

Deteksi Area Parkir Mobil Berbasis Marker Menggunakan Moment Invariants dan K-NN

<http://dx.doi.org/10.28932/jutisi.v5i1.920>

Kristian Adi Nugraha^{#1}

^{#1}*Informatika, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirohusodo 5-25 Yogyakarta 55224*

¹*adinugraha@ti.ukdw.ac.id*

Abstract — Smart car parking system has become very important these days because it is very helpful and saves a lot of time to find a parking spot. However, the installation of that system is too expensive, then not everyone can afford it. Affordable solution for car parking spot detection can be implemented using installed camera or CCTV combined with computer vision method. New problem appears if the camera position is limited like indoor building causing one marker shape can be captured in different shape due to the different camera angle. In this work, I do a research to detecting marker using moment invariant method. Even the shape of marker changed a little, moment invariant method can still recognize that shape with overall accuracy 91.94%.

Keywords— moment invariants, K-NN, parking area, image processing

I. PENDAHULUAN

Laju pertumbuhan jumlah kendaraan di Indonesia terus meningkat setiap tahunnya. Berdasarkan data Badan Pusat Statistik Tahun 2017, jumlah kendaraan bermotor di seluruh Indonesia mencapai kurang lebih 138 juta kendaraan [1]. Jumlah ini meningkat kurang lebih sebesar 7% dari tahun sebelumnya (2016), dan diperkirakan akan terus meningkat setiap tahunnya. Seiring bertumbuhnya jumlah kendaraan yang ada akan mengakibatkan meningkatnya kebutuhan lahan parkir. Semakin luas wilayah sebuah area parkir, maka pengemudi kendaraan akan semakin sulit menemukan tempat parkir karena harus menjelajahi area parkir yang cukup luas. Pada akhirnya pengemudi kendaraan akan menghabiskan lebih banyak waktu untuk sekedar memarkirkan kendaraannya. Selain itu, semakin lama kendaraan mencari tempat parkir, maka akan menghasilkan lebih banyak polusi udara akibat asap pembuangan yang dihasilkan oleh kendaraan tersebut.

Permasalahan di atas dapat diatasi salah satunya dengan menggunakan sistem parkir cerdas (*smart parking*) yang dapat memberi tahu pengguna letak area parkir yang dapat digunakan untuk memarkirkan kendaraan [2] [3] [4] [5]. Salah satu solusi yang ada adalah dengan memasang sensor khusus yang dipasang pada setiap slot parkir [6] [7] [8].

Sensor ini berfungsi untuk mengetahui apakah slot parkir tersebut sedang terisi atau tidak. Sensor tersebut akan mengirimkan data ke *server* secara berkala untuk memperbarui informasi ketersediaan lahan parkir. Pengguna yang hendak memarkirkan kendaraan cukup melihat informasi ketersediaan lahan parkir tersebut berdasarkan data terbaru yang terdapat pada *server*. Namun solusi tersebut membutuhkan biaya yang cukup besar, tergantung banyaknya slot parkir yang dibutuhkan pada area parkir tersebut. Sehingga solusi tersebut tidak dapat diakomodasi oleh pemilik lahan parkir yang memiliki anggaran terbatas.

Melihat permasalahan tersebut, penulis mencoba mengatasi permasalahan tersebut dengan membuat sistem pendeteksi ketersediaan tempat parkir secara otomatis dengan memanfaatkan citra yang didapatkan melalui kamera pada area parkir. Sumber citra yang akan digunakan tersebut juga dapat memanfaatkan kamera *Closed-Circuit Television (CCTV)* pada area parkir jika telah ada dan terpasang sebelumnya. Sistem ini bekerja dengan memanfaatkan gambar khusus (*marker*) yang dipasang pada tiap slot area parkir. Setiap slot pada area parkir memiliki gambar *marker* yang berbeda-beda agar dapat dibedakan antara slot satu dengan slot yang lain. Apabila sistem mendeteksi adanya gambar *marker* pada sebuah citra, maka bentuk dari *marker* tersebut akan diidentifikasi menggunakan metode *Moment Invariants* dan *K-Nearest Neighbor (K-NN)*. Metode *Moment Invariants* dapat digunakan untuk mengambil nilai ciri atau fitur dari sebuah bentuk *marker*, sedangkan metode *K-NN* digunakan untuk menentukan jenis *marker* tersebut berdasarkan nilai ciri atau fitur yang didapatkan melalui metode *Moment Invariants*. Sistem akan mencari pasangan bentuk *marker* tersebut dengan slot parkir tertentu, kemudian memperbarui informasi bahwa slot tersebut kosong. Apabila sistem tidak mendeteksi adanya *marker* pada sebuah citra, maka dapat dipastikan seluruh slot parkir yang terekam pada citra tersebut telah terisi. Informasi ketersediaan lahan parkir tersebut akan disimpan oleh sebuah *server* khusus yang dapat diakses oleh publik. Pengemudi yang hendak memarkirkan kendaraannya dapat mengakses informasi ketersediaan lahan parkir tersebut menggunakan telepon

seluler masing-masing yang terhubung oleh server melalui aplikasi tertentu atau melalui aplikasi *web browser*. Metode ini menghabiskan lebih sedikit biaya dibandingkan dengan implementasi sensor pada tiap slot parkir, karena sebuah kamera dapat menangkap citra dari banyak slot parkir. Selain itu, jika pada area parkir telah terpasang kamera CCTV, maka tidak perlu dilakukan penambahan kamera khusus, cukup memanfaatkan kamera CCTV yang telah ada sebelumnya. Pada penelitian ini, kendaraan yang menjadi obyek penelitian berfokus pada mobil atau kendaraan roda 4. Penulis tidak meneliti kendaraan lain seperti sepeda motor tidak memiliki slot khusus pada area parkir, sehingga sulit untuk diteliti.

II. TINJAUAN PUSTAKA

Chaudary, Bansal, dan Valarmathi mengembangkan sebuah sistem *Smart Parking* untuk mobil berbasis teknologi *IoT* dengan memanfaatkan komponen *Radio Frequency Identification (RFID)* [9]. Sistem tersebut mengatur pengelolaan lahan parkir berbasis otorisasi. Masing-masing pengemudi akan diberi kartu *RFID* yang berisi seperti informasi pengguna, kendaraan yang digunakan, dan informasi lainnya. Saat pengemudi datang ke area parkir tersebut, pengemudi harus memindai kartu *RFID* miliknya pada mesin yang tersedia. Mesin tersebut akan membaca informasi dari kartu *RFID* tersebut, kemudian sistem akan menentukan apakah slot parkir untuk kendaraan dengan kriteria yang sesuai informasi pada kartu *RFID* tersebut masih tersedia atau tidak. Apabila ternyata slot parkir telah penuh, maka pengemudi tersebut tidak diizinkan untuk memasuki area parkir. Sistem tersebut dapat diimplementasikan dan berjalan dengan baik. Namun pada kondisi temperatur udara yang cukup tinggi, mesin tidak dapat membaca kartu *RFID* dengan baik, sehingga sering terjadi kegagalan.

Penelitian lain yang serupa juga dilakukan oleh Sheelarani, Anand, Shamili, dan Sruthi, yaitu pembuatan sistem parkir berbasis *IoT* dengan memanfaatkan sensor perangkat keras pada slot parkir [10]. Sensor tersebut dipasang pada setiap slot parkir untuk mendeteksi kosong atau tidaknya slot parkir di tempat sensor berada, kemudian memperbarui data tersebut kepada *server* sistem parkir. Pengemudi yang hendak memarkirkan kendaraan dapat mengetahui informasi ketersediaan area parkir melalui aplikasi pada telepon seluler berbasis Android yang terhubung oleh *server* dari sistem parkir tersebut. Sistem tersebut dapat berjalan dengan baik, namun implementasinya memakan cukup banyak biaya khususnya untuk pembelian perangkat sensor pendeteksi ketersediaan lahan parkir. Penelitian serupa juga dilakukan oleh Adhatarao, Alfandi, Bochem, dan Hogrefe dengan membangun sistem parkir cerdas dengan menggunakan *Wireless Network Sensors (WSN)* [11]. Sensor tersebut merupakan sensor tanpa kabel yang memanfaatkan jaringan radio untuk mendeteksi adanya ketersediaan lahan parkir. Sistem tersebut berhasil diimplementasikan dan memudahkan para pengemudi yang hendak memarkirkan kendaraannya di area parkir.

Penelitian yang dilakukan oleh penulis memiliki pendekatan berbeda dibandingkan penelitian-penelitian lain di bidang sistem parkir cerdas. Apabila pada penelitian-penelitian terdahulu pada tabel I memanfaatkan perangkat keras khusus berupa sensor untuk mendeteksi ketersediaan lahan parkir, maka penulis mencoba mengambil solusi berbeda dengan cara memanfaatkan perangkat kamera CCTV. Solusi ini menghabiskan lebih sedikit biaya karena pada umumnya di beberapa area parkir telah terpasang CCTV, sehingga tidak perlu melakukan penambahan alat baru. Selain itu, sebuah kamera dapat menangkap banyak slot area parkir, tidak seperti sensor pendeteksi yang harus dipasang pada setiap slot area parkir.

TABEL I

PERBEDAAN PENELITIAN AKTUAL DENGAN PENELITIAN TERDAHULU

Judul	<i>Smart parking system for vehicles</i>	<i>Effective car parking reservation system based on internet of things technologies</i>	<i>Advanced CAR parking system using Arduino</i>	Deteksi Area Parkir Mobil Berbasis Marker Menggunakan Moment Invariants dan <i>K-NN</i>
Oleh	Sripriya Srikant Adhatarao, Omar Alfandi, Arne Bochem, Dieter Hogrefe	P. Sheelarani, S. Anand, Preethi, S. Shamili, K. Sruthi	Hemant Chaudhary, Prateek Bansal, B. Valarmathi	Kristian Adi Nugraha
Tahun	2014	2016	2017	2019
Metode	<i>Wireless Sensor Networks</i>	<i>IoT dan Sensor</i>	<i>RFID dan Arduino</i>	<i>Moment Invariants</i> dan <i>K-NN</i>
Obyek	Area parkir	Area parkir	Area parkir	Marker pada citra area parkir
Bahasa	Inggris	Inggris	Inggris	Indonesia

III. LANDASAN TEORI

A. Citra Aras Keabuan (Grayscale)

Citra aras keabuan atau *grayscale* merupakan jenis citra yang hanya terdiri dari satu lapisan kumpulan piksel dengan nilai antara 0 sampai dengan 255 untuk masing-masing piksel. Citra aras keabuan dapat diperoleh melalui citra berwarna yang memiliki tiga lapisan piksel merah, hijau, dan biru, kemudian nilai tiap piksel dihitung menggunakan rumus 1 untuk mendapatkan nilai aras keabuannya [12].

$$\text{gray} = (0.299 * R + 0.587 * G + 0.114 * B) \quad [1]$$

Dengan,

R = nilai intensitas warna merah

G = nilai intensitas warna hijau

B = nilai intensitas warna biru

B. Binarisasi

Binarisasi adalah proses mengubah citra aras keabuan menjadi citra hitam putih yang memiliki nilai biner, yaitu 0 atau 1 [13]. Binarisasi dilakukan dengan cara melihat nilai piksel setiap elemen pada citra aras keabuan. Apabila nilai piksel tersebut berada di bawah nilai ambang, maka nilai piksel tersebut berubah menjadi 0. Sebaliknya jika nilai piksel tersebut berada di atas nilai ambang, maka nilai piksel tersebut berubah menjadi 1. Pada umumnya, nilai ambang bernilai 127, yang merupakan nilai tengah antara 0 sampai 255. Nilai ambang tersebut dapat diubah sesuai kebutuhan masing-masing.

C. Segmentasi Warna

Segmentasi warna adalah proses pemisahan elemen pada sebuah piksel menjadi beberapa komponen yang memiliki informasi berbeda. Umumnya segmentasi warna dilakukan dengan cara mengubah informasi nilai lapisan RGB (*Red, Green, Blue*) pada citra menjadi nilai HSV (*Hue, Saturation, Value*), kemudian segmentasi dilakukan berdasarkan nilai HSV yang dikehendaki [14]. Proses segmentasi warna untuk mengubah nilai RGB menjadi nilai HSV ditunjukkan pada rumus 2.

$$H = \begin{cases} 0, & \text{if } Max = Min \\ \left(60^\circ \times \frac{G - B}{Max - Min} + 360^\circ\right) \times \text{mod } 360^\circ, & \text{if } Max = R \\ \left(60^\circ \times \frac{B - R}{Max - Min} + 120^\circ\right), & \text{if } Max = G \\ \left(60^\circ \times \frac{R - G}{Max - Min} + 240^\circ\right), & \text{if } Max = B \end{cases} \quad [2]$$

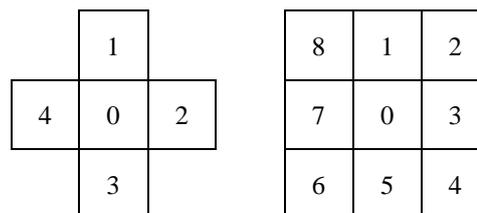
$$S = \begin{cases} 0, & \text{if } max = 0 \\ \frac{Max - Min}{Max} & \text{otherwise} \end{cases}$$

$$V = Max$$

Nilai H menunjukkan tipe warna sebuah piksel dalam satuan derajat. Nilai S menunjukkan tingkat intensitas warna putih terhadap nilai H. Sedangkan nilai V menunjukkan tingkat kecerahan sebuah piksel.

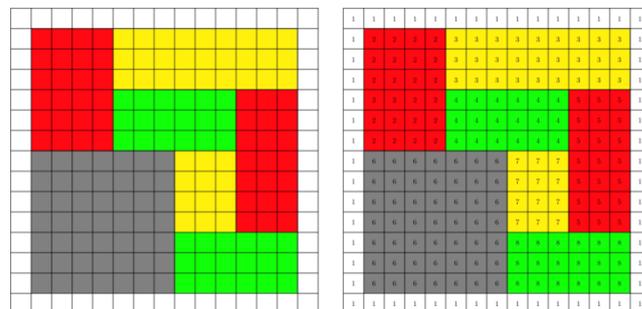
D. Connected-Component Labeling (CCL)

Connected-Component Labeling atau *CCL* merupakan proses pelabelan pada piksel-piksel yang merupakan satu kelompok [15]. Terdapat dua macam tipe *CCL*, yaitu 4 arah dan 8 arah seperti ditunjukkan pada gambar 1.



Gambar 1. CCL 4-arah dan 8-arah

Pada *CCL* 4-arah, sebuah piksel hanya akan menelusuri piksel lain yang berada pada sisi atas, kanan, bawah, dan kiri dari piksel tersebut. Sedangkan pada *CCL* 8-arah, sebuah piksel akan menjelajahi 4 arah yang sama seperti *CCL* 4-arah, ditambah dengan nilai diagonal dari keempat sudut yang ada pada piksel tersebut. Contoh simulasi hasil dari proses *CCL* ditunjukkan pada gambar 2.



Gambar 2. Simulasi hasil proses CCL

Dikutip dari: Schwenk & Huber, 2015, Connected Component Labeling algorithm for very complex and high-resolution images on an FPGA platform [15]

Pada penelitian ini, metode *CCL* digunakan untuk memisahkan obyek di dalam *marker* dari latar belakang *marker* itu sendiri.

E. Moment Invariants

Moment Invariants merupakan metode ekstraksi fitur yang merepresentasikan bentuk dari sebuah obyek ke dalam tujuh nilai angka [16]. Masing-masing nilai merepresentasikan karakteristik bentuk sebuah obyek.

Kelebihan utama dari metode *Moment Invariants* adalah metode ini memiliki nilai fitur yang tidak terpengaruh terhadap posisi, skala, maupun rotasi sebuah obyek. Sehingga apabila terdapat dua buah obyek dengan bentuk serupa, namun memiliki perbedaan di sisi posisi, skala, maupun rotasi, maka nilai fiturnya akan tetap sama.

Moment Invariants ditemukan pertama kali oleh Hu pada tahun 1961, persamaannya ditunjukkan pada rumus 3.

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y) \Delta A \quad [3]$$

Dengan,

M_{pq} = moment citra

p, q = orde momen

$I(x, y)$ = nilai intensitas pada koordinat piksel

ΔA = area dari piksel

Melalui rumus 1, dapat dihitung nilai translasi *invariant* seperti ditunjukkan pada rumus 4.

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \Delta A \quad [4]$$

Dengan,

$$\bar{x} = \frac{M_{10}}{M_{00}}, \text{ dan } \bar{y} = \frac{M_{01}}{M_{00}}$$

μ = momen pusat

\bar{x} = pusat sumbu x

\bar{y} = pusat sumbu y

Dari nilai translasi *invariant* pada rumus 4, dapat dihitung nilai *central moment* menggunakan rumus 5 dan 6.

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{pq})^\gamma} \quad [5]$$

Dengan,

$$\gamma = \frac{p + q}{2} + 1 \quad [6]$$

Melalui nilai *central moment* pada rumus 5 dan 6, dapat dihitung nilai ketujuh *moment invariants* dengan menggunakan rumus 7. Ketujuh nilai *Moment Invariants* yang didapatkan pada rumus 7 akan diolah lebih lanjut menggunakan metode *K-NN*.

$$M1 = \eta_{20} + \eta_{02}$$

$$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) +$$

$$((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2) +$$

$$3(\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 -$$

$$(\eta_{21} - \eta_{03})^2) \quad [2.7]$$

$$M6 = \quad [7]$$

$$(\eta_{20} + \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) +$$

$$4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03})$$

$$M7 = (3\eta_{21} - \eta_{03})(\eta_{30} - \eta_{12})((\eta_{30} + \eta_{12})^2$$

$$+ 3(\eta_{21} + \eta_{03})^2)$$

$$+ (3\eta_{12} - \eta_{30})(\eta_{21}$$

$$+ \eta_{03})(3\eta_{12} + \eta_{30})^2$$

$$- (\eta_{21} + \eta_{03})^2)$$

F. K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) merupakan sebuah metode klasifikasi yang dapat digunakan untuk mengklasifikasikan sebuah obyek ke dalam kategori tertentu berdasarkan kedekatan obyek tersebut terhadap sejumlah obyek lain sebanyak K pada data latih yang tersedia [17]. Untuk menghitung jarak terdekat antara obyek satu dengan obyek yang lain dapat menggunakan rumus Euclidean Distance seperti yang ditunjukkan pada rumus 8.

$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad [8]$$

Dengan,

k = jumlah fitur

x_i = nilai obyek pertama

y_i = nilai obyek kedua

Pada penelitian ini, *K-NN* digunakan untuk menghitung jarak antara ketujuh nilai *Moment Invariants* dari sebuah *marker* terhadap data seluruh *marker* yang terdapat pada data latih.

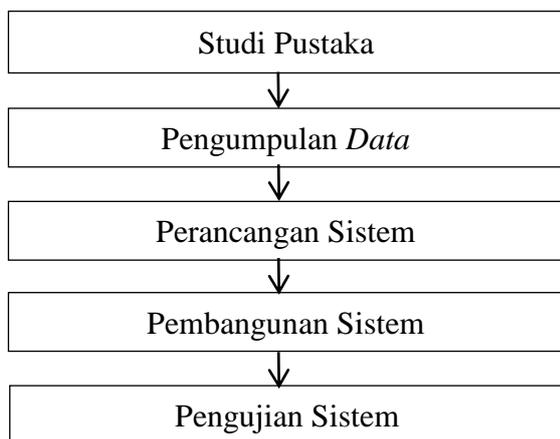
IV. METODE PENELITIAN

Penelitian yang akan dilakukan oleh penulis terdiri dari enam tahap seperti yang ditunjukkan pada Gambar 3.

Adapun penjelasan masing-masing tahapan adalah sebagai berikut:

A. Studi Pustaka

Studi mengenai metode *Moment Invariants* dan *K-NN* dilakukan agar dapat diimplementasikan ke dalam sistem yang dibangun. Penulis mempelajari metode tersebut berdasarkan buku teks mengenai pengolahan citra (*image processing*) dan kecerdasan buatan (*artificial intelligence*).



Gambar 3. Langkah Penelitian

B. Pengumpulan Data

Pengumpulan data dilakukan pada area parkir *basement* Universitas Kristen Duta Wacana. Proses pengumpulan data dilakukan pada saat kondisi area parkir cukup padat, yaitu pada pukul 8.00 hingga pukul 15.00 mulai dari hari Senin hingga hari Jumat selama kurang lebih dua minggu. Penulis mengambil citra dari beberapa kamera *CCTV* yang telah terpasang pada area parkir. Citra dari kamera tersebut diambil dalam berbagai kondisi kendaraan dengan variasi kondisi pencahayaan, jumlah kendaraan, ukuran kendaraan, warna kendaraan, dan posisi kendaraan yang sedang menempati area parkir.



Gambar 4. Contoh citra kamera *CCTV*. Kiri: pagi hari; kanan: siang hari.

Pada gambar 4 terdapat dua contoh citra kamera *CCTV* yang telah dikumpulkan. Kedua citra tersebut diambil pada lokasi dan waktu yang berbeda. Citra sebelah kiri diambil pada waktu pagi, sehingga kondisi cahaya matahari belum terlalu terang. Sedangkan citra sebelah kanan diambil pada waktu siang hari, ketika matahari berada tepat di atas

gedung, sehingga citra yang dihasilkan menjadi cenderung lebih terang dibandingkan citra di sebelah kiri. Jumlah data yang digunakan sebagai data uji adalah 30 data untuk masing-masing jenis marker, sehingga jumlah total keseluruhan data uji adalah 90 data.

C. Perancangan Sistem

Setelah mengumpulkan data, penulis melakukan perancangan sistem. Perancangan sistem meliputi perancangan basis data untuk data latih serta perancangan program yang akan dibangun. Beberapa parameter yang perlu diperhatikan dalam perancangan basis data adalah nilai-nilai fitur yang akan disimpan di dalamnya

D. Pembangunan Sistem

Sistem dibangun dengan menggunakan bahasa pemrograman *C#* dengan framework *.NET*. *Platform desktop* dipilih untuk memudahkan komunikasi antara sistem dengan perangkat kamera yang digunakan untuk mengambil gambar di lokasi area tempat parkir.

E. Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan tiga kategori bentuk *marker*, yaitu *marker* dengan bentuk bangun datar (gambar 5), bentuk sederhana yang terdiri dari huruf alfabet (gambar 6), dan simbol yang sedikit rumit menggunakan huruf mandarin (gambar 7). Tiap kategori bentuk marker terdiri dari empat macam simbol dengan asumsi dalam sebuah citra maksimal hanya terdapat empat buah slot parkir. Untuk tiap kategori bentuk akan diambil sebanyak 30 citra sebagai data pengujian. Proses pengujian metode *K-NN* dilakukan untuk nilai *K* sebesar 1, 3, dan 5. Nilai *K* yang digunakan merupakan nilai ganjil dengan pertimbangan agar *K*-kelas yang dihasilkan oleh metode *K-NN* tidak dapat dibagi dua secara merata, sehingga hasilnya akan lebih terpusat pada satu kelas saja. Jumlah data latih yang digunakan untuk masing-masing bentuk adalah 5 data, sehingga nilai *K* maksimal yang digunakan adalah 5.



Gambar 5. Bentuk bangun datar



Gambar 6. Bentuk alphabet

市夫车差

Gambar 7. Bentuk simbol huruf mandarin

Bentuk dan karakter yang digunakan pada *marker* dipilih berdasarkan peluang munculnya bentuk dan karakter tersebut pada obyek lain di area parkir, misalnya pada badan mobil atau pada papan tulisan. *Marker* tersebut akan dicetak pada sebuah kertas dengan latar belakang hijau. Warna tersebut dipilih karena jarang dijumpai mobil dengan warna hijau seperti yang digunakan pada *marker*. Hal bertujuan untuk mempermudah proses pendeteksian lokasi area parkir.

F. Analisis Hasil Pengujian

Setelah dilakukan pengujian terhadap seluruh data uji yang ada, penulis melakukan analisis terhadap hasil yang diperoleh guna mengetahui tingkat akurasi dari metode *Moment Invariants* dan *K-NN*. Akurasi dihitung berdasarkan hasil deteksi slot parkir kosong menurut sistem, dibandingkan dengan keadaan aktual di area parkir. Proses analisis juga dilakukan untuk mengetahui kelebihan serta kekurangan dari metode yang telah diterapkan sebagai referensi pada penelitian berikutnya atau penelitian lain yang menggunakan metode sejenis.

V. ANALISIS DAN PEMBAHASAN

Sistem yang telah dibangun mendapatkan *input* citra dari kamera *CCTV* yang terpasang pada area parkir seperti ditunjukkan pada gambar 8. Citra hasil proses pemotongan dari gambar 8 ditunjukkan pada gambar 9.



Gambar 8. Citra kamera CCTV

Citra tersebut akan dipotong (*cropping*) terlebih dahulu untuk mengurangi informasi yang tidak digunakan serta mempercepat proses utama pada sistem.



Gambar 9. Citra hasil pemotongan

Dari citra yang telah dipotong, dilakukan proses pengolahan warna dari citra *RGB* menjadi citra *HSV* seperti ditunjukkan pada gambar 10. Proses ini bertujuan untuk mengambil gambar *marker* dengan latar belakang hijau



Gambar 10. Citra HSV

Setelah citra *RGB* diubah menjadi citra *HSV*, proses selanjutnya adalah mencari nilai *Hue* yang sesuai dengan warna *marker* yang telah ditentukan sebelumnya, yaitu warna hijau. Nilai *Hue* warna hijau digunakan untuk proses filterisasi pada citra tersebut untuk mencari *region of interest*, sehingga gambar *marker* dengan nilai *Hue* yang sesuai akan dapat terdeteksi seperti ditunjukkan pada gambar 11.



Gambar 11. Citra hasil filterisasi

Pada citra hasil filterisasi pada gambar 11, dilakukan proses pemotongan pada obyek *marker* yang berhasil terdeteksi, yaitu obyek yang ditandai dengan warna putih. Dari proses tersebut, dihasilkan satu atau lebih gambar *marker* yang berhasil terdeteksi seperti ditunjukkan pada gambar 12. Apabila citra tidak mendeteksi adanya *marker* sama sekali, maka dapat disimpulkan bahwa seluruh area parkir yang tertangkap pada citra tersebut telah penuh.



Gambar 12. Citra marker yang telah dipotong



Gambar 13. Citra marker yang diproses menggunakan CCL



Gambar 14. Bentuk akhir citra marker yang terdeteksi

Setelah citra *marker* berhasil didapatkan, maka proses berikutnya adalah dengan melakukan CCL pada citra *marker* tersebut (gambar 13), kemudian dilanjutkan dengan proses pemilihan obyek di dalam wilayah berwarna putih dan dilakukan proses negasi pada citra, sehingga dihasilkan citra akhir seperti pada gambar 14. Dari citra akhir tersebut, dilakukan proses perhitungan nilai *Moment Invariants* dan dihasilkan nilai seperti ditunjukkan pada tabel II.

TABEL III

NILAI MOMENT INVARIANT LINGKARAN DAN PERSEGI

	Lingkaran	Persegi
Hu 1	0.269055	0.218966
Hu 2	0.046651	0.0203426
Hu 3	3.05504e-006	8.13026e-006
Hu 4	3.08947e-007	1.31378e-006
Hu 5	2.8427e-013	3.48233e-012
Hu 6	5.40294e-008	1.32639e-007
Hu 7	-9.63274e-014	2.51187e-012

Nilai *Moment Invariants* seperti yang ditunjukkan pada tabel II akan dihitung menggunakan metode *K-NN* untuk mencari bentuk serupa yang terdapat pada data latih serta relasinya terhadap posisi slot area parkir. Misalnya bentuk lingkaran untuk slot parkir nomor B1, sedangkan bentuk persegi untuk slot parkir nomor B2.

Hasil pengujian untuk citra dengan *marker* dengan metode *K-NN* untuk bentuk bangun datar ditunjukkan seperti pada tabel III. Nilai rata-rata terbaik diperoleh untuk $K = 1$, yaitu dengan tingkat akurasi sebesar 91.94%.

TABEL III
CITRA MARKER BENTUK BANGUN DATAR

No.	Hasil		
	k=5	k=3	k=1
1	100.00%	100.00%	100.00%
2	100.00%	100.00%	100.00%
3	100.00%	100.00%	100.00%
4	75.00%	75.00%	75.00%
5	100.00%	100.00%	100.00%
6	100.00%	100.00%	100.00%
7	100.00%	100.00%	100.00%
8	100.00%	100.00%	100.00%
9	50.00%	100.00%	50.00%
10	100.00%	100.00%	100.00%
11	50.00%	50.00%	50.00%
12	50.00%	50.00%	50.00%
13	33.33%	33.33%	33.33%
14	100.00%	100.00%	100.00%
15	100.00%	50.00%	100.00%
16	100.00%	100.00%	100.00%
17	100.00%	100.00%	100.00%
18	50.00%	50.00%	100.00%
19	75.00%	100.00%	100.00%
20	100.00%	100.00%	100.00%
21	100.00%	100.00%	100.00%
22	100.00%	100.00%	100.00%
23	100.00%	100.00%	100.00%
24	100.00%	100.00%	100.00%
25	100.00%	100.00%	100.00%
26	100.00%	100.00%	100.00%
27	100.00%	100.00%	100.00%
28	100.00%	100.00%	100.00%
29	100.00%	66.67%	100.00%
30	100.00%	100.00%	100.00%
Rata-rata	89.44%	89.17%	91.94%

Hasil pengujian untuk citra dengan *marker* dengan metode *K-NN* untuk bentuk huruf alfabet ditunjukkan seperti pada tabel IV. Nilai rata-rata terbaik diperoleh untuk $K = 1$, yaitu dengan tingkat akurasi sebesar 88.61%.

TABEL IV
CITRA MARKER BENTUK HURUF ALFABET

No.	Hasil		
	k=5	k=3	k=1
1	100.00%	100.00%	100.00%
2	100.00%	100.00%	100.00%
3	75.00%	75.00%	100.00%
4	100.00%	100.00%	100.00%
5	100.00%	100.00%	100.00%
6	100.00%	100.00%	100.00%
7	100.00%	100.00%	100.00%
8	100.00%	100.00%	50.00%
9	50.00%	50.00%	50.00%
10	66.67%	66.67%	66.67%
11	50.00%	100.00%	100.00%
12	100.00%	100.00%	100.00%
13	66.67%	100.00%	100.00%
14	66.67%	66.67%	66.67%
15	100.00%	100.00%	100.00%
16	100.00%	100.00%	100.00%
17	50.00%	100.00%	100.00%
18	100.00%	100.00%	100.00%
19	66.67%	66.67%	66.67%
20	100.00%	100.00%	100.00%
21	66.67%	66.67%	66.67%
22	50.00%	50.00%	50.00%
23	75.00%	75.00%	75.00%
24	50.00%	50.00%	100.00%
25	100.00%	100.00%	100.00%
26	66.67%	66.67%	66.67%
27	100.00%	100.00%	100.00%
28	100.00%	100.00%	100.00%
29	100.00%	100.00%	100.00%

No.	Hasil		
	k=5	k=3	k=1
30	100.00%	100.00%	100.00%
Rata-rata	83.33%	87.78%	88.61%

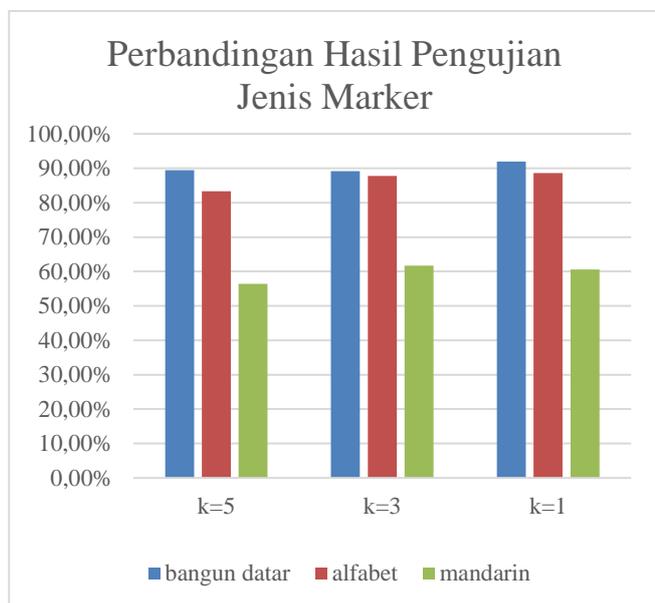
Hasil pengujian untuk citra dengan *marker* dengan metode *K-NN* untuk bentuk huruf mandarin ditunjukkan seperti pada tabel V. Nilai rata-rata terbaik diperoleh untuk $K = 3$, yaitu dengan tingkat akurasi sebesar 61.67%.

TABEL V
CITRA MARKER BENTUK HURUF MANDARIN

No.	Hasil		
	k=5	k=3	k=1
1	66.67%	66.67%	66.67%
2	100.00%	100.00%	100.00%
3	50.00%	50.00%	50.00%
4	33.33%	66.67%	66.67%
5	100.00%	100.00%	100.00%
6	50.00%	50.00%	50.00%
7	50.00%	50.00%	50.00%
8	33.33%	33.33%	66.67%
9	50.00%	50.00%	50.00%
10	25.00%	25.00%	25.00%
11	100.00%	100.00%	100.00%
12	50.00%	50.00%	50.00%
13	50.00%	50.00%	50.00%
14	50.00%	50.00%	50.00%
15	50.00%	100.00%	50.00%
16	50.00%	100.00%	50.00%
17	0.00%	0.00%	50.00%
18	100.00%	100.00%	100.00%
19	66.67%	66.67%	33.33%
20	50.00%	100.00%	50.00%
21	33.33%	33.33%	33.33%
22	100.00%	33.33%	66.67%
23	66.67%	66.67%	66.67%
24	75.00%	50.00%	50.00%

No.	Hasil		
	k=5	k=3	k=1
25	75.00%	75.00%	75.00%
26	33.33%	33.33%	33.33%
27	50.00%	50.00%	50.00%
28	33.33%	66.67%	66.67%
29	66.67%	66.67%	66.67%
30	33.33%	66.67%	100.00%
Rata-rata	56.39%	61.67%	60.56%

Berdasarkan hasil pengujian dari seluruh tipe bentuk *marker* yang ada, didapatkan grafik hasil perbandingan seperti ditunjukkan pada gambar 15. Pada grafik tersebut, dapat terlihat bahwa tingkat akurasi tertinggi dihasilkan oleh *marker* dengan tipe bangun datar, kemudian diikuti oleh *marker* dengan tipe alfabet. Untuk nilai akurasi paling rendah dihasilkan oleh *marker* dengan tipe huruf mandarin. *Marker* dengan tipe huruf mandarin sulit dikenali oleh sistem, karena huruf tersebut disusun oleh garis-garis dengan ukuran yang lebih tipis jika dibandingkan dengan *marker* tipe lain. Hal ini menyebabkan tidak semua garis dapat ditangkap dengan baik oleh kamera, sehingga sistem akan sulit mengenali bentuk huruf tersebut.



Gambar 15. Grafik perbandingan seluruh tipe bentuk *marker*

Kendala yang dialami selama penelitian adalah kondisi pencahayaan yang tidak menentu. Hal ini menyebabkan pantulan pada *marker*, sehingga warna *marker* yang tertangkap pada citra berubah menjadi warna lain seperti

ditunjukkan pada gambar 16 (*marker* di sebelah kanan mobil putih).



Gambar 16. Citra dengan pencahayaan terlalu terang

Pada gambar 16, terdapat dua buah *marker* pada yang berhasil tertangkap pada citra, yaitu di sisi kiri dan sisi kanan. Namun *marker* di sisi kanan terkena pantulan cahaya matahari, sehingga warna *marker* cenderung berubah menjadi warna putih. Hal ini mengakibatkan *marker* di sisi kanan gagal dideteksi menggunakan nilai *Hue* yang telah ditetapkan untuk warna hijau seperti ditunjukkan pada gambar 17. Sehingga hanya satu *marker* saja yang berhasil dideteksi oleh sistem, sehingga sistem menyimpulkan slot parkir kosong yang tersedia hanya satu, sementara pada kondisi sebenarnya terdapat dua slot parkir yang kosong.



Gambar 17. Citra dengan *marker* yang gagal terdeteksi

VI. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan bahwa sistem dapat mendeteksi area parkir dengan baik. Hal ini ditunjukkan dengan nilai rata-rata akurasi tertinggi sebesar 91.94% untuk nilai K pada K -NN sebesar 1 dan bentuk *marker* bertipe bangun datar. Untuk *marker* tipe lain, masing-masing memiliki rata-rata akurasi 88.61% ($K = 1$) pada *marker* bertipe alfabet dan 61.67% ($K = 3$) pada *marker* bertipe huruf mandarin. Kendala yang dialami dalam penelitian ini adalah kondisi cahaya yang tidak menentu, sehingga pada beberapa situasi mengakibatkan sistem gagal dalam mendeteksi *marker* yang terpasang pada slot parkir karena pantulan cahaya matahari yang terlalu terang. Selain itu, *marker* dengan bentuk huruf mandarin cukup sulit dikenali karena bentuk tersebut disusun oleh garis-garis yang tipis, sehingga akan sulit tertangkap oleh kamera, terutama jika *marker* diambil dengan sudut yang cukup tajam.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada saudara Aditya Yuga Pradhana sebagai asisten peneliti yang telah banyak membantu penulis selama proses penelitian berlangsung.

DAFTAR PUSTAKA

- [1] BPS, "Badan Pusat Statistik," Badan Pusat Statistik, 2017. [Online]. Available: <https://www.bps.go.id/linkTableDinamis/view/id/1133>. [Accessed 2019].
- [2] T. Lin, H. Rivano and F. L. Mouël, "A Survey of Smart Parking Solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229 - 3253, 2017.
- [3] A. O. Kotb, Y.-c. Shen and Y. Huang, "Smart Parking Guidance, Monitoring and Reservations: A Review," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 6 - 16, 2017.
- [4] G. Yan, W. Yang, D. B. Rawat and S. Olariu, "SmartParking: A Secure and Intelligent Parking System," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 18 - 30, 2011.
- [5] Y. Geng and C. G.Cassandras, "A new "Smart Parking" System Infrastructure and Implementation," in *Procedia - Social and Behavioral Sciences,15th Meeting of the EURO Working Group on Transportation*, Paris, 2012.
- [6] V. Paidi, H. Fleyeh, J. Håkansson and R. G. Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 735 - 741, 2018.
- [7] R. M. Patil, N. R. Vinay and P. D, "Application-based Smart Parking System using CAN Bus," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 2, pp. 759-764, 2018.
- [8] B. S. R, "Automatic Smart Parking System using Internet of Things (IOT)," *International Journal of Scientific and Research Publications*, vol. 5, no. 12, pp. 629-632, 2015.
- [9] H. Chaudhary, P. Bansal and B. Valarmathi, "Advanced CAR parking system using Arduino," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2017.
- [10] P. Sheelarani, S. P. Anand, S. Shamili and K. Sruthi, "Effective car parking reservation system based on internet of things technologies," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, Coimbatore, India, 2016.
- [11] S. S. Adhatarao, O. Alfandi, A. Bochem and D. Hogrefe, "Smart parking system for vehicles," in *2014 IEEE Vehicular Networking Conference (VNC)*, Paderborn, Germany, 2014.
- [12] R. Yellasiri, T. Sridevi, B. Poornima and B. Kalyani, "Segmentation And Object Recognition Using Edge Detection Techniques," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 6, pp. 153-161, 2010.
- [13] W. A. Mustafa and M. M. A. Kader, "Binarization of Document Image Using Optimum Threshold Modification," in *1st International Conference on Green and Sustainable Computing (ICoGeS) 2017*, Sarawak, Malaysia, 2017.
- [14] G. Saravanan, G. Yamuna and S. Nandhini, "Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, India, 2016.
- [15] K. Schwenk and F. Huber, "Connected Component Labeling Algorithm for very complex and high resolution images on an FPGA platform," in *High-Performance Computing in Remote Sensing V*, Toulouse, France, 2015.
- [16] A. Sit and D. Kihara, "Comparison of Image Patches Using Local Moment Invariants," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2369 - 2379, 2014.
- [17] S. Zhang, X. Li, M. Zong, X. Zhu and R. Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774 - 1785, 2017.