

## IMPLEMENTASI HISTOGRAM OF ORIENTED GRADIENT(HOG) PADA EMBEDDED SYSTEM UNTUK IDENTIFIKASI SLOT PARKIR SEBAGAI PENDUKUNG SMART PARKING SYSTEM

### *Implementation Histogram Of Oriented Gradient (HOG) in Embedded System For Identification Of Parking In Support Slot Smart Parking System*

Rahadian Nugraha<sup>1</sup>, Agung Nugroho Jati<sup>2</sup>, Umar Ali Ahmad<sup>3</sup>

<sup>1,3</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>rahadiannugraha@students.telkomuniversity.ac.id, <sup>2</sup>agungnj@telkomuniversity.ac.id, <sup>3</sup>uaa@ypt.or.id

---

#### Abstrak

Pada penelitian tugas akhir ini membuat sebuah sistem parkir yang dapat memudahkan para penggunanya. Untuk itu diperlukan suatu sistem yang dapat menjadi solusi tersebut. Sistem dapat mendeteksi ketersediaan slot parkir. Sistem membutuhkan masukkan citra dari kamera. Kemudian citra tersebut dimuat setelah itu sistem akan melakukan perbandingan feature. Setelah data diolah, sistem akan menghasilkan *logic* yang digunakan untuk proses selanjutnya dalam sistem secara keseluruhan. Sistem merupakan penerapan dari *Computer Vision*. Sistem akan mengambil citra menggunakan kamera yang sudah terpasang pada area parkir. Citra diproses menjadi *image feature hog* dengan menggunakan metode *Histogram Of Oriented Gradient (HOG)* untuk mendeteksi slot parkir. Dengan citra yang dirubah menjadi *feature hog*, maka citra tersebut memiliki nilai vector. Nilai inilah yang dibandingkan dengan citra masukkan lainnya. Dari hasil pengujian, posisi optimal dari kamera untuk mendapatkan hasil yang optimal adalah dengan jarak 8 meter dari slot parkir dengan akurasi *distance feature hog* sebesar 93.33 % serta dengan threshold sebesar 50 dengan akurasi sebesar 95% agar dapat mendeteksi objek.

**Kata Kunci :** *Smart Parking System, HOG feature, slot parkir*

---

#### Abstract

At this feature research made a parking system which could make parking their vehicles easier. Related to the mentioned problem before, a system which can be a solution of this problem is needed a system which can detect the availability of a parking lot. The system needs an image processing from a camera. After that, the image processing is loaded and the system will do feature comparison. Right after the data are processed, the system will produce logic which is used for the next project in the whole system. The system is the application of *Computer Vision*. The system will take the image processing by using an attached camera in the parking lot. An image processing will be processed to be a feature hog by using *Histogram of Oriented Gradient (HOG)* to detect the parking slot. With the image processing which is already changed into a feature hog, that changed image processing has vector value. This value will be compared with other image processing. The result of the test showed that the optimal position from a camera to get an optimal result was in height of  $\pm 6$  meter pole with distance of  $\pm 8$  meter from parking slot followed by 93.33 % accuracy distance feature hog and by threshold 50 so that it could detect objects with percentage of 95%.

**Key Words:** *Smart Parking System, HOG feature, parking slot*

---

## 1. Pendahuluan

Area parkir sangat dibutuhkan di tempat umum seperti pusat perbelanjaan (mall), tempat rekreasi, perkantoran, bandara, perhotelan dan lain-lain. Apalagi dengan berkembang pesatnya produksi mobil di dalam negeri mencapai 900 ribu unit. Kapasitas akan bertambah menjadi 1,43 juta unit pada tahun 2014[6]. Jumlah mobil yang semakin bertambah maka mengakibatkan sulitnya mencari ruang parkir. Dalam mencari ruang parkir sering terjadi kekeliruan dari jasa parkir yang tersedia yang menganggap area parkir kosong. Kurangnya informasi tentang area mana yang kosong dan terisi yang mengakibatkan pengguna jasa parkir terjebak di lokasi parkir dan harus memutar kembali kendaraan tersebut dan mencari lokasi parkir lainnya.

Pada tugas akhir ini dibuat sebuah alat yang bernama *Smart Parking System*. Kegunaan alat ini secara umum adalah sebagai mempermudah pengemudi untuk memilih slot parkir mana yang kosong. Alat ini akan memvisualisasikan pada monitor slot parkir mana yang kosong dan SPS ini di lengkapi sistem pembayaran tiket secara terkomputerisasi. Alat ini dirancang menggunakan kamera dengan metode *Histogram of Oriented Gradient (HOG)*, dimana metode ini adalah sebuah metoda untuk mengetahui objek yang bergerak, metode ini akan mengidentifikasi slot parkir yang terisi dan kosong dengan *Histogram of Oriented Gradient (HOG) feature*. Kamera dipasang pada suatu titik untuk *mengcover* beberapa tempat parkir yang terhubung dengan *raspi* secara *realtime* dan akan di visualisasikan dan ditampilkan pada sebuah monitor. Dalam monitor ini akan ditampilkan beberapa informasi terkait slot parkir yang kosong dan terisi. Pengelola juga dapat melihat situasi dengan adanya *dashboard* ini akan kepadatan dari area parkir secara berkala. Sistem ini juga di lengkapi dengan *Failover* agar menjamin ketersediaan data yang ada untuk klien.

Dengan adanya aplikasi ini diharapkan dapat mempermudah pengemudi dalam mencari tempat parkir, Serta meminimalisir kesalahan yang di akibatkan oleh manusia agar tidak merugikan pengemudi atau pengelola jasa parkir.

## 2. Tinjauan Pustaka

### 2.1 Pengolahan Citra (Image Processing)

Pengolahan citra atau Image Processing adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*)[5]. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka image processing tidak dapat dilepaskan dengan bidang computer vision[5].

### 2.2 Citra Digital

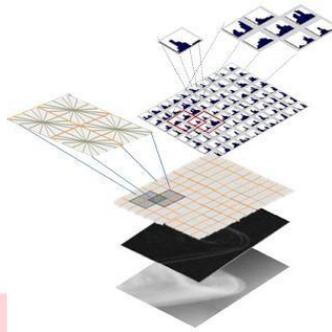
Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetic[2]. Citra digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Jadi informasi yang terkandung bersifat diskret. Citra digital tidak selalu merupakan hasil langsung data rekaman suatu sistem. Kadang-kadang hasil rekaman data bersifat kontinu seperti gambar pada monitor televisi, foto sinar-X, dan lain sebagainya. Dengan demikian untuk mendapatkan suatu citra digital diperlukan suatu proses konversi, sehingga citra tersebut selanjutnya dapat diproses dengan komputer[2].

### 2.3 Histogram Of Oriented Gradient (HOG)

*Histogram Of Oriented Gradient (HOG)* adalah sebuah metode yang digunakan dalam *image processing* untuk bertujuan deteksi objek. Teknik ini menghitung nilai *gradient* dalam daerah tertentu pada suatu *image*[3]. Tiap *image* mempunyai karakteristik yang ditunjukkan oleh distribusi *gradient*. Karakteristik ini diperoleh dengan membagi *image* kedalam daerah kecil yang disebut *cell*. Tiap *cell* disusun sebuah histogram dari sebuah *gradient*. Kombinasi dari histogram ini dijadikan sebagai descriptor yang mewakili sebuah obyek.

Pada metode ini, *feature HOG* dapat diperoleh dari membagi gambar kedalam *cell* berukuran  $n \times n$ , lalu dikelompokkan yang saling beririsan satu sama lain. Dari tiap *cell* masing-masing blok, dihitung magnitude dan orientasi *gradientnya*[4].

Berikut adalah bentuk *cell* dan blok dalam HOG :



Gambar 1 Cell dan Blok HOG[3]

Dari tiap *cell* dan masing-masing blok, dihitung magnitude dan orientasi gradient-nya.

$$G = G_x + G_y$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

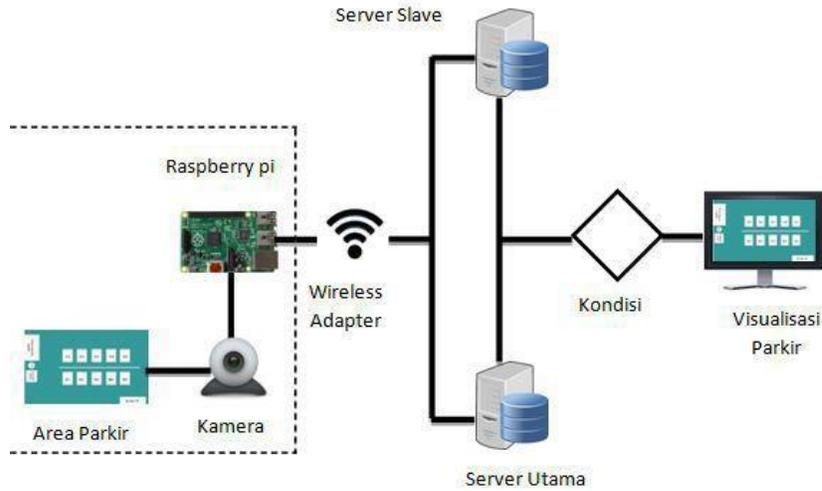
Nilai orientasi ini dihitung dengan menghitung konvolusi dengan matrix  $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$  dan  $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$  terlebih dahulu. Kemudian nilai orientasi tiap pikselnya dikuantisasi kedalam 9 kanal, yaitu  $10^\circ, 30^\circ, 50^\circ, 70^\circ, 90^\circ, 130^\circ, 150^\circ$  dan  $170^\circ$  menggunakan histogram[3]. Kontribusi piksel terhadap tiap kanal bergantung pada nilai *gradient* magnitude-nya. Nilai dari seluruh kanal dari tiap *cell* dimasukkan kedalam *vector*. *Vector* inilah yang akan menggambarkan *feature HOG* dari suatu citra.



Gambar 2 Visualisasi Feature HOG[7]

#### 2.4 Gambaran Umum Sistem

Pada perancangan sistem menjelaskan alur dari proses yang dikerjakan pada tugas akhir ini.



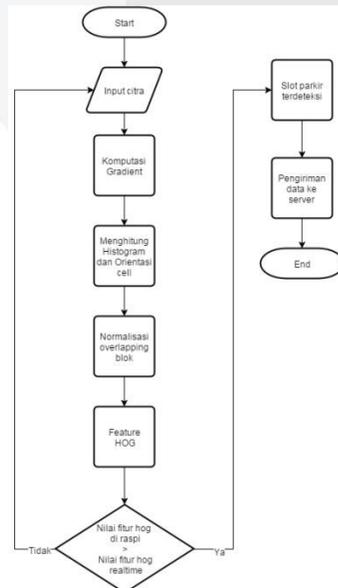
Gambar 3 Sistem Keseluruhan

Sistem ini dibuat untuk area parkir di mana dengan adanya *smart parking system* maka akan mempermudah pengunjung untuk mengetahui keadaan area parkir apakah penuh atau tidak. Untuk mengetahui parkir kosong atau tidak menggunakan kamera yang terintegrasi dengan *raspberry pi* untuk pendeteksian slot parkir. Metode yang digunakan untuk deteksi slot adalah *histogram of oriented gradient* yang mana (HOG) berfungsi sebagai pengenalan obyek. Keluaran *histogram of oriented gradient* berupa nilai *true or false*. Untuk sistem pengiriman data menggunakan *wireless* sebagai media pengiriman data antara *raspberry pi* ke dua server.

*Server* utama akan bekerja apabila *server* utama mengalami gangguan maka *server slave* akan mengambil alih kinerja *server* utama. Pengecekan apakah *server* mengalami gangguan atau tidak menggunakan *TCP\_CHECK* dan *HTTP\_GET*. Apabila *server* utama kembali *up* maka *server* utama ngambil kembali sistem dan menyamakan data yang hilang ketika *server* utama *down* dengan menggunakan replikasi. Aplikasi *smart parking system* berguna untuk pengunjung parkir melihat apakah slot parkir yang tersedia penuh atau tidak lewat aplikasi yang disediakan. Fitur *chargingparking* berfungsi untuk pendataan kapan pengunjung parkir datang dan keluar area parkir.

**2.5 Diagram Alir System**

Berikut ini merupakan diagram alir gambaran umum sebagai berikut :



Gambar 4 Flowchart system keseluruhan

Proses deteksi slot parkir yang diimplementasikan pada *raspberry pi* menggunakan metode *HOG* untuk membandingkan *image true* dan *image false*. Tahap pertama yaitu adalah menghitung nilai *gradient* dalam daerah tertentu pada suatu citra. Tiap citra mempunyai karakteristik yang ditunjukkan oleh *gradient*. Karakteristik ini diperoleh dengan membagi citra kedalam daerah kecil yang disebut dengan *cell*. Tiap *cell* disusun dari sebuah histogram dari sebuah *gradient*. Kombinasi dari histogram ini dijadikan sebagai descriptor yang mewakili sebuah objek. Karakteristik ini visualisasikan sebagai *feature HOG*. Kemudian nilai dari *image feature hog* yang sudah diolah pada *raspi* akan dibandingkan dengan *image feature hog* yang didapatkan secara *realtime* yang didapat dari area parkir. Keluaran dari pengolahan pada *raspi* yaitu berupa nilai *true or false*, jika nilai *image feature hog* dari *raspi* bernilai lebih besar dari nilai *image feature hog realtime* maka sistem akan menghasilkan keluaran *true*. Kemudian langkah terakhir dari sistem ini adalah sistem akan mengirimkan nilai *true* tersebut ke server utama dan akan ditampilkan pada layar monitor server.

**3. Pembahasan**

Sistem deteksi slot parkir ini menggunakan *raspberry pi*. Pengujian dilakukan untuk mendapatkan akurasi dari metode *hog*. Pengujian terhadap system seperti tingkat kemiripan feature dan tingkat kecepatan system dalam mendeteksi ketersediaan slot parkir.

**3.1 Pengujian Distance Feature hog**

Pengujian *distance feature hog* bertujuan untuk menentukan tingkat kesamaan atau ketidaksamaan dua *feature hog*. Pengujian dilakukan sebanyak 15 kali, dilakukan sebanyak 15 kali karena *raspi* terlalu berat dalam melakukan pengolah *image feature hog*. Untuk mendapatkan hasil yang maksimal, posisi kamera sangatlah menentukan. Untuk pengujian ini digunakan sebuah tiang yang diletakkan pada area parkir dengan tinggi tiang setinggi ± 6 meter dan jarak dari area parkir sejauh ± 8 meter.

Tabel 3.1 Pengujian Distance Feature hog

No	Distance	Jumlah Pengujian	Jumlah yang benar	Slot terdeteksi	Akurasi (%)
1.	230.55	15	14	7	93.33
2.	350.55	15	12	6	80
3.	450.55	15	11	5	73.33
4.	550.65	15	9	2	60
5.	670.75	15	5	1	33.33
6.	700.75	15	0	0	0

Berdasarkan dari hasil pengujian yang didapatkan, bahwa penentuan distance pada pendeteksian sangat berpengaruh untuk deteksi slot parkir. Dapat dilihat dari hasil yang baik yaitu pada distance 230.55 dengan hasil akurasi sebesar 93.33% dan masih dapat mendeteksi slot parkir pada distance 350.55. Sedangkan pada distance 670 dan 700 hanya dapat mendeteksi slot parkir sebanyak 1 saja. Dapat diambil kesimpulan bahwa distance yang ditentukan sangat berpengaruh pada hasil deteksi slot parkir. Yang perlu diperhatikan mencari sudut yang tepat untuk tata letak kamera dengan posisi slot parkir yang ingin dideteksi agar bisa didapatkan tingkat akurasi yang baik.

**3.2 Pengujian terhadap Threshold**

*Threshold* digunakan untuk menghilangkan *noise* pada *frame* yang mendeteksi dari sebuah objek. Pengujian *threshold* dilakukan sebanyak 20 kali karena *raspi* mengenali objek dengan waktu 5 detik setiap kali deteksi, maka dari itu pengujian *threshold* dilakukan sebanyak 20 kali disetiap pengujian batas *threshold* dalam mengenali objek untuk mendapatkan hasil yang maksimal dimana objek tersebut yang berjalan secara *realtime*. Tujuan digunakannya *threshold* adalah untuk mengetahui batas *threshold* yang cocok agar objek terdeteksi.

Tabel 3.2 Pengujian Terhadap *threshold*

No	Threshold	Jumlah Pengujian	Jumlah yang benar	Slot terdeteksi	Akurasi (%)
1.	50	20	19	7	95
2.	100	20	17	6	85
3.	200	20	16	3	80
4.	300	20	8	2	40
5.	500	20	5	1	25

Berdasarkan dari hasil pengujian yang telah dilakukan, sistem dapat mengenali objek dengan *threshold* 50 dengan persentase 95%. Karena jika *frame* terlalu banyak *noise* sulit untuk mendeteksi objek. Dapat diambil kesimpulan bahwa nilai *threshold* pada suatu pendeteksi objek sangat penting digunakan, karena nilai *threshold* sangat mempengaruhi untuk pendeteksian suatu objek.

#### 4. Kesimpulan dan Saran

##### 4.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada system, dapat diambil beberapa kesimpulan sebagai berikut :

1. Dari penelitian yang sudah dilakukan, dengan metode *Histogram Of Oriented Gradient(HOG)* dapat mengidentifikasi slot parkir. Distance optimal yang digunakan untuk deteksi seluruh slot adalah sebesar 230.33 untuk mendapatkan hasil yang baik.
2. Pada pengujian *threshold* untuk mampu mendeteksi slot parkir optimal pada 50 dan masih dapat mendeteksi sebesar 100. Sedangkan dengan *threshold* sebesar 300 sytem sulit untuk mendeteksi slot parkir karena terlalu banyak *noise*.

##### 4.2 Saran

Setelah aplikasi ini berhasil dibangun, masih diperlukan pengembangan kearah yang lebih baik agar lebih bermanfaat untuk penelitian selanjutnya.

1. Perlu dilakukan analisis selanjutnya untuk mengidentifikasi yang kosong dan yang terisi menggunakan metode deteksi objek dalam metode yang lain, agar keseluruhan penelitian dalam identifikasi slot parkir ini menjadi lebih baik lagi.
2. Dapat ditambahkan langkah antisipasi untuk mendeteksi slot parkir agar lebih banyak lagi yaitu dengan memilih tempat parkir yang sesuai dengan pemasangan kamera yang cukup tinggi agar mencakup banyak slot parkir pada area parkir.

**DAFTAR PUSTAKA**

- [1] Fernando Nuno Rodrigues Fernandes da Costa, "Detection and Classification of moving objects with Raspberry-Pi for the Future Cities Project," *Faculdade de Engenharia da Universidade do Porto*, Feb. 2014.
- [2] Darma Citra, *Pengolahan Citra Digital*, Westriningsih, Ed. Yogyakarta, Indonesia: C.V ANDI OFFSET, 2010.
- [3] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," *INRIA Rh<sup>one</sup>-Alps, 655 avenue de l'Europe, Montbonnot 38334, France*, 2005.
- [4] I Ketut Eddy Purnama dan Muhtadin Cahyo Permata, "Deteksi Mobil Menggunakan Histogram of Oriented Gradient," *Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)*, July 2012.
- [5] <http://elektronika-dasar.web.id/>. [Online]. <http://elektronika-dasar.web.id/definisi-dan-pengolahan-citra-digital/>
- [6] media industri, "Pertumbuhan industri manufaktur," p. 32, 2013.
- [7] (2015, Mei) <http://stats.stackexchange.com/>. [Online]. <http://stats.stackexchange.com/questions/149421/obtaining-a-hog-feature-vector-for-implementation-in-svm-in-python>