

# PERBANDINGAN JARAK POTRET DAN RESOLUSI KAMERA PADA TINGKAT AKURASI PENGENALAN ANGKA KWH METERMENGGUNAKAN SVM

Dini Amputri<sup>1)</sup>, Siti Nadra<sup>2)</sup>, Gasim<sup>\*3)</sup>, M. Ezar Al Rivan<sup>4)</sup>

<sup>1), 2), 3), 4)</sup> Program Studi Teknik Informatika STMIK Global Informatika MDP  
Jl. Rajawali No.14 Palembang Kode pos 30113

Email : [diniamputri@mhs.mdp.ac.id](mailto:diniamputri@mhs.mdp.ac.id)<sup>1)</sup>, [nadra.sn@mhs.mdp.ac.id](mailto:nadra.sn@mhs.mdp.ac.id)<sup>2)</sup>, [gasim@mdp.ac.id](mailto:gasim@mdp.ac.id)<sup>3)</sup>, [meedzhar@mdp.ac.id](mailto:meedzhar@mdp.ac.id)<sup>4)</sup>

## ABSTRACT

*Electricity meter is a tool used to measure the electricity power consumption. Electricity meter has the numeral part known as electricity meter number which load the electricity power consumption. Usually, the recording is done once a month, by recording the number on the electric meter into the book, and then the recording officer take pictures the power meter. Electricity meter number can be analyzed through an image using the knowledge of pattern recognition in image processing. In analyzing picture of electricity meter number, we used Histogram Of Oriented Gradients (HOG) as the feature extraction and Support Vector Machine (SVM) as the classification method. The result using 100 train-set and 30 test-set for each combination of category shows that the best resolution is 10 MP and 14 MP and the picture capturing distance is at 30 cm by and 10 cm by 73,33% accuracy for each image and 86,67% for each number and confusion matrix shows that presentation of all number is 75,48%.*

**Key words:** Number Recognition, Histogram Of Oriented Gradients (HOG), Support Vector Machine (SVM)

## 1. Pendahuluan

Stand kWh meter adalah alat yang digunakan untuk mengukur besarnya pemakaian daya listrik. Pada bagian kWh meter terdapat angka-angka yang dikenal dengan angka *stand kWh* meter yang menampilkan besarnya pemakaian daya listrik, sehingga perlu dicatat. Pencatatan atau pembacaan kWh meter dilakukan oleh petugas baca meter. Petugas mendatangi rumah pelanggan sesuai dengan Rute Baca Meter (RBM) yang telah disediakan, kemudian petugas membaca angka *stand kWh* meter dan mencatat angka tersebut pada Dokumen Pembacaan Meter (DPM) sesuai identitas pelanggan. Selanjutnya petugas mengambil citra kWh meter sebagai bukti dengan menggunakan kamera.

Metode pencatatan tersebut sebenarnya dapat dilakukan secara otomatis melalui pengenalan angka melalui citra menggunakan ilmu pengenalan pola. Akan tetapi dalam melakukan pengenalan angka *stand kWh* meter terdapat beberapa hal yang mempengaruhi tingkat akurasi dari sisi proses pengambilan citra, yaitu jarak pengambilan citra, resolusi kamera, serta pencahayaan saat pengambilan citra.

Penelitian-penelitian sebelumnya telah menerapkan berbagai macam ekstraksi fitur dan klasifikasi dalam pengenalan angka *stand kWh* meter, yaitu pengenalan angka *stand kWh* meter menggunakan sistem otomatisasi dan *Support Vector Machine* (SVM), penelitian ini memberikan tingkat akurasi 78,10% untuk pengenalan satu set angka *stand kWh* meter menggunakan berbagai jenis kualitas citra [1].

Penerapan *Optical Character Recognition* (OCR) untuk pembacaan meteran listrik, penelitian ini menunjukkan bahwa sistem tidak bisa mengenali karakter pada citra yang belum dilatih, faktor yang

mempengaruhi adalah *noise* pada citra, pencahayaan, sudut pengambilan citra [2].

Penelitian menggunakan ekstraksi ciri *Histogram Of Oriented Gradients* (HOG) untuk pengenalan plat kendaraan dengan metode klasifikasi SVM dengan akurasi yang didapat untuk pengenalan angka sebesar 99% dan 95% pada angka dan huruf [3].

Penelitian membahas tentang klasifikasi digit angka, menggunakan ekstraksi ciri HOG dengan metode klasifikasi *Multiclass SVM* dan metode klasifikasi *Artificial Neural Network* (ANN). Hasil penelitian menunjukkan tingkat akurasi kebenaran sebesar 84% untuk klasifikasi menggunakan ANN, sedangkan untuk klasifikasi menggunakan SVM menunjukkan tingkat akurasi sebesar 97,5% [4].

Penelitian mengenai pengenalan tulisan tangan menggunakan HOG dan klasifikasi ANN dengan metode BP (*Back Propagation*) Hasil penelitian menunjukkan tingkat akurasi sebesar 98,2% [5].

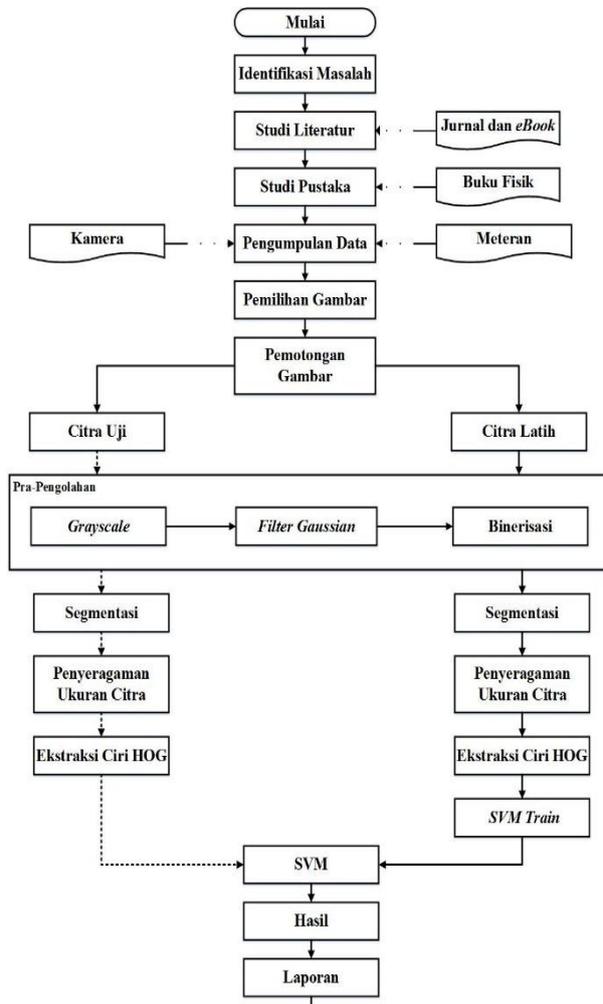
Penelitian mengenai pengenalan tulisan tangan pada huruf Marathi menggunakan ciri Rectangle HOG (R-HOG). Penelitian ini menggunakan 40 tulisan tangan huruf Marathi yang ditulis oleh masing-masing orang (10x40) yang diklasifikasi menggunakan *Feed Forward Artificial Neural Network* (FFANN) dan SVM. Hasil penelitian menunjukkan tingkat akurasi pada klasifikasi FFAN sebesar 97,15% dan klasifikasi SVM sebesar 95,64% [6].

Penelitian mengenai pengenalan angka secara otomatis menggunakan ekstraksi fitur HOG dan pengenalan tulisan menggunakan *multiclass SVM*. Penelitian ini menggunakan database tulisan tangan MNIST dari New York University dengan hasil akurasi sebesar 97,25 [7]. Berdasarkan penelitian-penelitian tersebut maka akan dilakukan penelitian menggunakan

ekstraksi ciri HOG dan metode klasifikasi SVM untuk pengenalan angka *stand kWh* meter.

A. Metodologi Penelitian

Alur penelitian disajikan dalam bentuk framework pada Gambar 1.



Gambar 1. Framework

1) Identifikasi Masalah

Identifikasi masalah yang dilakukan dengan menelusuri jurnal-jurnal sebagai referensi atas penelitian-penelitian sejenis yang sudah pernah dilakukan. Studi literatur dan studi pustaka guna menunjang keilmuan terkait pengenalan angka pada kWh meter, HOG dan SVM.

2) Pengumpulan Data

Pengumpulan data dilakukan pada siang hari menggunakan kamera digital Casio EX-ZS5 dengan resolusi 6 MP, 10 MP, dan 14 MP dan jarak antara kWh meter dan kamera adalah lebih kurang 10cm, 20cm, dan 30cm.

3) Pemilihan Gambar

Pemilahan citra dilakukan karena pada proses pengambilan citra (pemotretan), tidak semua citra dalam

keadaan baik, artinya terdapat citra yang tidak fokus (kabur). Selanjutnya, dilakukan perubahan nama file citra untuk memisahkan citra per-jarak dan resolusi.

4) Pemotongan Gambar

Pemotongan citra secara manual untuk mengambil daerah bagian angka sehingga diperoleh hasil citra berupa daerah yang terdiri dari 5 digit angka *stand kWh* meter (Gambar 2).



Gambar 2. Citra Sudah dilakukan Pemotongan

5) Citra Latih dan Citra Uji

Setelah didapatkan citra angka *stand kWh* meter melalui proses pemotongan maka langkah selanjutnya adalah mengelompokkan citra menjadi dua, yaitu citra latih (*training set*) dan citra uji (*test set*). Citra latih dan citra uji memiliki 9 jenis berdasarkan resolusi kamera dan jarak pengambilan gambar, yaitu:

1. Resolusi 6 MP dengan jarak 10 cm.
2. Resolusi 10 MP dengan jarak 10 cm.
3. Resolusi 14 MP dengan jarak 10 cm.
4. Resolusi 6 MP dengan jarak 20 cm.
5. Resolusi 10 MP dengan jarak 20 cm.
6. Resolusi 14 MP dengan jarak 20 cm.
7. Resolusi 6 MP dengan jarak 30 cm.
8. Resolusi 10 MP dengan jarak 30 cm.
9. Resolusi 14 MP dengan jarak 30 cm.

Tiap-tiap jenis citra latih memiliki 10 kelas (kelas 0-9) dan masing-masing kelas memiliki 10 citra sehingga total citra latih adalah 900 buah, sedangkan citra uji terdiri dari 30 buah untuk setiap kategori sehingga total citra uji adalah 270 buah.

Citra latih akan melalui beberapa proses yaitu, segmentasi, penyeragaman ukuran gambar, ekstraksi fitur dengan HOG, SVM train, SVM serta pengenalan angka. Sama halnya dengan citra latih, citra uji akan melalui beberapa proses yang ada pada citra latih, namun citra uji tidak akan digunakan pada proses SVM *train*.

6) Grayscale

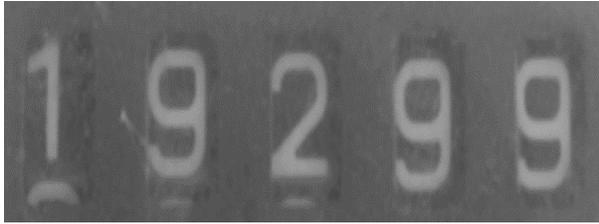
Tahapan pertama yang dilakukan pada citra latih dan citra uji yaitu konversi menjadi citra *grayscale*. Hasil konversi ke *grayscale* dapat dilihat pada Gambar 3.



Gambar 3. Hasil Konversi Citra Menjadi Grayscale

7) *Filter Gaussian*

Citra uji dan citra latihan yang telah diubah menjadi citra *grayscale* selanjutnya direduksi *noise* nya menggunakan fungsi *filter gaussian*. Hasil *filtering* citra dapat dilihat pada Gambar 4.



Gambar 4. Hasil *Filtering* Citra

8) *Biner*

Setelah penghalusan citra menggunakan *filter gaussian*, langkah selanjutnya yaitu mengubah citra menjadi citra biner (hitam putih). Hasil binerisasi citra dapat dilihat pada Gambar 5.



Gambar 5. Hasil *Konversi Menjadi Citra Biner*

9) *Segmentasi*

Setelah didapatkan citra hitam putih, tahap selanjutnya adalah tahap segmentasi. Citra uji dan citra latih yang sudah diolah pada tahap sebelumnya dipisahkan satu persatu sehingga menghasilkan beberapa citra dengan 1 digit angka didalamnya.

Pertama-tama dilakukan penghapusan *noise* pada citra biner (*bwareopen*), kemudian menghitung jumlah objek pada citra sebelumnya (*regionprops*). Dalam fungsi *regionprops*, *region* merepresentasikan objek pada citra biner di mana pendekatan letak atau posisi objek menggunakan properti *BoundingBox*. Properti *BoundingBox* akan memuat semua *region* dengan bentuk persegi panjang yang disimpan dalam *obj*.

Selanjutnya, untuk menentukan banyaknya *region* digunakan fungsi *length* pada *obj*. Tahap ketiga yaitu memisahkan objek-objek yang telah dideteksi menggunakan fungsi *imcrop*. Pertama-tama, temukan posisi *BoundingBox* pada *obj* lalu simpan ke dalam *posisi Obj*. Setelah posisi objek didapatkan, dilakukan pemotongan menggunakan fungsi *imcrop*.

Proses ini akan terus berulang sampai semua objek terpisah. Berikut contoh segmentasi citra dapat dilihat pada Gambar 6



Gambar 6. Hasil *Segmentasi*

10) *Penyeragaman Ukuran*

Setelah didapatkan hasil segmentasi citra, selanjutnya dilakukan penyeragaman ukuran. Penyeragaman ukuran ini dibutuhkan untuk ekstraksi ciri HOG karena citra hasil segmentasi memiliki ukuran yang berbeda-beda sedangkan ekstraksi ciri HOG harus menggunakan citra dengan ukuran yang sama. Adapun hasil penyeragaman ukuran citra dapat dilihat pada Gambar 7.



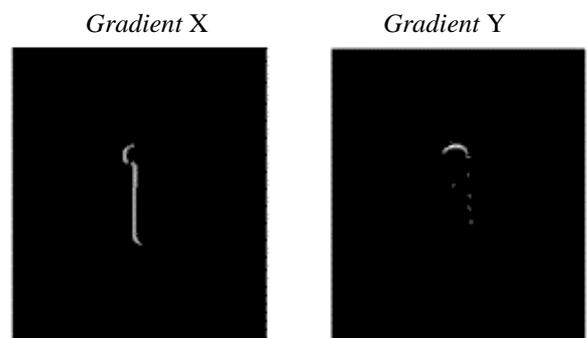
Gambar 7. Citra Hasil *Penyeragaman Ukuran*

11) *Ekstraksi Fitur*

Setelah melalui proses penyeragaman ukuran citra maka langkah selanjutnya adalah melakukan proses ekstraksi fitur menggunakan HOG pada citra latihan dan citra uji guna mendapatkan nilai vektor yang akan digunakan pada proses pengenalan.

a. *Komputasi Gradient*

Tahap pertama pada ekstraksi fitur HOG adalah menghitung *gradient* yaitu menghitung perubahan pada intensitas warna melalui operasi *central difference*. Nilai *gradient X* dan *gradient Y* merupakan matrik dengan ukuran citra yang berisi nilai 1 atau 0. Bernilai 1 jika terdapat perubahan intensitas dan bernilai 0 jika tidak terdapat perubahan. Hasil perubahan *gradient X* dan *Y* dapat dilihat pada Gambar 8.





Gambar 8. Citra Hasil Perubahan Gradient X dan Y

Selanjutnya, nilai matriks ini akan digunakan untuk menghitung *magnitude* (jarak *gradient*) dan *direction* atau (arah *gradient*) menggunakan fungsi *hogGradient*. Nilai *magnitude* dihitung menggunakan fungsi *hypot* pada MATLAB, yang diperoleh dari akar hasil kuadrat *gradient X* ditambah dengan kuadrat *gradient Y*. Hasil *magnitude* akan menunjukkan tepi dari objek gambar, seperti yang terlihat pada Gambar 9.



Gambar 9. Citra Hasil Magnitude

Sedangkan arah *gradient* dihitung menggunakan fungsi *atan2* pada MATLAB berdasarkan dengan perhitungan *inverse* tangen dari *gradientX* dan *gradientY*. Ilustrasi hasil perhitungan arah *gradient* dapat dilihat pada Gambar 10.



Gambar 10. Ilustrasi Hasil Perhitungan Arah Gradient

**b. Orientasi Binning**

Tahap selanjutnya yaitu menentukan nilai Bin pada setiap *cell*. Ukuran *cell* yang digunakan pada penelitian ini yaitu 8x8 sehingga jika citra memiliki ukuran 400x320 maka terdapat 50 *cell* horizontal dan 80 *cell* vertikal. Setiap *cell* dihitung nilainya menggunakan fungsi *hitungOrientasiBin*.

```
function [centerBin, indeksBin] = hitungOrientasiBin(x, jarakBin)
% index bin J
```

```
jarakBin = single(jarakBin);
jarakBin2 = 1./jarakBin;
bin = floor(x.*jarakBin2 - 0.5);
% Bin center centerBin Cj
centerBin = jarakBin * (bin + 0.5);
% add 2 to get to 1-based indexing
indeksBin = int32(bin + 2);
```

Fungsi *hitung OrientasiBin* menghitung *center bin* dan indeks bin pada setiap *cell* di mana masing-masing *cell* sendiri memiliki 9 bin. Indeks bin diperoleh melalui hasil bagi dari arah *gradient* dengan *jarakBin*, di mana *jarakBin=20* diperoleh dari 180 dibagi bin=9. Hasil bagi tersebut kemudian dikurangi dengan 0.5, berdasarkan persamaan 1.

$$j = \left[ \frac{\theta}{w} - \frac{1}{2} \right] \text{ mod } B \dots \dots \dots (1)$$

Sedangkan *centerBin* diperoleh dari hasil kali *jarakBin* dengan *indeksBin + 0.5*, berdasarkan persamaan  $C_j = w \left( j + \frac{1}{2} \right)$  sehingga akan berisi sudut 10, 30, 50, 70, 90, 110, 130, 150 dan 170.

Selanjutnya menentukan selisih nilai dari *arahGradient* dengan *centerBin*, yang didapatkan dari pengurangan 1 dengan hasil pengurangan arah *gradient* dengan *centerBin* yang dibagi dengan *jarakBin*. Nilai bin pada setiap *cell* diperoleh dari perkalian antara selisih nilai dengan *magnitude* berdasarkan Persamaan 2.

$$V_j = \mu \frac{C_{j+1} - \theta}{w} \dots \dots \dots (2)$$

Tahapan ini belum selesai karena nilai *gradient* masih memiliki nilai yang berbeda pada setiap *cell*-nya. Oleh karena itu perlu adanya pengelompokkan *cell* menjadi *block*.

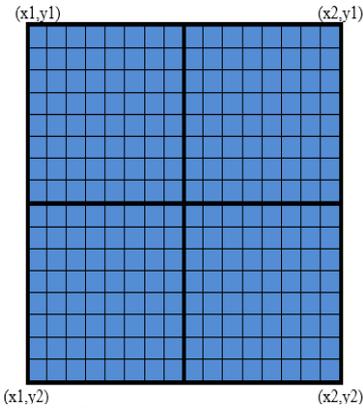
Penelitian ini menggunakan normalisasi *block R-HOG* sehingga *cell* yang berukuran 8x8 dikelompokkan menjadi *block* berukuran 2x2 akan memiliki ukuran 16x16. Langkah berikutnya yaitu menentukan batas dari *block* menggunakan fungsi *bobotHistogramSpasial*.

```
% Hitung bobot utk ruang histogram utk semua cell
function ukuran = bobotHistogramSpasial(data)
lebarPerBlok = single(data.BlockSize(2)*data.CellSize(2));
panjangPerBlok = single(data.BlockSize(1)*data.CellSize(1));
```

```
x = 0.5:1:lebarPerBlok;
y = 0.5:1:panjangPerBlok;
```

```
[centerBin, cellX1] = hitungOrientasiBin(x, data.CellSize(2));
[y1, cellY1] = hitungOrientasiBin(y, data.CellSize(1));
wx1 = 1 - (x - centerBin)./single(data.CellSize(2));
wy1 = 1 - (y - y1)./single(data.CellSize(1));
ukuran.x1y1 = wy1' * wx1;
ukuran.x2y1 = wy1' * (1-wx1);
ukuran.x1y2 = (1-wy1)' * wx1;
ukuran.x2y2 = (1-wy1)' * (1-wx1);
% also store the cell indices
ukuran.cellX = cellX1;
ukuran.cellY = cellY1;
```

Ilustrasi penggunaan fungsi *bobot Histogram Spasial* dapat dilihat pada Gambar 11.

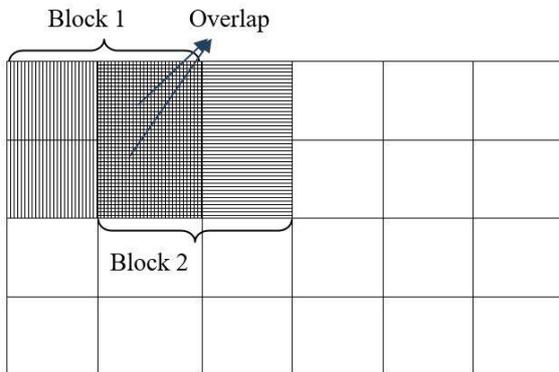


**Gambar 11.** Ilustrasi Penggunaan Fungsi *bobot Histogram Spasial*

Nilai bin pada masing-masing *block* diperoleh dengan menggabungkan nilai bin pada *cell block* sesuai dengan batas *block*, sehingga 1 *block* berisi 36 bin diperoleh dari 4 *cell* dengan masing-masing 9 bin pada setiap *cell*-nya. Selanjutnya, diperoleh panjang vektor dari hasil kali jumlah *block* vertikal= ... jumlah *block* horizontal=... dan nilai bin pada tiap *cell* dalam *block*=36 adalah 6

c. *Normalisasi*

Tahapan akhir pada ekstraksi fitur HOG yaitu melakukan normalisasi pada setiap *block* karena setiap *cell* yang telah dikelompokkan menjadi *block* biasanya memiliki *overlap* atau mengalami tumpang tindih. Ilustrasi *overlap* dapat dilihat pada Gambar 12



**Gambar 12.** Ilustrasi *Overlap*

Maka dari itu diperlukan adanya normalisasi, di mana normalisasi yang digunakan pada penelitian ini adalah L2-Hys sesuai dengan yang digunakan pada fungsi *extractHogFeatures*. Normalisasi ini akan menormalkan nilai vektor yang diperoleh menjadi 0.2 sehingga didapat panjang vektor baru yang dipakai untuk fitur pada tahap pengenalan/klasifikasi angka.

d. *SVM Train dan SVM*

Setelah melalui proses ekstraksi fitur menggunakan HOG, didapatkan nilai vektor pada citra, maka langkah selanjutnya adalah melakukan pelatihan pada citra latih

menggunakan SVM yaitu fungsi *svmTrain* untuk mendapatkan nilai koefisiennya dan akan dikelompokkan ke dalam grup yang merepresentasikan kelas dari masing-masing angka (angka 0-9).

Setelah data latih dikelompokkan, maka angka akan diklasifikasikan dengan cara mengambil data dari models (grup) dan vektor HOG dari citra uji (*svmclassify (models, testset)*).

Pelatihan pada tahap SVM Train akan menghasilkan SVM yang terlatih, dimana data SVM terlatih ini akan digunakan pada tahap pengenalan. Tahap pengenalan dilakukan dengan memasukkan nilai vektor HOG dari citra uji sebagai input pada SVM terlatih sehingga didapatkan hasil pengenalannya.

2. Hasil dan Pembahasan

Tahap pengujian dilakukan dengan cara memasukkan citra *stand kWh* meter dan melakukan pemrosesan mulai dari *grayscale*, *filter gaussian*, binerisasi, segmentasi, penyeragaman ukuran citra, ekstraksi menggunakan HOG dan klasifikasi SVM sehingga didapatkan hasil pengenalan.

Tahap pengujian ini menggunakan 30 citra untuk masing-masing kategori dan dipisahkan menjadi pengujian per-citra dan pengujian per-angka. Adapun hasil pengujian dapat dilihat pada Tabel 1.

**Tabel 1.** Tabel Hasil Pengujian

No	Jarak Pengambilan Citra	Resolusi Kamera	Persentase Pengenalan Per-citra	Persentase Pengenalan Per-angka
1	±10 cm	6 MP	30%	74,67%
2	±10 cm	10 MP	20%	65,33%
3	±10 cm	14 MP	10%	52%
4	±20 cm	6 MP	33,33%	72,67%
5	±20 cm	10 MP	46,67%	82%
6	±20 cm	14 MP	60%	84,67%
7	±30 cm	6 MP	43,33%	70,67%
8	±30 cm	10 MP	73,33%	86%
9	±30 cm	14 MP	70%	86,67%

3. Kesimpulan

Hasil pengujian menunjukkan bahwa citra pada masing-masing jarak pengambilan dan resolusi memiliki persentase pengenalan yang berbeda. Hal ini dikarenakan nilai ekstraksi ciri yang diperoleh akan berbeda sehingga mempengaruhi proses pengenalan.

Pada penelitian ini, citra yang memiliki persentase pengenalan tertinggi pada pengenalan per-citra yaitu citra dengan jarak pengambilan ±30 cm dan resolusi 10 MP dengan persentase sebesar 73,33%, sedangkan persentase tertinggi pada pengenalan per-angka yaitu citra dengan jarak pengambilan ±30 cm dan resolusi 14 MP yang memiliki persentase sebesar 86,67%.

Citra dengan pencahayaan yang terlalu terang (silau), tidak dapat dikenali karena pada tahap binerisasi

cahaya akan menutupi angka sehingga citra tidak bisa dilakukan segmentasi dengan baik dan tidak dapat dilanjutkan ketahap berikutnya.

#### Daftar Pustaka

- [1] Pujiharsono, H 2016, *Sistem Otomasi Robust Berbasis Citra untuk Mendeteksi dan Mengenali Angka Pemakaian Energi Listrik pada kWh Meter Pascabayar*, Universitas Gajah Mada, Yogyakarta.
- [2] Gunawan, R., dkk 2014, *Penerapan Optical Character Recognition (OCR) untuk Pembacaan Meteran Listrik PLN*, INFORMATIKA, Vol.10, h. 127-134.
- [3] Ning, G 2013, *Vehicle License Plate Detection and Recognition*, University of Missouri, Columbia.
- [4] Banjare, K& Massey S 2016, *Numeric Digit Classification Using HOG Feature Space and Multiclass Support Vectore Machine Classifier*, International Journal of Scientific Research and Education, Vol.4, h. 5339-5345.
- [5] Lawgali, A 2016, *Recognition of Handwritten Digits using Histogram of Oriented Gradients*, International Journal of Advances Research in Science, Engineering and Technology, Vol. 3, h. 2359-2363.
- [6] Kamble, P & Hegadi R 2015, *Handwrittem Marathi Character Recognition Using R-HOG Feature*, International Conference on Advanced Computing Technologies and Applications, Procedia Computer Science 45, h. 266-274.
- [7] Ebrahimzadeh, R2014, *Efficient Handwritten Digit Recognition Based on Histogram of Oriented Gradients and SVM*, International Journal of Computer Applications, Vol.104, h. 10-13.