

Sistem Cerdas Pencari Tempat Parkir Dengan Algoritma Blob Detection Dan Manhattan Distance

Elga Asfa Erwanto¹, Ardi Sanjaya², Julian Sahertian³

Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
E-mail: ¹elga@hi2.in, ²dersky@gmail.com, ³Juliansahertian@unpkediri.ac.id

Abstrak – Tempat parkir adalah sarana yang cukup penting pada sebuah instansi manapun karena digunakan untuk menitipkan kendaraan dalam jangka waktu yang tidak ditentukan. Pasalnya, hampir semua orang menggunakan kendaraan bermotor untuk bepergian, jadi jika mereka ingin pergi ke suatu tempat pastilah menuju tempat parkir terlebih dahulu. Namun terkadang tempat parkir itu cukup ramai dan diperlukan waktu untuk mencari tempat kosong yang bisa digunakan untuk menitipkan kendaraan, bahkan mungkin jika pengendara menghabiskan lebih dari 15 menit hanya untuk mencari tempat parkir, belum lagi jika ternyata tempat parkir nya penuh dan pengendara harus mencari tempat parkir di luar instansi. Dari permasalahan tersebut peneliti ingin mempercepat proses pencarian tempat parkir dengan menggunakan gabungan dari metode blob detection dan manhattan distance untuk mendeteksi tempat parkir yang kosong. Kemudian hasil dari deteksi nantinya di tampilkan pada website. Website inilah yang akan digunakan pengendara untuk melihat kondisi tempat parkir, jadi pengendara bisa langsung menuju tempat yang sekiranya kosong dan bisa digunakan untuk parkir tanpa harus memutar tempat parkir terlebih dahulu. Dari penelitian ini akurasi yang didapat mencapai 100% pada keadaan terang dan 60 saat pencahayaan kurang.

Kata Kunci — Blob, Detection, Distance, Manhattan, Opencv

1. PENDAHULUAN

Tempat parkir adalah sarana yang sangat penting dalam berbagai instansi untuk tempat penitipan sementara kendaraan bermotor karena kendaraan bermotor tidak mungkin akan di parkir di jalan luar instansi yang nantinya dapat menyebabkan kemacetan tambahan, apalagi jika jalan yang ada di luar instansi tidak begitu besar. Luas tempat parkir dan jumlah kendaraan yang bisa ditampung juga beraneka ragam, jumlah kendaraan bermotor yang ada pada tempat parkir juga bisa menjadi sebuah kendala untuk pengendara lain yang ingin memarkir kendaraan pribadinya. Pasalnya, untuk melakukan parkir pengendara bebas memilih tempat acak sesuai dengan keinginan mereka, walaupun jika pengendara dapat melihat rekaman kamera CCTV, mereka membutuhkan kuota internet yang cukup banyak untuk melakukan streaming dan hardware dengan spesifikasi tinggi agar rekaman tidak putus - putus.

Saat ini teknologi bisa dimanfaatkan untuk pengolahan citra digital yang memungkinkan pengendara mencari tempat parkir dengan lebih cepat. Pengolahan citra adalah “suatu teknik yang digunakan untuk memproses dan memanipulasi sebuah citra digital untuk mendapatkan informasi tertentu dari citra yang diproses” [1]. Teknik ini digunakan untuk mendeteksi tempat kosong yang nantinya bisa digunakan untuk parkir kendaraan bermotor dengan cara mengambil citra digital

kendaraan pada lahan parkir menggunakan kamera, kemudian diolah oleh komputer yang akan menghasilkan data berupa tempat parkir yang telah terisi dan tempat parkir yang masih kosong. Pada penelitian sebelumnya [2] tentang Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection berhasil mendeteksi kendaraan bermotor di persimpangan jalan, jadi *Blob Detection* digunakan peneliti saat ini untuk proses pendeteksi kendaraan. Kemudian pada penelitian yang dilakukan oleh mahasiswa Universitas Nusantara PGRI Kediri [3] tentang Sistem Penghitung Kendaraan di Jalan Raya Menggunakan Metode *Blob Tracking* dengan *Euclidean Distance* sebagai metode klasifikasi dan aplikasinya berhasil menghitung jumlah kendaraan yang ada pada jalan raya. Dari referensi yang ditulis sebelumnya peneliti saat ini ingin mencoba mencari tempat parkir kosong melalui kamera CCTV dan nantinya hasil data yang telah diolah akan ditampilkan di halaman website agar pengendara bisa melihat tempat parkir mana yang akan digunakan.

Penelitian ini nantinya diharapkan akan mempermudah dan mempercepat pengendara dalam mencari tempat kosong yang dapat digunakan untuk parkir sehingga waktu pengendara bermotor tidak habis dipakai hanya untuk mengitari tempat parkir. Hasil dari aplikasi ini diharapkan tidak akan memberati perangkat yang digunakan pengendara untuk mencari tempat parkir karena pengendara

tidak memerlukan streaming sehingga tidak akan mengurangi banyak kuota internet.

1.1 Batasan Masalah

Batasan Masalah dari penelitian yang dilakukan adalah :

- a) Data yang digunakan adalah rekaman CCTV yang didapat dari Internet
- b) Metode yang digunakan adalah *Blob Detection* dan *Manhattan Distance*
- c) Bahasa pemrograman yang digunakan adalah Python, PHP serta jQuery AJAX
- d) Menggunakan Library pendukung Numpy, OpenCV untuk Python

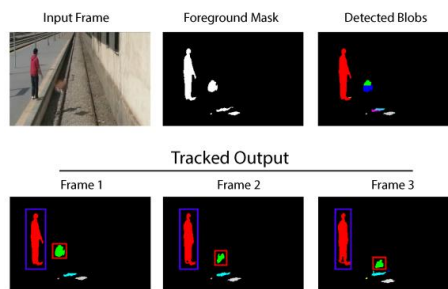
1.2 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah untuk mempermudah pengendara dalam mencari tempat parkir serta menghemat kuota internet yang digunakan, karena dalam *website* yang ditampilkan bukan rekaman kamera CCTV secara langsung, melainkan hanya menampilkan data parkir berbentuk balok yang lebih ringan namun sudah mempresentasikan data yang ada dalam rekaman kamera CCTV.

1.3 Landasan Teori

a) Blob Detection

Blob detection adalah metode yang digunakan untuk mendeteksi pola yang akan menghasilkan objek dengan warna terang dan background dengan warna gelap. Jadi dalam Blob Detection, objek hasil akan dipisah dengan background untuk memudahkan perhitungan jarak yang akan dilakukan, objek dengan warna terang yang didapat dari metode blob detection biasa disebut blob. Untuk mengenali sebuah pola yang ada pada blob detection, dibutuhkan algoritma tambahan yang berguna untuk memberi label dari tiap blob yang terdeteksi, algoritma tersebut adalah blob labeling. Blob labeling digunakan untuk memberi label pada tiap blob, jadi tiap blob nantinya akan memiliki label yang berbeda-beda dan tiap label tidak boleh sama, bisa dikatakan blob labeling sama seperti primary key yang ada pada sebuah database dimana suatu data memiliki nilai unik yang tidak boleh sama satu sama lain.

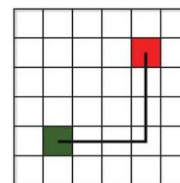


Gambar 1. Contoh Deteksi Blob

Pada blob detection objek yang ukurannya tidak sesuai bisa hilang, juga 2 objek yang berdekatan bisa dianggap sebagai 1 objek. Objek yang sudah dideteksi warnanya bisa disesuaikan untuk membedakan antara objek satu dengan yang lainnya.

b) Manhattan Distance

Manhattan distance atau yang biasa disebut city block merupakan salah satu dari banyak metode yang digunakan untuk menghitung jarak. city block distance menghitung nilai perbedaan absolut dari 2 vektor. Dikatakan nilai absolut karena hasil perhitungannya pasti menghasilkan nilai positif[4].



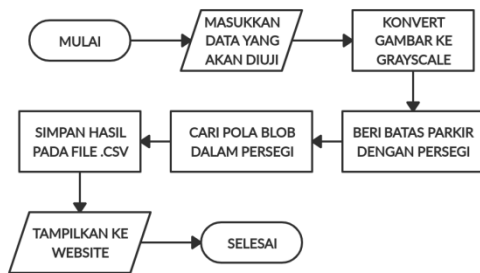
Gambar 2. Skema Pencarian Jarak pada Manhattan Distance

Berikut rumus yang digunakan untuk mencari jarak menggunakan *Manhattan Distance* :

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}| \dots\dots\dots(3)$$

2. METODE PENELITIAN

Pada sistem lama masih menampilkan hasil dari kamera CCTV dan dirasa memberati sistem yang membukanya, juga mengurangi banyak data internet dari user karena model nya streaming, sedangkan dalam sistem yang saat ini dibuat peneliti hanya menampilkan halaman website tanpa harus streaming, dan hasilnya mempresentasikan apa yang ada pada cctv. Sehingga user tidak memerlukan spesifikasi sistem yang tinggi dan tidak harus memiliki koneksi yang mumpuni.



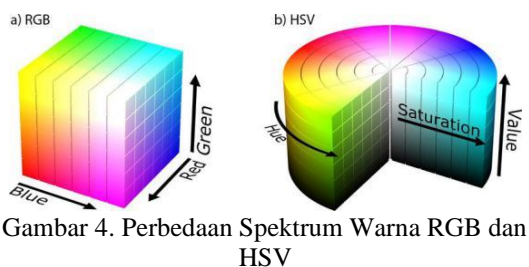
Gambar 3. Alur Kerja Aplikasi

2.1 Pengambilan Data Input

Data yang akan digunakan dalam penelitian ini berasal dari video hasil rekaman cctv yang diambil dari internet, kemudian hasil rekaman video tersebut akan dipecah menjadi beberapa frame dan akan mendapatkan data berupa gambar dari video yang sudah dipecah. Data yang sudah didapat tadi yang nantinya akan diolah dan ditampilkan pada website.

2.2 Transform Warna dari RGB ke HSV

Setiap hasil gambar atau video pasti berupa warna RGB (Red, Green, Blue) karena sudah menjadi format default yang sudah di akui internasional, namun untuk pengolahan data RGB dirasa kurang bagus dan warna yang dihasilkan kurang cocok untuk mata manusia. Karena itu gambar yang nantinya diolah harus dijadikan HSV agar objek lebih mudah dikenali.



Gambar 4. Perbedaan Spektrum Warna RGB dan HSV

Untuk pengaturan warna HSV peneliti menggunakan kombinasi bawah dan atas berturut-turut (79,68,60) dan (229,165,255) sehingga akan menghasilkan gambar seperti berikut



Gambar 5. Tampilan Gambar Setelah Konversi

2.3 Pemberian Batasan

Sebelum kendaraan yang ada pada tempat parkir bisa dikenali, harus diberikan batasan yang membedakan antara jalan dan tempat parkir. Jika tidak diberikan batasan, nantinya jalan yang sering dilewati banyak kendaraan bisa dimasukkan kategori tempat parkir. Untuk pemberian batas digunakan

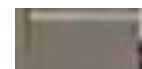
perintah *rectangle* yang sudah disediakan oleh library *OpenCV*.



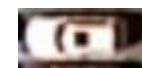
Gambar 6. Pemberian Batas Dengan Persegi Panjang Berwarna Biru

2.4 Pengenalan Blob

Daerah yang akan di olah nantinya cuma daerah dalam persegi panjang berwarna biru dan mengabaikan daerah sekitar yang tidak diberi tanda. Jika blob di dalam area persegi panjang mencakup lebih dari 80% maka dianggap tempat parkir kosong dan persegi panjang akan berubah warna menjadi hijau, sedangkan untuk tempat parkir dengan nilai blob kurang dari 80% maka akan dianggap ada objek yang menempati kemudian persegi panjang akan berubah warna merah. Untuk mencari persentase blob yang ada dalam area persegi panjang digunakan *manhattan distance* dimana objek yang dikenali akan dibandingkan dengan objek yang ada pada data latih.



Gambar 7. Contoh Data Latih Untuk Tempat Parkir Kosong



Gambar 8. Contoh Data Latih Untuk Tempat Parkir Yang Sudah Ditempati

Objek yang menempati tempat parkir disini tidak hanya spesifik pada mobil/kendaraan bermotor lainnya karena terkadang gerobak pun bisa di letakkan pada tempat parkir.



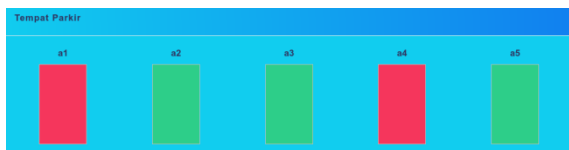
Gambar 9. Tampilan Blob yang Sudah Dikenali

Data yang sudah diolah akan dimasukkan ke dalam file dengan format *.csv* (*comma separated value*) untuk ditampung sementara. Format

penulisan pada file .csv adalah [nama,status] nama disini berarti label pada persegi panjang, ditulis a1,a2,dan seterusnya. Sedangkan status berarti penjelas apakah daerah dalam persegi panjang terisi atau kosong, ditulis 1 jika tempat parkir terisi dan 0 apabila kosong. Contoh penulisan dala file .csv adalah "a1,0". Data di dalam file .csv ini yang nantinya digunakan untuk acuan tempat parkir yang nantinya tampil pada website.

2.5 Tampilkan Pada Website

Website akan membaca data yang sudah ada pada file .csv setelah proses pengenalan blob, tampilan website dibuat sesederhana mungkin sehingga user tidak menghabiskan banyak kuota serta lebih cepat dalam memuat data karena data dalam file .csv akan berubah terus selama program berjalan. Tampilan persegi panjang yang ada pada form website mempresentasikan tempat parkir, warna hijau artinya tempat parkir kosong dan merah berarti terisi. Untuk memuat ulang tampilan tempat parkir secara otomatis digunakan library tambahan yaitu AJAX Jquery sehingga user tidak harus memuat ulang website untuk mengetahui kondisi terbaru yang ada pada tempat parkir.



Gambar 10. Tampilan Website

3. HASIL DAN PEMBAHASAN

3.1 Skenario Uji Coba

Untuk menguji seberapa akurat dan seberapa cepat program yang telah dibuat dapat menentukan tempat parkir, akan dilakukan ujicoba dengan konsentrasi warna, kecerahan, dan jenis kendaraan yang berbeda. Data akan disajikan dalam bentuk tabel untuk mempermudah dalam penguraian dan pengambilan kesimpulan. Penghitungan akurasi dilakukan dengan membandingkan antara jumlah tempat parkir yang berhasil dideteksi oleh sistem dan tempat parkir yang dideteksi oleh mata manusia.

Dalam hasil pengujian rumus yang digunakan untuk mencari akurasi dan tingkat kesalahan (*error*) adalah :

$$\text{Akurasi} = \frac{\text{Jumlah Berhasil}}{\text{Jumlah Uji coba}} \times 100\% \dots\dots\dots (4)$$

$$\text{Error} = 100\% - \text{Akurasi} \dots\dots\dots (5)$$

3.2 Hasil Pengujian

a) Pengujian Pertama

Menggunakan kombinasi warna HSV (79,68,60) dan (229,165,255) hasil yang didapat dari pengujian ini adalah banyaknya tempat parkir yang dapat dideteksi oleh sistem.

Tabel 1 Hasil Dari Pengujian Pertama

Kondisi	Jumlah Tempat Parkir Yang Dideteksi		Akurasi	Error
	Manusia	Sistem		
Siang	7	7	100%	0%
Sore	9	4	44%	56%
Malam	11	2	18%	82%

Akurasi Keseluruhan = 48%

Error Keseluruhan = 52%

Dari hasil pengujian pertama dapat dilihat bahwa akurasi pada siang hari sangat baik mencapai 100% karena pencahayaan cukup terang sehingga tempat parkir mudah dideteksi. Sedangkan pada sore dan malam hari akurasi turun drastis karena pencahayaan yang ada pada tempat parkir sangat minim dan sistem hanya bisa mendeteksi tempat parkir yang ada disekitar lampu penerangan.

b) Pengujian Kedua

Menggunakan kombinasi warna HSV (98,89,50) dan (229,152,255), video yang digunakan sama seperti pada pengujian pertama

Tabel 2 Hasil Pengujian Kedua

Kondisi	Jumlah Tempat Parkir Yang Dideteksi		Akurasi	Error
	Manusia	Sistem		
Siang	7	7	100%	0%
Sore	9	5	55%	45%
Malam	11	5	45%	55%

Akurasi Total = 62%

Error Total = 38%

Untuk akurasi pada siang hari masih tetap sama dengan pengujian pertama, namun akurasi pada sore dan malam hari bertambah karena

pencahayaan disekitar lampu penerangan menjadi lebih luas akibat pergantian seleksi warna.

a. Confusion Matriks

Pengujian ini bertujuan untuk mengevaluasi aplikasi dan digunakan untuk mencari presisi,recall, *f1-Score* dan akurasi pada *blob detection* berdasarkan *predictive value theory*. Nilai yang akan diambil yaitu *True Positive* (TP) artinya Tempat Parkir Kosong dan sistem mengenali nya sebagai tempat kosong, *True Negative* (TN) artinya Tempat Parkir Kosong tapi sistem menganggap ada yang menempati, *False Positif* (FP) artinya tempat parkir sudah ada yang menempati tapi sistem menganggapnya kosong, *False Negatif* (FN) artinya tempat parkir ada yang menempati dan sistem mengenali nya sebagai tempat yang sudah ditempati.

Presisi dapat disebut juga prediksi positif, atau banyaknya data positif yang ada pada perhitungan confusion matriks tanpa melihat apakah itu data benar/salah. Recall adalah data benar yang ada pada perhitungan confusion matriks artinya dalam kasus ini jika tempat parkir kosong maka sistem akan menyebutnya kosong (*Trup Positif*) dan jika tempat parkir terisi maka sistem akan mengatakan tempat parkir ada yang menempati(*False Negatif*). *F1-Score* adalah perhitungan yang digunakan untuk melihat keseimbangan antara data presisi dan recall, peneliti menganggap *f1-score* penting karena data positif dari presisi harus berjarak sedekat mungkin dengan data positif yang sebenarnya dari recall. Perlu diketahui bahwa data akurasi yang ada pada confusion matriks berbeda dari data positif yang ada pada pengujian sebelumnya, karena dari pengujian sebelumnya hanya mencari berapa banyak tempat parkir yang dapat dideteksi sistem tanpa terkecuali, sedangkan pada confusion matriks mempertimbangkan juga tempat parkir yang tidak dideteksi dan tempat parkir yang ditempati.

Untuk rumus yang digunakan untuk mencari Presisi, *f1-Score* dan akurasi antara lain :

$$\text{Presisi} = \frac{TP}{TP+FP} \dots\dots\dots (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \dots\dots\dots (7)$$

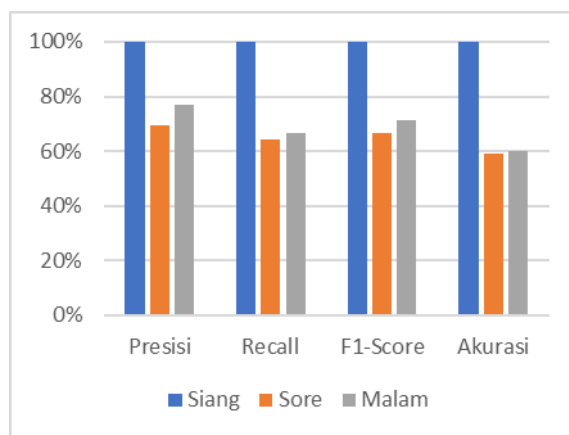
$$\text{F1 - Score} = \frac{2xTP}{2TP+FP+FN} \dots\dots\dots (8)$$

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (9)$$

Tabel 3. Tabel Confussion Matriks

	SIANG	SORE	MALAM
TP	14	9	10

TN	8	4	2
FP	0	4	3
FN	0	5	5
PRESISI	100%	69%	77%
RECALL	100%	64%	67%
F1-SCORE	100%	67%	71%
AKURASI	100%	59%	60%



Gambar 11. Grafik Perhitungan Confusion Matriks

Tabel dan grafik diatas menggambarkan performa keseluruhan dari aplikasi yang dibuat dengan acuan pengujian pertama dan kedua yang sudah dilakukan sebelumnya. Pengujian hanya dilakukan dua kali karena peneliti menyimpulkan *blob detection* tidak dapat mendeteksi objek dengan pencahayaan yang minim walaupun warna kendaraan terang.

Berikut tampilan hasil dari aplikasi yang sudah dibuat saat mendeteksi tempat parkir kosong:



Gambar 12. Tampilan Saat Siang Hari

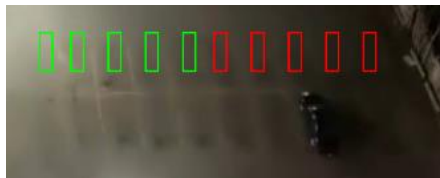
Bisa dilihat aplikasi dapat mengenali semua tempat parkir yang kosong dan memberikan label berupa persegi panjang berwarna hijau. Selain itu, tempat parkir yang terisi mobil juga dapat dikenali

dan aplikasi memberikan label persegi panjang berwarna merah.



Gambar 13. Tampilan Saat Sore Hari

Saat sore hari karena warna latar yang dijadikan acuan untuk tempat parkir berubah, jadi warna mobil yang terlihat terang dianggap sebagai tempat parkir kosong, sedangkan tempat parkir yang kosong karena terkena bayangan dari kendaraan disebelahnya jadi terlihat lebih gelap dan dianggap ada yang menempati



Gambar 14. Tampilan Saat Malam Hari

Saat malam hari hanya tempat parkir yang dekat dengan lampu yang dapat dikenali sebagai tempat kosong karena mendapat pencahayaan yang cukup dan *background* nya jadi terang, selain itu tempat parkir kosong lainnya dianggap ada yang menempati karena warnanya yang gelap.

4. KESIMPULAN

Dari hasil penelitian yang dilakukan, aplikasi dapat bekerja dengan sangat baik pada siang hari/ saat intensitas cahaya tinggi dan warna kendaraan atau objek dalam tempat parkir tidak sama dengan warna latar tempat parkir. Website dapat mempresentasikan hasil yang dikirim dari aplikasi pendeteksi tempat parkir dengan sangat baik, walaupun saat tempat parkir dalam kondisi ramai sekalipun. Blob detection juga tidak dapat mengenali spesifik objek dengan baik tanpa dataset yang banyak.

5. SARAN

Dari penelitian yang sudah dilakukan pasti ada kekurangan seperti tidak bisa mendeteksi objek saat keadaan gelap, saran dari peneliti untuk menanggulangi masalah ini antara lain bisa menggunakan background subtraction jadi untuk mendeteksi objek yang bergerak dan memisahkan objek dengan background walaupun warna objek

dan latar belakang mirip. Menggunakan deteksi tepi untuk mendeteksi tempat parkir, jadi tidak harus membuat batasan yang membedakan jalan dan tempat parkir. Menggunakan database seperti MySQL untuk menampung data karena pengolahannya lebih cepat dan baik, file .csv memang dapat menampung sedikit data, tapi untuk menyimpan track record dirasa memberatkan dan pengolahan datanya kurang lengkap.

DAFTAR PUSTAKA

- [1] Jupiyandi, Sisco, Fadhil Rizqullah Saniputra, Yoga Pratama, Muhammad Robby Dharmawan, Imam Cholissodin. 2018. PENGEMBANGAN DETEKSI CITRA MOBIL UNTUK MENGETAHUI JUMLAH TEMPAT PARKIR MENGGUNAKAN CUDA DAN MODIFIED YOLO. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK). Vol.6, No.4: 413-419
- [2] Hidayati, Qory. 2017. Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection. JNTETI. Vol. 6, No. 2: 215-221
- [3] Dewi, Yunita Surya. 2018. "Sistem Penghitung Kendaraan Di Jala Raya Menggunakan Metode Blob Tracking". diakses dari: <http://simki.unpkediri.ac.id/detail/14.1.03.02.0214>, pada 30 Desember 2019
- [4] Putra, Darma. 2010. Pengolahan Citra Digital. Yogyakarta: ANDI