

## Binarisasi Citra Menggunakan Pencocokan Piksel

Teady Matius Surya Mulyana

*Program Studi Teknik Informatika, Fakultas Teknik Dan Desain, Universitas Bunda Mulia  
Jl. Lodan Raya No.2, Daerah Khusus Ibukota Jakarta 14430  
E-mail: [tmulyana@bundamulia.ac.id](mailto:tmulyana@bundamulia.ac.id), [teadymatius@yahoo.com](mailto:teadymatius@yahoo.com)*

### ABSTRACT

*The binarization to separate between object from background of image is necessary. The problem is background of image has variations of color. Its needs process to differentiate object with its background that has variety of color. As the pre-processing of computer vision, the segmentation need be done as simple as possibly that not waiste time dan effort to do that process. Every pixel at the image has value that will produce color at the image. To simplify image recognition process, the color must change to the simple format as the binary image format. By this simplify can be gained object of image by reduce the background. Image binarization process do by matching the pixel between image contain background only with the tested image. The matching needed range of value to decide that the pixel contain of background or containt of object. Pixel matching can be use to separate object and background. This process has been implemented at a simple application sample to trigger the record image automatically when there are object at the image.*

**Keywords:** *image binarization, image segmentation, pixel matching.*

### ABSTRAK

Proses binarisasi untuk memisahkan antara obyek dengan latar belakang sangat diperlukan pada visi komputer. Permasalahan yang sering terjadi adalah latar belakang seringkali mempunyai warna yang bervariasi. Sehingga diperlukan proses untuk membedakan obyek dengan latar belakangnya yang mempunyai warna yang bervariasi tersebut. Sebagai pre-processing dari visi komputer, diperlukan proses yang sederhana yang tidak menyita waktu dan usaha untuk melakukan binarisasi tersebut. Setiap piksel pada citra mempunyai nilai tertentu yang akan menghasilkan warna pada citra. Untuk mempermudah proses pengenalan citra, warna tersebut harus disederhanakan dalam bentuk citra biner. Dengan penyederhanaan ini didapatkan citra yang merupakan obyek citra dengan menghilangkan piksel yang merupakan latar. Proses binarisasi citra dilakukan dengan pencocokan piksel antara citra latar dengan citra yang diuji. Pencocokan ini memerlukan batas toleransi tertentu. Dimana piksel yang berada pada batasan tertentu yang menganggap pada range tersebut dianggap piksel yang berisi informasi latar, sedangkan yang berada diluar batasan dianggap piksel yang berisi informasi obyek. Pencocokan piksel mampu untuk memisahkan obyek dari latar belakangnya. Proses ini diterapkan pada contoh aplikasi sederhana untuk memicu perekaman gambar dimana jika dideteksi adanya obyek, maka program akan merekan gambar secara otomatis.

**Kata Kunci:** *binarisasi citra, segmentasi citra, pencocokan piksel*

### PENDAHULUAN

Pada visi komputer diperlukan citra dalam bentuk sederhana seperti citra biner untuk dapat dianalisa oleh komputer. Hal ini diperlukan agar komputer dapat memberikan informasi sesuai dengan kebutuhan penggunaannya.

Salah satu pemanfaatan citra biner misalkan untuk mengenali adanya obyek pada suatu daerah. Mendeteksi adanya obyek pada suatu daerah dapat dapat dimanfaatkan pada berbagai bidang, seperti pada aplikasi pemantau lahan parkir, pendeteksi adanya hama, maupun untuk perekam gambar otomatis.

Proses yang sederhana diperlukan mengingat binarisasi citra hanyalah merupakan pre-processing dari keseluruhan proses pada visi komputer. Sehingga diharapkan dengan sederhananya proses, perhatian pengembang aplikasi tidak terpusat pada proses binarisasinya. Selain itu juga dengan sederhananya proses diharapkan mampu mempercepat proses binarisasi.

### Model Warna Rgb Dan Citra Biner

Salah satu model yang dipergunakan secara luas pada penelitian pengolahan citra adalah Model warna RGB, Sutoyo (Sutoyo, 2009) [1] menjelaskan bahwa

warna dari tiap piksel merepresentasikan warna dasar yang dikombinasi dari warna dasar *red* (merah), *green* (hijau) dan *blue* (biru), dimana tiap warna mempunyai 256 nilai gradasi yang bervariasi dari 0 samai 256. Model ini dinamakan model warna RGB. Variasi dari kombinasi ketiga nilai intensitas akan menghasilkan variasi warna yang berbeda. [2][3].

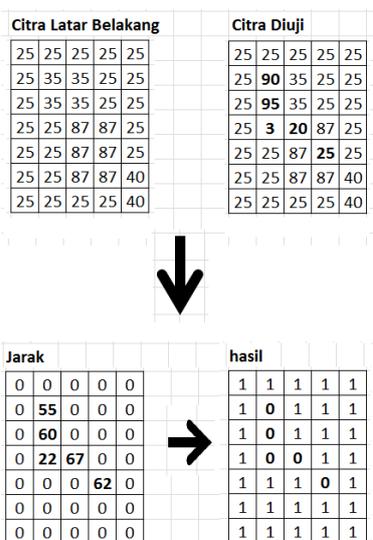
Citra biner adalah sebuah citra yang hanya terdiri dari dua warna, umumnya terdiri dari obyek yang berwarna hitam yang dianggap dan dinotasikan dengan 0, dan latar belakang yang berwarna putih dan dinotasikan dengan 1. [4][5]

**Pixelmatching / Pencocokan Piksel**

Mulyana (Mulyana, 2014)[6] menjelaskan pencocokan piksel atau *pixel matching* adalah proses pencocokan piksel latar dengan piksel citra yang diuji. Proses ini dilakukan untuk mendapatkan adanya piksel bersesuaian yang berbeda antara citra yang diuji dengan citra latar. Adanya perbedaan piksel bersesuaian dua citra yang diuji ini mengindikasikan adanya obyek pada citra yang diuji.

Sebagai ilustrasi dari pencocokan piksel dapat dilihat pada gambar 1. pada gambar tersebut diperlihatkan contoh bitmap citra latar dengan bitmap citra diuji. Dengan kondisi nilai bitmap yang idel, yaitu hanya piksel yang berisi obyek yang mempunyai nilai yang berbeda. Nilai pada bagian sel yang dicetak tebal adalah piksel yang berisi piksel yang berbeda antara citra diuji dengan citra latar.

Untuk melakukan pengenalan gerakan tangan secara optis ini terdiri dari proses akuisisi citra, *pre-processing*, ekstraksi fitur dan pengenalan citra. Setiap proses akan berkesinambungan antara satu proses dengan proses lainnya.



Gambar 1. Pixel Matching

Matriks ketiga pada gambar 1. adalah nilai jarak antara piksel bersesuaian dari citra latar dengan citra diuji. Nilai ini didapat dari rumus (1). Dimana jarak yang merupakan perbedaan citra yang didapat dari perbedaan nilai piksel latar dengan piksel uji. Jarak didapat dari harga mutlak piksel pada citra latar

dikurangi piksel pada citra uji pada posisi sel yang bersesuaian.

$$\text{Jarak}(x,y) = |\text{Latar}(x,y) - \text{Uji}(x,y)| \dots\dots\dots (1)$$

Keterangan:

- Jarak(x,y) : perbedaan nilai antara pada piksel Latar pada posisi (x,y) dengan piksel Uji pada posisi (x,y)
- Latar(x,y) : nilai piksel citra Latar pada posisi (x,y)
- Uji(x,y) : nilai piksel citra Uji pada posisi (x,y)
- x : posisi kolom piksel
- y : posisi baris piksel

Matriks ketiga adalah piksel hasil, dimana 0 melambangkan hitam dan 1 melambangkan putih. Piksel bersesuaian antara citra latar dengan citra uji yang berjarak 0 diset menjadi 1 (putih). Piksel hasil yang bernilai 1 menandakan bahwa pada sel tersebut tidak ada perbedaan piksel. Sehingga dianggap bernilai sama dengan latar atau tidak ada obyek pada sel tersebut. Sedangkan citra yang bernilai lebih dari 0 diset menjadi 0 (hitam). Menandakan bahwa pada sel tersebut terdapat perbedaan piksel antara citra latar dengan citra uji. Sehingga dianggap ada obyek pada sel tersebut.

Ilustrasi dari pemisahan obyek dengan latar dapat dilihat pada gambar 2. Pada gambar terdapat citra Latar, dengan citra berisi obyek dengan latar yang sama dengan citra latar. Hasil pemisahan menghilangkan latar dengan obyek dimana obyek ditampilkan dengan warn hitam, sedangkan warna putih adalah latar yang telah dihilangkan.

Pada kenyataannya, pencocokan piksel tidak dapat dilakukan dengan cara seperti pada rumus (1), karena adanya perubahan kondisi ketika citra latar diambil dengan kondisi ketika citra uji diambil.



Gambar 2. Pemisahan Obyek Dari Latar Belakang

Lebih lanjut, Mulyana[6] juga menjelaskan bahwa Pencocokan piksel dengan rumus (1) memerlukan memerlukan rekayasa salah satunya adalah seperti pada rumus(2), karena adanya perubahan kondisi ketika citra latar diambil dengan kondisi ketika citra uji diambil. Kondisi pencahayaan suatu latar tidak selalu sama, bahkan dalam satu detik, terjadi lebih dari satu kali perubahan kuat

cahaya. Hal ini mengakibatkan pencocokan piksel akan menghasilkan jarak, sehingga semua piksel dianggap berbeda. Gambar 3 menunjukkan cahaya yang meningkat, sehingga nilai intensitas piksel meningkat. Peningkatan intensitas piksel mengakibatkan adanya nilai jarak, sehingga semua piksel diset 0.

Citra Latar Belakang					Citra Diuji				
25	25	25	25	25	27	27	27	27	27
25	35	35	25	25	27	<b>90</b>	37	27	27
25	35	35	25	25	27	<b>95</b>	37	27	27
25	25	87	87	25	27	<b>3</b>	<b>20</b>	89	27
25	25	87	87	25	27	27	89	<b>25</b>	27
25	25	87	87	40	27	27	89	89	42
25	25	25	25	40	27	27	27	27	42

↓

Jarak					hasil				
2	2	2	2	2	0	0	0	0	0
2	<b>55</b>	2	2	2	0	<b>0</b>	0	0	0
2	<b>60</b>	2	2	2	0	<b>0</b>	0	0	0
2	<b>22</b>	<b>67</b>	2	2	0	<b>0</b>	<b>0</b>	0	0
2	2	2	<b>62</b>	2	0	0	0	<b>0</b>	0
2	2	2	2	2	0	0	0	0	0
2	2	2	2	2	0	0	0	0	0

Gambar 3. Nilai Intensitas Citra Uji Yang Lebih Tinggi Dari Citra Latar

Gambar 4 menunjukkan cahaya yang menurun atau lebih rendah, sehingga nilai intensitas piksel menurun. Penurunan intensitas piksel mengakibatkan adanya nilai jarak, sehingga semua piksel diset 0. Dapat dilihat pada piksel citra latar belakang yang bernilai 25 yang lebih besar sebanyak 2 dibandingkan piksel latar belakang pada citra yang diuji. Sehingga pada bitmap hasil semua piksel diset dengan 0 atau dianggap ada obyek karena adanya jarak antara citra latar belakang dengan citra yang diuji.

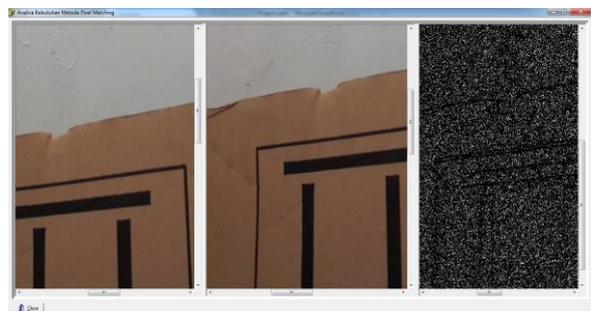
Akibat dari kondisi tersebut dapat dilihat pada gambar 5. Pada gambar tersebut terdapat tiga citra, citra pertama adalah citra yang diambil pertama, citra kedua adalah citra yang diambil beberapa milidetik berikutnya. Sedangkan citra ketiga adalah hasil proses pencocokan piksel. Terlihat citra ketiga yang seharusnya putih polos karena kedua citra sama secara kasat mata seperti yang ditunjukkan pada gambar 6, malah terlihat bercak-bercak yang menandakan kedua citra sebenarnya tidak sama secara nilai intensitas grayscale-nya.

Citra Latar Belakang					Citra Diuji				
25	25	25	25	25	23	23	23	23	23
25	35	35	25	25	23	<b>90</b>	33	23	23
25	35	35	25	25	23	<b>95</b>	33	23	23
25	25	87	87	25	23	<b>3</b>	<b>20</b>	89	23
25	25	87	87	25	23	23	85	<b>25</b>	23
25	25	87	87	40	23	23	85	85	38
25	25	25	25	40	23	23	23	23	38

↓

Jarak					hasil				
2	2	2	2	2	0	0	0	0	0
2	<b>55</b>	2	2	2	0	<b>0</b>	0	0	0
2	<b>60</b>	2	2	2	0	<b>0</b>	0	0	0
2	<b>22</b>	<b>67</b>	2	2	0	<b>0</b>	<b>0</b>	0	0
2	2	2	<b>62</b>	2	0	0	0	<b>0</b>	0
2	2	2	2	2	0	0	0	0	0
2	2	2	2	2	0	0	0	0	0

Gambar 4. Nilai Intensitas Citra Uji Yang Lebih Rendah Dari Citra Latar



Gambar 5. Akibat Dari Perbedaan Intensitas



Gambar 6. Hasil Ideal Yang Diharapkan

### Pixelmatching Dengan Toleransi

Pixel matching dengan nilai toleransi diujicoba oleh penulis pada skripsi dari Hendrik, (Hendrik, 2014)[7]. Pada skripsi tersebut, rumus (1) dimodifikasi dengan nilai toleransi seperti pada rumus (2).

$$.. Hasil(x,y) = IIF(AND(Latar(x,y)-T < Uji(x,y), Latar(x,y)+T > Uji(x,y),1,0)..... (2)$$

Keterangan:

Jarak(x,y) : perbedaan nilai antara pada piksel Latar pada posisi (x,y) dengan piksel Uji pada posisi (x,y)  
 Latar(x,y) : nilai piksel citra Latar pada posisi (x,y)  
 Uji(x,y) : nilai piksel citra Uji pada posisi (x,y)  
 T : nilai toleransi  
 x : posisi kolom piksel  
 y : posisi baris piksel

Latar(x,y) : nilai piksel citra Latar pada posisi (x,y)  
 Uji(x,y) : nilai piksel citra Uji pada posisi (x,y)  
 T1 : nilai intensitas turun terbesar  
 T2 : nilai intensitas naik terbesar  
 x : posisi kolom piksel  
 y : posisi baris piksel

Jika nilai piksel citra yang bersesuaian antara citra latar atau citra inisial dengan citra uji atau citra tiap saat setelah citra inisial berada diantara range nilai citra inisial dikurangi nilai toleransi T dan nilai citra inisial ditambah nilai toleransi T maka piksel dianggap berisi informasi nilai latar belakang, tetapi jika tidak berada diantara nilai range tersebut, maka piksel dianggap berisi informasi obyek.

**Nilai intensitas turun terbesar** adalah nilai intensitas yang turun dari semua piksel yang bersesuaian antara suatu citra inisial dengan citra inisial berikutnya. Dari semua perbedaan yang turun tersebut diambil nilai turun yan terbesar.

Pada penelitiannya mengenai perancangan perekam gambar otomatis[8], Mulyana (Mulyana, 2015) menentukan nilai toleransi berdasarkan nilai T1 dan nilai T2, dimana rumus 2 di modifikasi menjadi rumus (3).

**Nilai intensitas naik terbesar** adalah nilai intensitas yang naik dari semua piksel yang bersesuaian antara suatu citra inisial dengan citra inisial berikutnya. Dari semua perbedaan yang naik tersebut diambil nilai naik yang terbesar.

$$\text{Hasil}(x,y) = \text{IIF}(\text{AND}(\text{Latar}(x,y)-T1 < \text{Uji}(x,y), \text{Latar}(x,y) + T2 > \text{Uji}(x,y)), 1, 0) \dots \dots \dots (3)$$

Alasan tiap-tiap citra inisial dibandingkan dengan citra inisial sebelumnya adalah untuk mencari perubahan nilai intensitas tiap pengambilan citra nya.

Keterangan:

Pseudo-code untuk menentukan nilai T1 dan T2 dapat dilihat pada gambar 7. Pada gambar tersebut diperagakan semua nilai diambil berdasarkan berapa citra inisial yang dibandingkan tiap piksel yang bersesuaian dengan citra inisial sebelumnya. Variabel T1 diwakili dengan variabel 'tertinggi' dan T2 diwakili dengan variabel 'terendah'.

Jarak(x,y) : perbedaan nilai antara pada piksel Latar pada posisi (x,y) dengan piksel Uji pada posisi (x,y)

```

tertinggi := 0;
terendah := 0;
for x:=1 to ImageIni.Width do
begin
  for y:=1 to ImageIni.Height do
  begin
    for i:=2 to JumIni do
    begin
      if inisial[i,x,y]-inisial[i-1,x,y] > tertinggi then
        tertinggi := inisial[i,x,y]-inisial[i-1,x,y];
      if inisial[i,x,y]-inisial[i-1,x,y] < terendah then
        terendah := inisial[i,x,y]-inisial[i-1,x,y];
    end;
  end;
end;
end;
    
```

Gambar 7. Pseudo-code menentukan nilai T1 dan T2

Rumus 3 diterapkan dengan pseudo-code pada gambar 8. Tiap-tiap citra inisial dibandingkan dengan citra inisial pertama. Array uji[x,y] adalah nilai piksel dari citra tiap saat yang sedang diuji keberadaan obyeknya. Array hasil[x,y] adalah piksel pada posisi citra x,y dari bitmap hasil. Nilai 0 atau hitam diberikan jika didapat nilai uji berada diluar range nilai yang dapat ditoleransi berdasarkan T1 dan T2. Hal ini menandakan didapati adanya piksel yang berisi obyek. Sedangkan nilai 255 atau putih diberikan jika nilai uji dari berada diantara nilai dapat ditoleransi berdasarkan T1 dan T2. Hal ini

menandakan adanya piksel yang berisi latar belakang seperti pada citra inisial pertama.

```

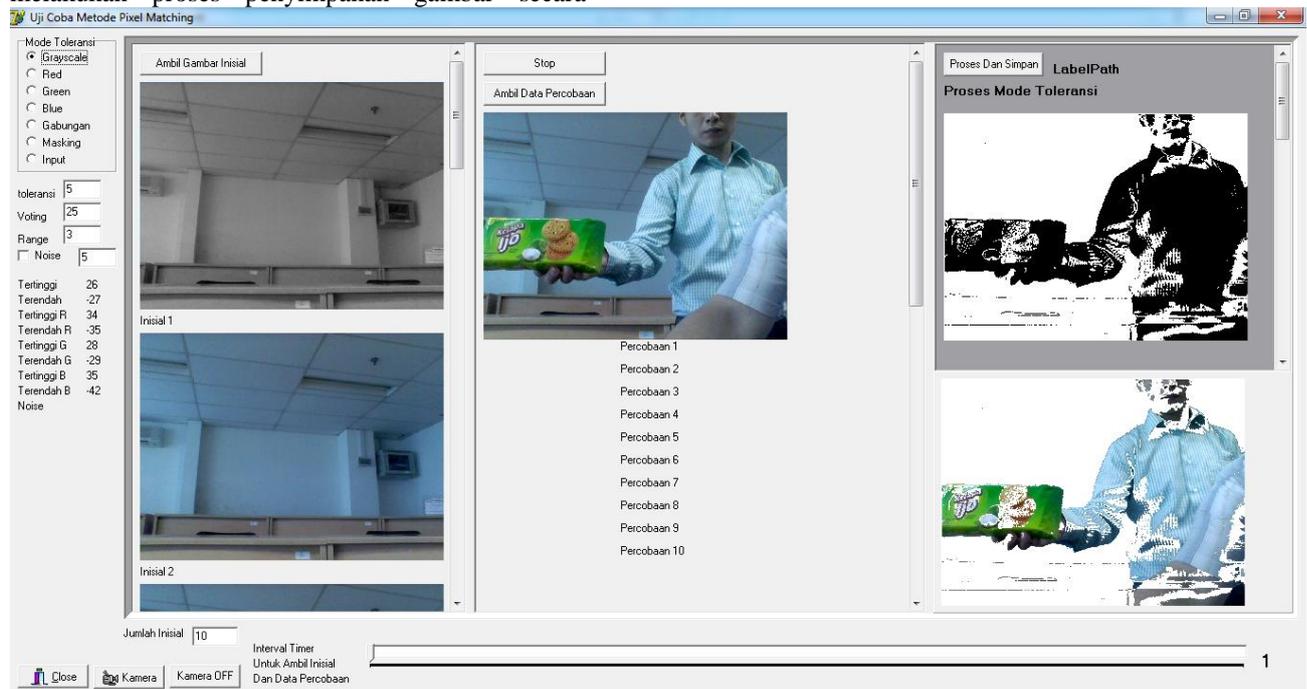
IF (uji[x,y] >= inisial[1,x,y]+terendah)
AND (uji[x,y] <= inisial[1,x,y]+tertinggi) THEN
  Hasil[x,y] := 255
ELSE
  Hasil[x,y] := 0;
    
```

Gambar 8. Pseudo-code Pixel Matching Dengan Nilai Toleransi T1 dan T2

## PEMBAHASAN

Untuk menguji keberhasilan proses, proses pixel matching diterapkan pada program penyimpan gambar otomatis. Dimana program ini akan melakukan proses penyimpanan gambar secara

otomatis jika didapati adanya obyek pada gambar yang ditangkap oleh kamera. Tampilan dari program ujicoba tersebut adalah seperti pada gambar 9.

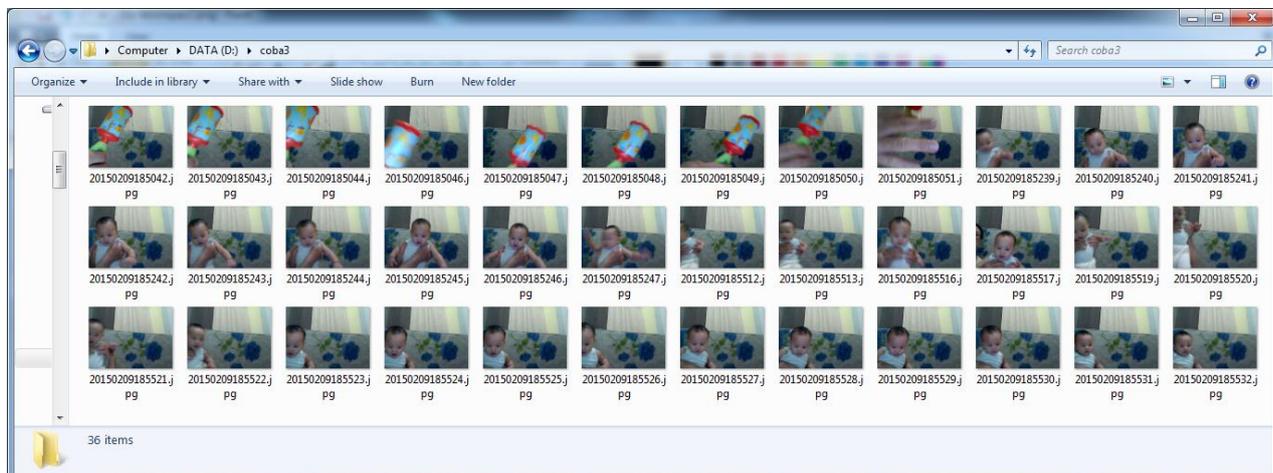


Gambar 9. Contoh Penerapan Pixel Matching.

Pada gambar tersebut terlihat pada panel terakhir sebuah citra hanya berisi obyek. Sedangkan latar belakang sudah dihilangkan. Pada panel kanan atas terdapat citra biner, sedangkan pada panel kiri terdapat citra-citra inisial.

Sedangkan pada citra tiap saat yang akan diuji ada pada panel di tengah.

File-file gambar yang berisi obyek disimpan pada folder tertentu. Gambar 10 adalah contoh file-file gambar yang disimpan.



Gambar 10. Contoh File-file Gambar Yang Disimpan

Dari hasil percobaan yang dilakukan didapatkan hasil percobaan seperti yang sudah dirangkum pada tabel 1.

Pengujian dilakukan pada beberapa kondisi latar belakang, yaitu dekat, sedang, jauh, sangat dekat, dan pada beberapa lokasi. Hasil pengujian menunjukkan bahwa tidak ada satupun gambar yang tidak berisi obyek.

Tabel 1. Hasil Percobaan.

Lokasi	Kondisi Latar	Jumlah Gambar	Gambar tanpa obyek	Gambar berisi obyek
Lokasi1	Dekat	59	0	59
Lokasi2	Sedang	46	0	46
Lokasi3	Jauh	9	0	9
Lokasi4	Sangat Dekat	5	0	5
Lokasi5	Sangat Dekat	36	0	36

## KESIMPULAN DAN SARAN

Kesimpulan dari binerisasi citra dengan *pixel matching* ini adalah:

1. Adanya perubahan nilai intensitas pada citra-citra dengan informasi citra yang sama yang diambil dengan perbedaan waktu yang sangat singkat.
2. Nilai toleransi turun dan nilai toleransi naik mampu mengatasi adanya perbedaan nilai intensitas pada kesimpulan 1.

## DAFTAR PUSTAKA

- [1] Sutoyo, T., Mulyanto, E., Suhartono V., Nurhayanti OD., Wijanarto. 2009. *Teori Pengolahan Digital*. Yogyakarta: Andi Offset.
- [2] Noor A. Ibraheem, Mokhtar M. Hasan, Rafiqul Z. Khan, Pramod K. Mishra. 2012. *Understanding Color Models: A Review*. ARPN Journal of Science and Technology. Vol. 2, No. 3, (p265-275).
- [3] Che-Yen Wen, Chun-Ming Chou. 2004. *Color Image Models and its Applications to Document Examination*. Forensic Science Journal. pp. 23-32 .Vol. 3(1).
- [4] Aviv, Rotem. 2011. *Algorithms for testing connectivity - Implementation on binary images*. Tel-Aviv University, Tel-Aviv. p:14
- [5] Mulyana, Teady.M.S., Suherman, Anggie. 2013. *Segmentasi Objek Citra Berdasarkan Perbandingan Channel RGB*. Laporan Penelitian. Jakarta: Universitas Bunda Mulia. (tidak dipublikasikan)
- [6] Mulyana, Teady Matius Surya. 2014. *Uji Coba Perbedaan Intensitas Piksel Tiap Pengambilan Gambar*. Jurnal Teknologi Informasi, Universitas Bunda Mulia. Volume 10 Nomor 2.
- [7] Hendrik. 2014. *Penerapan Fitur Matriks Populasi Piksel Untuk Pendeteksian Status Parkir Mobil*, Skripsi S1. Jakarta: Universitas Bunda Mulia. (Tidak dipublikasikan)
- [8] Mulyana, Teady.M.S., Christy, Beatrix., Ricaldo, Rico. 2015. *Perekam Gambar Otomatis*. Laporan Penelitian. Jakarta: Universitas Bunda Mulia. (tidak dipublikasikan)