

Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video

Eka Ardhianto, Wiwien Hadikurniawati dan Zuli Budiarmo

Program Studi Teknik Informatika, Universitas Stikubank

email: eka@unisbank.ac.id, hadikurniawati@gmail.com, zbudiarmo@gmail.com

Abstrak

Bidang pengolahan citra digital mulai berkembang pesat sejak ditemukannya foto digital. Hal ini berlanjut ke bidang pengolahan foto bergerak yang sampai saat ini disebut sebagai video. Kemajuan ini tidak terlepas dengan adanya semakin canggihnya teknologi digital yang menunjang pengolahan citra digital pada gambar diam dan gambar bergerak. Pada umumnya kamera digunakan hanya untuk menangkap gambar dari lingkungan luar dan kemudian di simpan pada suatu media penyimpanan. Hasil perekaman sebagian besar hanya digunakan untuk koleksi saja. Sehingga penelitian ini dilakukan untuk melakukan pengolahan gambar bergerak untuk mendeteksi jumlah sebuah atau lebih objek pada video.

Kemampuan alat antu computer dalam menyelesaikan masalah sangat berperan untuk mendeteksi jumlah objek warna pada video. Penelitian ini menerapkan metode image subtracting dan penggunaan fungsi regionprops untuk menyelesaikan masalah tersebut dengan beberapa tambahan script.

Hasil yang didapatkan dari pengimplementasian cara image subtracting dan pemaanaan fungsi regionprops sampai hasil yang dicapai pada penelitian ini menunjukkan bahwa objek dengan kandungan warna pokok dapat terdeteksi jumlahnya.

Kata kunci : Pengolahan citra digital, deteksi objek

PENDAHULUAN

Bidang pengolahan citra secara *digital* menjadi populer diminati mulai pada awal tahun 1921. Sekitar tahun 1960 baru tercatat suatu perkembangan pesat seiring dengan munculnya teknologi komputer yang sanggup memenuhi suatu kecepatan proses dan kapasitas memori yang dibutuhkan oleh berbagai algoritma pengolahan citra. Sejak itu berbagai aplikasi mulai dikembangkan. Kemampuan dasar sebuah komputer dari generasi terakhir ini selalu dikaitkan dengan fasilitas dalam bidang grafika desain dan juga multimedia yang tidak hanya dapat mengolah suatu citra atau gambar tetapi juga dalam bentuk video.

Terlebih dengan semakin mudahnya penggunaan kamera live view (web cam), akan lebih banyak pengolahan yang dapat dilakukan tidak hanya untuk gambar diam namun gambar video juga dapat dilakukan pengolahan. Salah satu pengolahan citra dalam bentuk video adalah

untuk melakukan proses pengamatan suatu objek atau benda yang tidak perlu dilakukan pengamatan secara langsung atau terus menerus pada suatu objek, namun cukup meletakan suatu kamera yang mengarah pada objek yang diinginkan lalu mengamatinya dari layar monitor. Dengan mengamati citra yang terekam kamera dapat diketahui kondisi dari objek tersebut.

Masalah timbul karena selama ini kamera hanya dapat menangkap suatu objek tetapi tidak dapat memberikan informasi tentang gerakan dari objek tersebut. Penentuan gerakan objek tersebut biasanya dilakukan dengan mengamati citra yang terekam oleh kamera. Hal ini menjadi tidak efektif bila terlalu banyak citra yang akan diamati, dan hal ini memungkinkan terjadinya kesalahan pengamatan.

Penelitian terus dikembangkan untuk melakukan pengolahan citra digital. Banyak metode yang terciptakan untuk hal hal tersebut.

Salah satunya adalah metode regionprops yang mampu mengklusterkan objek dasar pada sebuah citra digital diam. Dan metode lain yang dapat mendukung pengolahan citra diam. Berdasarkan latar belakang masalah diatas, penulis mengemukakan gagasan membuat perangkat lunak untuk mendeteksi gerakan objek dengan mengimplementasi Metode Image Subtracting dan Metode Regionprops untuk mendeteksi jumlah objek berwarna RGB dan lokasi koordinat pada File Video.

TUJUAN DAN MANFAAT

Tujuan penelitian ini adalah dapat terdeteksinya objek benda dengan yang memiliki warna pokok RGB pada suatu file gambar bergerak serta dapat diketahui jumlah benda yang dimaksud. Berkaitan dengan hasil yang dimaksudkan manfaat yang diharapkan adalah metode yang digunakan dapat menghasilkan informasi lokasi pendeteksian objek dan dapat digunakan untuk menghitung jumlah objek yang terdapat pada frame.

TINJAUAN PUSTAKA

1. Definisi Citra Digital

Menurut kamus Webster, citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra dapat dikatakan sebagai citra digital jika citra tersebut disimpan dalam format digital (dalam bentuk *file*). Hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain jika akan diolah dengan komputer harus diubah dulu menjadi citra digital.

Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks pada gambar 1.1.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Gambar 1. Matriks citra digital N x M

Berdasarkan gambar 1 matriks citra digital, Suatu citra f(x,y) dalam fungsi matematis dapat dituliskan persamaan (1) sebagai berikut :

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x,y) \leq G-1 \dots\dots\dots (1)$$

dimana :

M = jumlah piksel baris (*row*) pada array citra

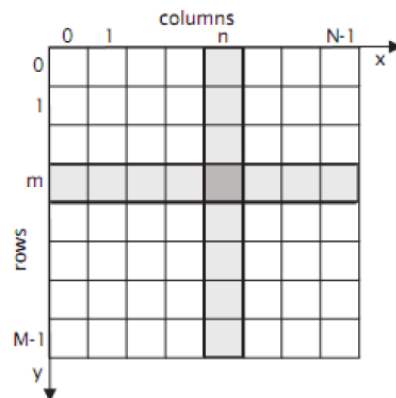
N = jumlah piksel kolom (*column*) pada array citra

G = nilai skala keabuan (*graylevel*)

Besarnya nilai **M**, **N** dan **G** pada umumnya merupakan perpangkatan dari dua seperti yang terlihat pada persamaan (3.2).

$$M = 2m ; N = 2n ; G = 2k \dots\dots\dots (2)$$

Dimana nilai **m**, **n** dan **k** adalah bilangan bulat positif. Interval (0,G) disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan 28 = 256 warna (derajat keabuan). (RD. Kusumanto : 2011)

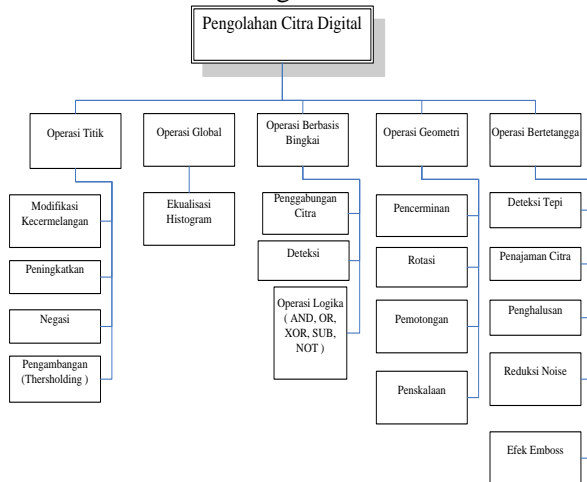


Gambar 2. Representasi citra digital dalam 2 dimensi

2. Pengolahan Citra Digital

Kegiatan untuk mengubah informasi citra fisik non digital menjadi digital disebut sebagai pencitraan (*imaging*). Citra digital dapat diolah dengan komputer karena berbentuk data numeris. Suatu citra digital melalui pengolahan citra digital (*digital image processing*) menghasilkan citra digital yang baru termasuk di dalamnya adalah perbaikan citra (*image restoration*) dan peningkatan kualitas citra (*image enhancement*).

Seperti yang terlihat dalam gambar 3 operasi – operasi dasar pada pengolahan citra digital.



Gambar 3. Operasi-operasi dasar pada Pengolahan Citra Digital

Operasi – operasi dasar pengolahan citra digital antara lain:

1. Operasi Titik
2. Operasi global
3. Operasi berbasis bingkai (*frame*)
4. Operasi geometri
5. Operasi bertetangga

3. Segmentasi

Suatu citra *menjadi* beberapa daerah atau objek yang dimilikinya dengan cara melakukan segmentasi terhadap objek yang ada dalam citra tersebut. Menurut Castleman (1996), segmentasi citra merupakan suatu proses memecah suatu citra menjadi banyak segmen atau bagian daerah yang tidak saling bertabrakan (*non-overlapping*).

Dalam citra digital daerah hasil segmentasi tersebut merupakan kelompok pixel yang bertetangga atau berhubungan.

Segmentasi dapat dilakukan melalui beberapa pendekatan, menurut Castleman terdapat 3 macam pendekatan, yaitu :

1. Pendekatan batas (*boundary approach*)
2. Pendekatan tepi (*edge approach*)
3. Pendekatan daerah (*region approach*)

Proses segmentasi digunakan dalam berbagai penerapan, meskipun metode yang digunakan bervariasi, namun memiliki tujuan yang sama yaitu mendapatkan representasi sederhana yang berguna dari suatu citra.

4. Nilai Batas Ambang

Thresholding adalah istilah populer dalam Pengolahan Citra Digital untuk sebuah nilai batas ambang, yang merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (Katz,2000). Dalam pelaksanaannya *Thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*.

Thresholding digunakan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang lebih besar dari nilai *threshold T* sebagai latar depan dan yang lebih kecil dari nilai *threshold T* sebagai latar belakang. Biasanya pengaturan nilai *threshold* dilakukan berdasarkan histogram *grayscale* (Gonzales dan Woods, 2002; Fisher, dkk, 2003; Xiaoyi dan Mojon, 2003).

Pencarian metode *threshold* yang sederhana tidak memerlukan pengetahuan yang lebih tentang citra dan *thresholding* pun bisa bekerja pada citra yang memiliki *noise* , Metode iterative merupakan pendekatan yang baik untuk dilakukan seperti berikut :

1. Memilih *initial* dari *threshold (T)*. Dapat dilakukan secara random atau menurut metoda yang diinginkan

2. Citra ini disegmentasikan ke dalam piksel objek dan piksel *background* seperti persamaan 3.5 dan 3.6.

$$G_1 = \{f(m,n) : f(m,n) > T\} \dots\dots\dots(5)$$

$$G_2 = \{f(m,n) : f(m,n) \leq T\} \dots\dots\dots(6)$$

di mana :

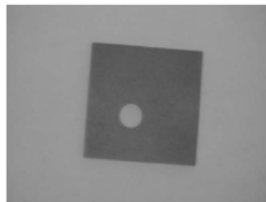
- G_1 adalah nilai piksel objek.
- G_2 adalah nilai piksel *background*.
- $f(m,n)$ adalah nilai dari piksel yang terletak pada m^{th} kolom dan n^{th} baris

3. Hitung nilai rata – rata *gray value* μ_1 dan μ_2 pada piksel dalam G_1 dan G_2

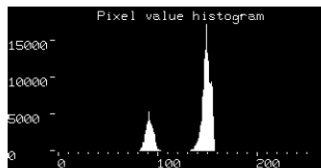
4. Hitung nilai *threshold* baru dengan persamaan 3.7 :

$$T = \frac{1}{2} (\mu_1 + \mu_2) \dots\dots\dots(7)$$

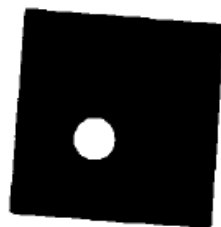
5. Ulangi langkah ke 2 sampai dengan langkah ke 4 dengan nilai T yang berbeda sampai nilai *threshold* yang baru sama dengan nilai yang sebelumnya.



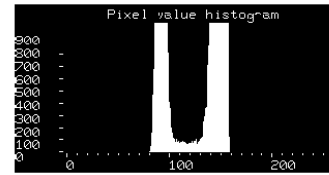
Gambar 4. Objek dengan warna gelap dan diletakkan pada *background* terang



Gambar 5. Histogram dari Gambar 4



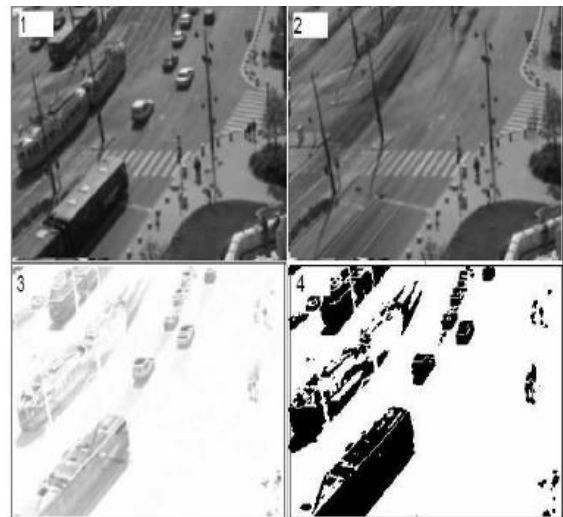
Gambar 6. Objek yang telah *dithreshold* dengan nilai *threshold* sekitar 120



Gambar 7. Histogram dari gambar 6

5. Background Subtraction

Background subtraction adalah proses untuk menemukan objek pada gambar dengan cara membandingkan gambar yang ada dengan sebuah model latar belakang. Prosedur *background subtraction* terdiri dari 3 tahap, yaitu *pre-processing*, *background modeling*, dan *foreground detection*.



Gambar 8. *Background subtraction*

(1) gambar sekarang, (2) *background model*, (3)- hasil *background subtraction*, (4) hasil *background subtraction* setelah *threshold*

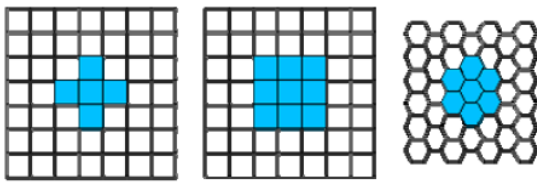
Tahapan dalam *Background subtraction*

a. Pre-processing

Pada tahap ini data mentah dari kamera (atau input lainnya) diproses menjadi bentuk yang dapat dimengerti oleh bagian program lain. Pada tahapan awal ini dilakukan *noise removal* dan eliminasi objek kecil pada gambar agar menjadi lebih informatif. Eliminasi objek kecil dilakukan dengan menggunakan *mathematical morphology* yaitu transformasi *Opening*.

- b. Background modeling
- c. *Foreground* detection
- d. *Median filter*
- e. *Connected* component labeling

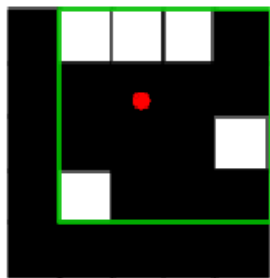
Algoritma *Connected Component Labeling* digunakan untuk melabeli tiap objek pada gambar binary dengan suatu label unik. Pengelompokan piksel sebagai satu objek ditentukan dari status ketetanggaan mereka. Sebuah piksel disebut bertetangga dengan piksel lain apabila piksel tersebut bertetangga langsung dengan piksel lain itu ataupun piksel lain itu merupakan tetangga dari tetangga piksel tersebut. Kriteria sebuah piksel merupakan tetangga dari piksel lain dapat berupa *4-connectivity*, *6-connectivity* ataupun *8-connectivity* seperti terlihat pada gambar 9.



Gambar 9. 4, 8 dan 6-connectivity

f. *Regionprops*

Dalam fungsi *regionprops* sebuah obyek direpresentasikan sebagai sebuah *region* dengan pendekatan bentuk persegi panjang. Gambar 10 menunjukkan sebuah *region* dari kumpulan piksel berwarna putih yang direpresentasikan dengan pendekatan bentuk persegi panjang.



Gambar 10. Representasi region dengan pendekatan bentuk persegi panjang

6. Matlab

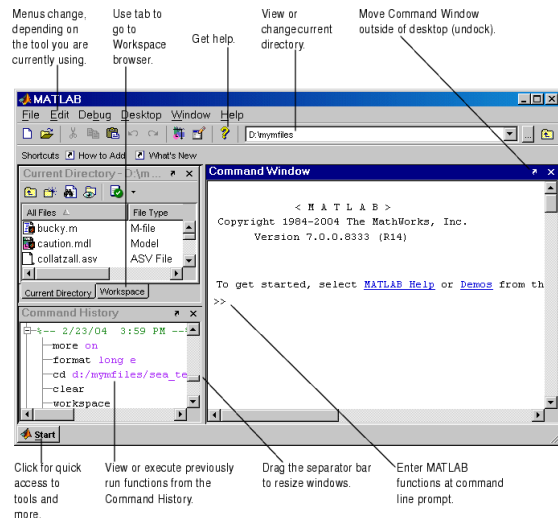
Apakah MATLAB itu?, Matlab adalah sebuah bahasa dengan performa tinggi untuk

komputasi teknik. Dari wikipedia ensiklopedia bebas, matlab adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat.

MATLAB mengintegrasikan komputasi, visualisasi dan pemrograman pada perangkat yang mudah digunakan dimana permasalahan dan penyelesaian dapat diekspresikan dengan notasi matematika yang familiar. Matlab dikembangkan oleh MathWork.Inc www.mathwork.com.

a. Lingkungan Matlab

Ketika memulai MATLAB, hal pertama yang anda lihat adalah MATLAB desktop. Yang terdiri dai perangkat GUI untuk memanajemen file, variabel dan aplikasi yang berkaitan dengan MATLAB. Pertama anda memulai MATLAB, desktop muncul dengan lay-out default seperti terlihat pada gambar 11 anda dapat mengubah sesuai kebutuhan seperti merubah ukuran, menggeser dan menutup tools.

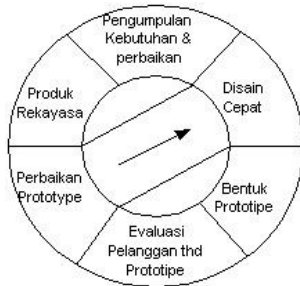


Gambar 11. Lngkungan Matlab

METODE PENELITIAN

Metode penelitian yang digunakan untuk melakukan implementasi metode image subtracting dan *regionprops* untuk mendeteksi jumlah dan likasi koordinat objek warna adalah dengan menggunakan model prototipe sebagai metode pengembangan perangkat lunak (software).

Model pengembangan perangkat lunak dengan model prototipe ini akan menghasilkan sebuah aplikasi dalam bentuk prototipe sebelum aplikasi tersebut memasuki tahap design. Dalam fase ini, prototype yang telah dirancang akan dievaluasi. Tahap ini akan terus menerus diulang sampai aplikasi benar benar sesuai dengan keinginan. Apabila prototype telah selesai, maka tahapan aplikasi akan kembali berlanjut ketahap design. Gambar 12 menjelaskan bagaimana urutan proses pengembangan perangkat lunak dengan model prototipe.



Gambar 12 . Prototyping Method

HASIL DAN PEMBAHASAN

1. Bahan Penelitian

Penelitian ini menggunakan bahan inputan berupa file gambar kontinyu atau file live video yang dihasilkan dari capture kamera atau webcam yang terdapat pada perangkat keras komputer dengan resolusi 600 x 800 piksel dengan format berwarna.

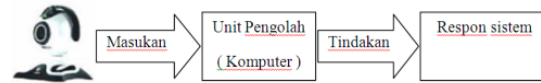
2. Pemilihan Perangkat Lunak Dan Perangkat Keras

Dalam melakukan implementasi dan pengkonstruksian aplikasi dari mode rancangan menjadi sebuah aplikasi jadi, pada penelitian ini digunakan perangkat lunak Matlab dan komputer laptop dengan spesifikasi prosesor intel core2 duo 2.00 Ghz, RAM 1.75GB dan VGA shared 250 MB.

3. Alur Algoritma

Alur perancangan algoritma yang digunakan untuk membuat aplikasi ini adalah dengan melakukan tahapan pembuatan suatu system pengolahan citra yang mampu mendeteksi gerakan objek berwarna merah yang tertangkap oleh webcam.

Seperti terlihat pada gambar, sistem ini akan merespon setiap gerakan dari objek berwarna merah untuk dapat terdeteksi secara langsung (*real time*). Objek yang terdeteksi akan ditandai dengan *boundingbox* dan jumlah objek yang akan terdeteksi secara otomatis mengikuti setiap gerakan objek. Dan seluruh gerakan objek yang terdeteksi akan disimpan dalam file video.



Gambar 13 Diagram blok sistem pendeteksi gerakan

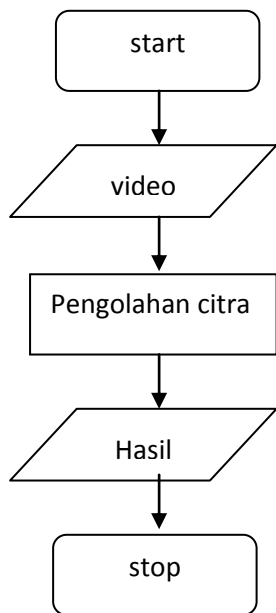
Dalam proses pengolahan citra untuk mendeteksi gerak berdasarkan warna dari input citra RGB secara garis besar prosesnya dapat dibagi menjadi 2 tahapan antara lain :

1. Mendeteksi warna dengan metode *image subtracting*
2. Mendeteksi gerakan objek dengan fungsi *regionprop*

Proses – proses dari ke dua tahapan diatas akan terus berulang pada setiap frame dari file video yang diakuisisi oleh sistem dan akan berhenti setelah sistem dihentikan.

1) Perancangan Sistem

Sistem ini menggunakan kamera *webcam* yang sudah tersedia atau *built in* pada *perangkat laptop*, karena lebih praktis. Pada dasarnya citra bergerak atau video merupakan gabungan dari beberapa citra diam yang berubah ubah dengan kecepatan tertentu, sehingga mata manusia menangkapnya sebagai citra yang bergerak.



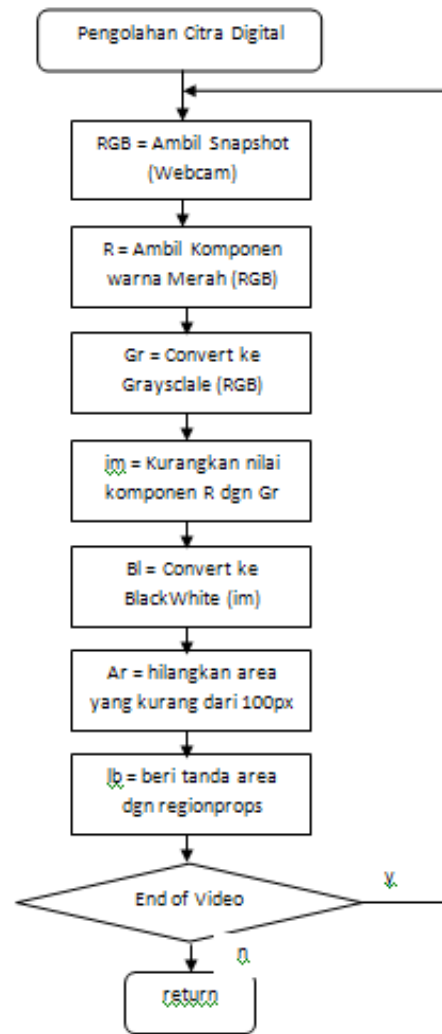
Gambar 14 diagram flowchart sistem deteksi gerak

2) Perancangan Sistem Pendeteksian Warna

Perancangan sistem pada penelitian ini menggunakan perangkat lunak Matlab 7 untuk analisis citra sehingga dapat mendeteksi gerakan objek yang berwarna merah. Dalam sistem ini pertama – tama program akan mengambil menginisialisasi *webcam* pada *laptop* . Setelah itu dilakukan pemanggilan terhadap webcam tersebut dan perintah untuk merekam setiap citra yang ada.

Berdasarkan gambar 15 diagram flowchart program pendeteksi gerakan secara keseluruhan ini terdiri dari beberapa bagian sebagai berikut :

1. Pengambilan Gambar dari Webcam
2. Pengambilan Informasi Komponen Warna
3. Mengubah Citra RGB menjadi Grayscale
4. Image subtraction
5. Image to Black White (biner)
6. Black white area open
7. Black white label



Gambar 15 diagram flowchart program pendeteksi gerakan secara keseluruhan

IMPLEMENTASI

Tahapan ini dilakukan setelah perancangan sistem selesai dilakukan dan selanjutnya diimplementasikan pada bahasa pemrograman yang digunakan. Tujuan implementasi sistem adalah untuk menerapkan perancangan yang telah dilakukan terhadap perangkat lunak sehingga nantinya maksud dan tujuan pembangunan perangkat lunak dapat tercapai.

- 1) Proses Akuisisi Citra
 - a. Pengaktifan Webcam
 - b. Pengambilan Gambar
- 2) Proses Pengolahan Citra

Video adalah suatu citra atau frame yang putar secara berurutan dengan kecepatan tertentu sehingga terlihat seperti bergerak. Maka dalam sistem ini video akan diolah berdasarkan citra atau frame yang diakuisisi oleh sistem secara berurutan. Adapun tahapan proses pengolahan citra yang harus dilalui suatu citra berwarna merah agar dapat terdeteksi oleh sistem deteksi gerak berdasarkan warna dapat dilihat pada tahapan proses pengolahan citra pada sistem deteksi gerak berdasarkan warna pada file citra RGB. Adapun pengolahan yang harus dilalui agar dapat mendeteksi objek berwarna merah adalah sebagai berikut :

1. Pengambilan Informasi Komponen Warna

Citra inputan yang berupa citra RGB akan diambil komponen warna merahnya (red), sehingga menghasilkan citra baru yang hanya memiliki komponen warna merah saja.

2. Mengubah Citra RGB menjadi Grayscale

Kemudian masih dari citra inputan yang berupa citra RGB yang kemudian akan diubah menjadi citra grayscale menggunakan fungsi RGB to gray yang ada pada matlab 7.

3. Image subtraction

Hasil dari langkah ke 3 atau citra grayscale akan *subtract* (dikurangi) dengan hasil dari langkah ke 2 yaitu citra RGB yang telah diambil komponen warna merahnya. Sehingga akan menghasilkan sebuah citra yang memiliki nilai grayscale yang menunjukkan area berwarna merah dari input citra RGB. Pada langkah (2) dan (3) ini selanjutnya dianggap sebagai proses image subtracting. Source code (1) adalah penggalan kode untuk melakukan proses image subtracting.

<p>Source code (1)</p> <pre> diff_im = imsubtract(data(:,:,1), rgb2gray(data)); </pre>
--

4. Image to Black White (biner)

Proses image subtracting sebelumnya memiliki nilai grayscale yang kemudian dari

citra grayscale ini kemudian akan diubah menjadi citra hitam putih atau black white atau biner menggunakan fungsi im2bw pada matlab 7.

5. Black white area open

Citra biner dari hasil pengkonversian ke mode black and white akan kembali diproses untuk mendeteksi lokasi yang dianggap objek dengan cara menghilangkan piksel – piksel yang berukuran kecil.

6. Black white label

Untuk melabeli semua yang piksel dari citra biner yang dihasilkan dari tahapan proses sebelumnya. Pada proses ini memanfaatkan fungsi regionprops untuk memberikan nilai pada area yang telah dideteksi pada proses sebelumnya.

Pada proses ini akan selanjutnya akan dimanfaatkan untuk mendeteksi jumlah objek yang terdeteksi dari hasil proses sebelumnya untuk ditampilkan menjadi informasi.

<p>Source code (2)</p> <pre> diff_im2 = im2bw(diff_im,0.08); diff_im3 = bwareaopen(diff_im2,100); bw = bwlabel(diff_im3, 8); </pre>
--

7. Object Counting

Untuk mengetahui jumlah objek yang terdapat pada gambar yang terdeteksi, implementasi dari proses adalah memanfaatkan fungsi regionprops dengan melakukan perulangan jumlah objek yang telah diketahui. Source code (3) menunjukkan proses pendeteksian lokasi objek yang telah di deteksi.

<p>Source code (3)</p> <pre> hold on for object = 1:length(stas) bb = stats(object).BoundingBox; bc = stats(object).Centroid; </pre>

```

rectangle('Position',bb,'EdgeColor','r','LineWi
dth',2)

plot(bc(1),bc(2), '-m+')

a=text(10,430,num2str(length(stats)));

%a=text(bc(1)+15,bc(2), strcat('X: ',
num2str(round(bc(1))), ' Y: ',
num2str(round(bc(2)))));

set(a, 'FontName', 'Arial', 'FontWeight',
'bold', 'FontSize', 30, 'Color', 'red');

%b=strcat('X: ', num2str(round(bc(1))), '
Y: ', num2str(round(bc(2))))

while (k>=90)

c=text(20,430,'BERHASIL... o(^)o');

set(c, 'FontName', 'Arial', 'FontWeight',
'bold', 'FontSize', 15, 'Color', 'yellow');

end

end

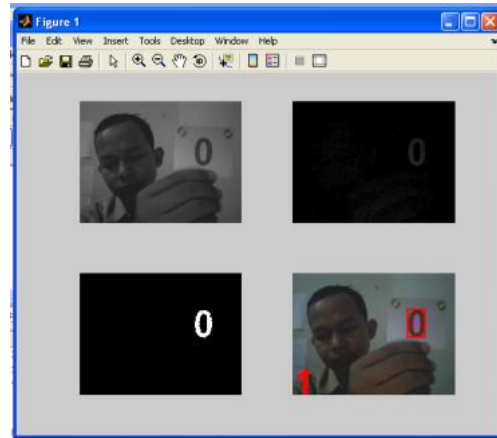
hold off.....
    
```

PENGUJIAN

Proses pengujian suatu program bertujuan untuk memastikan bahwa program yang telah dibuat dapat di implementasikan secara akurat dan tepat. Sehingga pengguna dapat menggunakan program ini dengan baik sesuai dengan fungsi program tersebut dibuat.

Sesuai dengan tujuan dari sistem ini yaitu menghasilkan aplikasi yang dapat mendeteksi objek berwarna merah pada file video, maka pengujian sistem deteksi gerak berdasarkan file video menggunakan video yang terekam secara langsung menggunakan kamera *webcam* yang sudah tersedia atau *built in*.

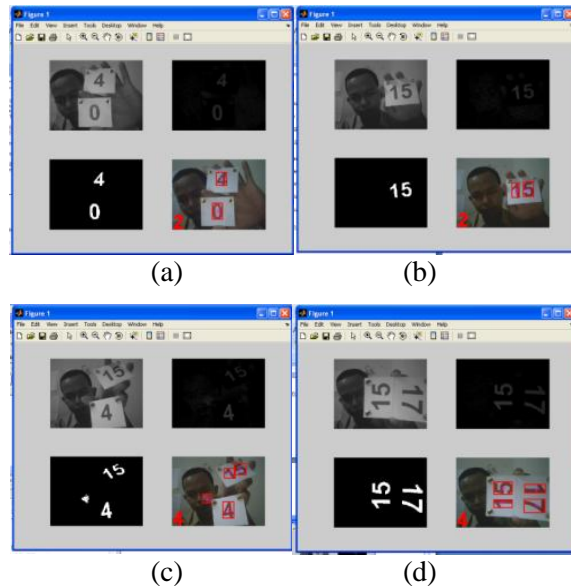
Gambar 16 memberikan gambaran bentuk hasil pengujian yang dilakukan pada gambar objek dengan komponen warna merah.



Gambar 16 Pengujian deteksi objek warna merah tunggal

Pada pengujian dilakukan dengan menggunakan objek dengan warna merah. Dan dapat dideteksi jumlah objek dengan benar. Seperti terlihat pada gambar 17 dengan menggunakan objek jamak, pengujian jumlah tetap dapat terhitung dengan benar.

Permasalahan terjadi pada saat sebuah objek terbelah dengan objek lain yang tidak sewarna, sehingga menyebabkan objek utama berwarna merah menjadi terpecah menjadi beberapa objek, atau dikenali sebagai bukan objek tunggal seperti terlihat pada gambar 5.5 c, terdapat objek yang seharusnya bukan objek yang dimaksudkan.



Gambar 17. Pengujian dengan beberapa objek

KESIMPULAN

Dari pelaksanaan penelitian dan pengujian yang dilakukan dapat ditarik kesimpulan yaitu :

1. Penggunaan cara image subtracting dapat digunakan untuk melakukan pendeteksian objek berdasarkan warna dengan mengambil warna yang ditentukan dan dikurangkan dengan warna rerata dari objek asil.
2. Fungsi regionprop yang digunakan untuk memberikan labe pada objek dari citra yang telah diolah dapat digunakan untuk menghitung jumlah objek yang dideteksi dengan cara image subtracting.
3. Penggabungan image subtracting dan fungsi region props dapat memberikan hasil baik untuk menghitung pengenalan sebuah objek berdasarkan warna.
4. Kendala yang terjadi adalah jika sebuah objek utuh kemudian terhalang oleh objek lain yang tidak sewarna akan menghasilkan objek dikenali sebagai objek jamak.

SARAN

1. Warna objek yang dapat dikenali adalah warna objek yang merupakan warna pokok, sehingga dipandang perlu untuk dikembangkan untuk melakukan penambahan utilitas pengenalan dengan menggunakan warna campuran.
2. Pengembangan lebih lanjut diharapkan model penggabungan ini dikembangkan dengan menambahkan model pendeteksian objek secara utuh dan bukan objek lain yang sebenarnya bukan dianggap objek.
3. Penelitian ini tidak termasuk digunakan pengenalan objek dengan menggunakan dasar bentuk, sehingga juga dipandang perlu untuk dilakukan penelitian dan percobaan lebih lanjut dengan menambahkan fungsi pengenalan bentuk objek untuk dapat lebih spesifik pengenalan objek.

DAFTAR PUSTAKA

Castleman K.R. (1996). *Digital image processing*. New Jersey : Prentice Hall.

Catur, Ibnu Mustofa. (2008) *Monitoring Gerakan pada Ruangan Menggunakan Webcam dan Motor Stepper*, Skripsi, Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Malang, Malang

Mahanani, Putri R. (2010) *Tracking Obyek Menggunakan Metode Counterpropagation*, Institut Teknologi Sepuluh Nopember, Surabaya

Munir, R. (n.d.). *Pengolahan Citra Dijital*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital/ab-2_Pembentukan%20Citra.pdf. Diakses tanggal 28 Januari 2013.

Munir, R. (n.d.). *Pengolahan Citra Dijital*. http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital/ab-4_Operasi-operasi%20Dasar%20Pengolahan%20Citra%20Dijital.pdf. Diakses tanggal 28 Januari 2013.

Pressman, Roger S.(2002) *Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku 1)*, Andi, Yogyakarta

Putra, Darma. 2010. *Pengolahan Citra Digital*, Andi, Yogyakarta

Sudirman, (2010). *Pengolahan Citra*. http://sorisoga.blogspot.com/2010/10/pengolahan-citra_20.html

Winarno. E. (n.d.). *Blognya Edy Winarno | Blog pribadi dan seputar dunia Informatika*. Diakses 13 Maret 2013, dari <http://edywinarnosemarang.wordpress.com/materi-kuliah>

Wijaya, Marvin CH & Agus Prijono, (2007) *Pengolahan Citra Digital Menggunakan Matlab*. Penerbit : Infomatika, Bandung

Yoga, Benedictus Budi Putranto, Widi Hapsari dan Katon Wijana. (2010). *Segmentasi Warna Citra Dengan Detekdi Warna HSV Untuk Mendeteksi Objek*, Universitas Kristen Duta Wacana Yogyakarta, Yogyakarta