalan tersebut harus tetap dapat terlihat secara baik saat dilihat secara

visual.

Pengembangan yang dapat dilakukan terhadap hasil penelitian ini adalah dengan menghubungkannya dengan media informasi lain seperti Web atau SMS. Melalui media informasi tersebut, pengguna jalan dapat mencari atau

meminta informasi kondisi suatu jalan tertentu melalui handphone. Setelah itu sistem dapat memberikan balasan berupa informasi tentang kondisi jalan yang diminta melalui media yang sama.

## DAFTAR PUSTAKA

1. Fisher, R., Perkins, S., Walker A., Wolfart. E, (2003). Gaussian Smoothing. (<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>). Diakses tanggal 25 Januari 2010.
2. Green, B. (2002). Canny Edge Detection Tutorial. ([http://www.pages.drexel.edu/~weg22/can\\_tut.htm](http://www.pages.drexel.edu/~weg22/can_tut.htm)). Diakses tanggal 25 Januari 2009
3. Moeslund, T. (2009). Canny Edge Detection. ([http://www.cvmt.dk/education/teaching/f09/VGIS8/AIP/canny\\_09gr820.pdf](http://www.cvmt.dk/education/teaching/f09/VGIS8/AIP/canny_09gr820.pdf)). Diakses tanggal 25 Januari 2010.
4. Nixon, M., Aquado A.S., (2008). Feature Extraction & Image Processing. Academic Press.

*PROSIDING SEMINAR NASIONAL  
RISET DAN TEKNOLOGI TERAPAN  
(RITEKTRA) 2010*

*"Teknologi Terapan dalam Upaya  
Meningkatkan Produktivitas dan Daya Saing  
Industri Nasional"*

Jakarta, 16 – 17 Juni 2010



Fakultas Teknik  
Unika Atma Jaya Jakarta



Ikatan Sarjana Teknik  
dan Manajemen Industri



Badan Kerjasama Penyelenggara  
Pendidikan Tinggi Teknik Industri



IBK Enterprise

ISBN: 978-602-97094-0-7

*PROSIDING SEMINAR NASIONAL  
RISET DAN TEKNOLOGI TERAPAN  
(RITEKTRA) 2010*

Hak Cipta @ 2010 pada penerbit

**Tim Editor:**

Yanto, ST. MSc.  
Veronica Windha M. ST. MT  
Catherine Olivia, ST. MT.  
Harjadi Gunawan, ST. MEngsc.

**Tim Reviewer:**

Prof.Dr.Ir. Wegie Ruslan, MSMath., MBA., IPM  
Prof. Ir. Hadi Sutanto, MMAE, PhD.  
Dr. Ir. Maria A. Kartawidjaja, M.Kom  
Dr. Ir. P. J. Prita Dewi Basoeki  
Dr. Henry Kartarahardja  
Dr. Lukas, ST., MAI.  
Dr. A. Adya Pramudita, ST. MT.  
Hotma A. Hutahaeen ST. MT.

Penerbit:



Fakultas Teknik  
Unika Atma Jaya Jakarta  
Jalan Jendral Sudirman 51  
Jakarta 12930

TE-B-01	<b>Implementasi Oracle In Memory Database Untuk Optimasi Waktu Proses dan High Availability System</b> Kristoko Dwi Hartomo, Atik Setiyanti, , Okta Ryan Kurniawan - Universitas Kristen Satya Wacana	112
TE-B-02	<b>Desain dan Implementasi Sistem Penerjemah Bahasa Berbasis Google AJAX Language API dan cURL Menggunakan Framework CodeIgniter</b> Kristoko Dwi Hartomo, S.Kom, M.Kom, George Nikijuluw, S.Pd., Dicky Kurniawan, S.Kom - Universitas Kristen Satya Wacana	124
TE-B-03	<b>Perancangan Bagian Prosesor Sparc 32 Bit Untuk Mengolah Instruksi Arithmetic/Logic/Shift Menggunakan Modelsim</b> Adjie Prasetyo, Karel Octavianus Bachri - Universitas Katolik Indonesia Atma Jaya	135
TE-B-04	<b>Sistem Perencana Jalur dengan Metode anytime dynamic a* (ad*) Pada kursi roda mandiri e-chair</b> Lukas, Adrianus R. Hernando - Universitas Katolik Atma Jaya Jakarta	149
TE-B-05	<b>Perancangan Sistem Pengenalan Personal Terpadu menggunakan RFID</b> Andi Adriansyah, Mukhlis Taher dan Abdul Halim - Universitas Mercu Buana	164
TE-B-06	<b>Aplikasi Pemetaan Kawasan Universitas Katolik Atma Jaya Kampus Semanggi</b> Stefan Febri Santoso, Raymond Bahana - Universitas Katolik Atma Jaya Jakarta	170
TE-B-07	<b>Pembuatan Sistem Pakar untuk Diagnosa Penyakit pada Anjing</b> Monica Widiasri, S. Kom., Susana Limanto, M.Si. - Universitas Surabaya	183
TE-B-08	<b>Aplikasi Penunjang Keputusan Pemilihan Calon Tenaga Kerja</b> Susana Limanto - Universitas Surabaya	190
TE-B-09	<b>Aplikasi Kernel Dimensional Reduction Untuk Klasifikasi Laporan Penelitian</b> Agus Widodo - BPPT	196
TE-B-10	<b>Pendeteksi Tingkat kepadatan Jalan melalui Canny Edge Detection</b> Budi Hartanto, Sholeh Hadi Setiawan, Chandra Budi Wijoyo - Universitas Surabaya	206
TE-B-11	<b>Perancangan Aplikasi Terintegrasi Bidang Infrastruktur, Tata Ruang dan Lingkungan Hidup</b> Maniah, Ir., M.T. - Universitas Widyatama Bandung	217
TE-B-12	<b>Pengenalan Bahasa Isyarat Tangan Berbasis Deteksi Tepi dan Nilai Eigen</b> Rahmadi Kurnia - Universitas Andalas, Padang	228
TE-B-13	<b>Knowledge Management dan Social Learning</b> Liliana, S.T. - Universitas Surabaya	234
TE-B-14	<b>Representasi Database berbasis Ontologi dengan Resource Description Framework (RDF)</b> Erna Kumalasari Nurnawati, Khabib Mustofa - Institut Sains & Teknologi AKPRIND dan Universitas Gadjah Mada, Yogyakarta	248