

NAMA : MUHAMAD SLAMET ARIYANTO
NIM : 18041002
KELAS : 4G
MK : UAS COMPUTER VISION

JURNAL 1

PENGEMBANGAN COMPUTER VISION SYSTEM SEDERHANA UNTUK MENENTUKAN KUALITAS TOMAT

DI TULIS OLEH :

RUDIATI EVI MASITHOH , BUDI RAHARDJO , LILIK SUTIARSO , AGUS HARDJOKO

A. ISI

Kualitas merupakan karakter suatu produk yang menunjukkan derajat keunggulan suatu produk atau kemampuannya dalam memenuhi kebutuhan tertentu (Abbott, 1999). Warna merupakan parameter utama dari konsumen untuk menentukan kualitas (Francis, 1995). Instrumen pengukur warna semakin berkembang, sehingga pengamatan yang sebelumnya hanya dapat dilakukan oleh indra manusia sekarang dapat dilakukan dengan instrumen. Instrumen pengukur warna konvensional, misalnya colorimeters dan spektrofotometer, menyediakan informasi warna dalam format XYZ, RGB dan $L^*a^*b^*$ (Mendoza dkk., 2006). Beberapa metode subyektif untuk mengukur warna telah dikembangkan antara lain color chart dan rating scale (Grierson dan Kader, 1986). CVS merupakan sistem berbasis pada pengembangan algoritma dan komputer untuk menstimulasi visualisasi manusia agar dapat secara otomatis mengekstrak informasi berharga dari suatu obyek (Panigrahi dan Gunasekaran, 2001).

Mata manusia adalah sebuah sistem canggih yang melakukan respon atas rangsangan visual. Secara fungsional, computer vision dan penglihatan manusia adalah sama, dengan tujuan menafsirkan data spasial yaitu data yang diindeks oleh lebih dari satu dimensi. Meskipun demikian, computer vision tidak dapat diharapkan untuk mereplikasi persis seperti mata manusia. Hal ini disebabkan pengetahuan tentang bagaimana sistem mata dan otak bekerja belum sepenuhnya dipahami, sehingga tidak dapat merancang sebuah sistem untuk mereplikasi secara tepat fungsi mata manusia. Yang dapat dilakukan adalah teknik computer vision yang dapat mereplikasi dan, dalam beberapa kasus bahkan memperbaiki sistem penglihatan manusia (Nixon dan Aguado, 2008). CVS terdiri dari komputer yang merupakan analog dari otak manusia, sensor atau kamera digital (analog dari mata manusia), dan sistem penerangan berupa sumber cahaya untuk memfasilitasi akuisisi citra. Secara umum, terdapat 3 (tiga) operasi utama dalam CVS yaitu akuisisi citra (image acquisition), pengolahan citra (image processing), dan penerjemahan citra (image understanding) (Panigrahi dan Gunasekaran, 2001). Secara lebih rinci, tahapan-tahapan dalam analisis citra adalah: 1) pembentukan citra, dimana citra suatu obyek diambil dan disimpan dalam komputer, 2) pre-proses citra, dimana kualitas citra digital diperbaiki untuk meningkatkan detail citra, 3)

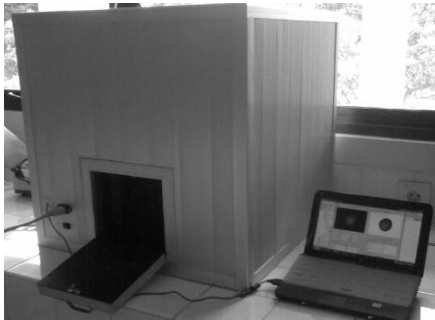
segmentasi citra, dimana citra obyek diidentifikasi dan dipisahkan dari latar belakangnya, 4) pengukuran citra, dimana beberapa fitur-fitur yang signifikan dikuantitaskan, dan 5) interpretasi citra, dimana fitur-fitur yang telah diekstrak kemudian diinterpretasikan (Mery dan Pedreschi, 2005). CVS membantu kerja manusia dalam mengambil keputusan secara cepat, akurat, dan konsisten. Namun demikian, kinerja CVS memerlukan kajian dan pengembangan yang berkelanjutan karena keterbatasannya untuk melakukan analisis kuantitatif secara mendalam (Abdullah, dkk., 2004). Hal ini disebabkan karena kamera yang digunakan pada CVS biasanya hanya beroperasi pada panjang gelombang 10-4 – 10-7 m yang hanya dapat menghasilkan informasi parameter eksternal dari obyek dan bukan parameter internal. Tetapi, beberapa penelitian telah dilakukan untuk mengetahui hubungan antara kenampakan eksternal yang diperoleh dari CVS dengan parameter internalnya, seperti kandungan kadar air (Masithoh dan Kusuma, 2008), klorofil (Kawashima dan Nakatani, 1998), serta kadar gula dan pH pada jeruk (Kondo dkk., 2000). Dengan mengembangkan metode CVS ini maka pengukuran dapat dilakukan secara berulang untuk memperoleh hasil yang lebih akurat pada sampel yang sama karena menggunakan metode non-destruktif, serta dapat digunakan di lapangan untuk mengukur kualitas secara cepat. Penelitian ini bertujuan untuk mengukur RGB warna dan memprediksi kualitas buah secara non-destruktif berdasarkan warna untuk mendapatkan informasi hayati secara real-time dengan menggunakan tomat sebagai obyek validasi.

METODE PENELITIAN

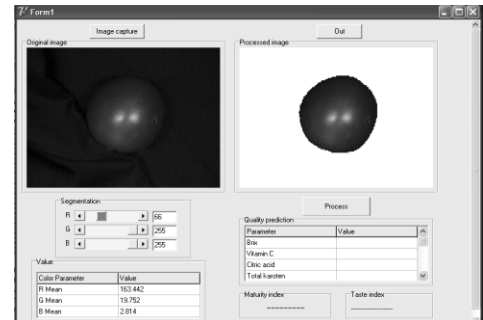
Penelitian ini mencakup dua tahap, yaitu pembuatan instrumen penangkap citra dan pembuatan perangkat lunak untuk menentukan kualitas tomat berdasarkan parameter citra dan jaringan syaraf tiruan. CVS yang dibangun terdiri dari (1) kotak untuk meletakkan obyek dan menangkap citra, (2) webcam untuk mendapatkan gambar dalam format digital, (3) sumber cahaya yang berasal dari lampu Light Emitting Diode (LED) sepanjang 5 meter dan 4 (empat) buah lampu TL masing-masing berdaya 8W, (4) komputer untuk penyimpanan gambar dan mengolah data, berikut monitor untuk menampilkan citra, dan (5) analisis citra yang dilengkapi dengan jaringan syaraf tiruan (JST) untuk menentukan kualitas tomat.

Hasil dan Penelitian

CVS yang dikembangkan tampak seperti Gambar 1. CVS terdiri dari kotak aluminium tertutup yang di dalamnya terdapat webcam, lampu LED dan 4 buah lampu TL sebagai sumber cahaya, serta komputer sebagai pengolah citra dan penampil data hasil ekstraksi citra. Citra yang diambil kemudian diolah dengan program pengolah citra yang dibuat. Gambar 2 merupakan tampilan antar muka atau graphical user interface (GUI) dari perangkat lunak pengolah citra yang dibuat. Bagian-bagian dari GUI ini adalah citra awal dan citra hasil segmentasi, nilai rata-rata warna Red Green Blue (RGB) dari citra yang telah disegmentasi, dan prediksi nilai kualitas tomat yang meliputi Brix, vitamin C, asam sitrat, dan gula total. Bagian-bagian dari GUI ini adalah citra awal dan citra hasil segmentasi, nilai rata-rata warna Red Green Blue (RGB) dari citra yang telah disegmentasi, dan prediksi nilai kualitas tomat yang meliputi Brix, vitamin C, asam sitrat, dan gula total.



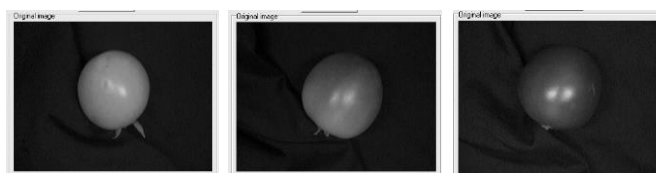
Gambar 1. *Computer vision system* yang dikembangkan untuk prediksi kualitas tomat



Gambar 2. Tampilan antar muka

Warna dan Kualitas Tomat

Gambar 5 menunjukkan nilai warna RGB tomat pada berbagai kelas kematangan. Tampak bahwa pada kelas kematangan yang berbeda maka nilai yang dihasilkan juga berbeda. Dari semua citra tomat yang dianalisis, bila ditampilkan dalam bentuk grafik maka dapat dilihat pada Gambar 6. Tampak bahwa nilai R akan semakin besar jika kelas kematangan tomat semakin meningkat. Jika kelas kematangan meningkat yang ditandai dengan semakin meningkatnya warna merah pada buah, maka nilai R juga meningkat. Nilai R merupakan representasi nilai pigmen tomat berupa karoten dan likopen yang mengalami sintesis selama pemasakan, sehingga warna merahnya akan meningkat (Grierson dan Kader, 1986). Nilai G tomat akan semakin menurun dengan meningkatnya kelas kematangan (Gambar 6.). Hal ini terjadi karena nilai G mencerminkan klorofil warna hijau pada tomat yang mengalami degradasi selama pemasakan berlangsung. Sedangkan untuk parameter warna B, maka semakin besar kelas kematangan, maka nilai B akan menurun, dengan penurunan yang besar pada kelas kematangan Green sampai Light red, kemudian akan mendekati konstan sampai dengan akhir pemasakan (kelas Red). Hal ini sejalan dengan yang ditemukan oleh Gautier dkk. (2008) yaitu nilai Blue akan semakin menurun mengarah ke warna Yellow. Akan tetapi, fenomena yang berbeda ditemukan oleh Lana dkk. (2006) yang meneliti bahwa nilai B pada tomat iris akan naik.



Color Parameter	Value
R Mean	105.512
G Mean	133.733
B Mean	3.395

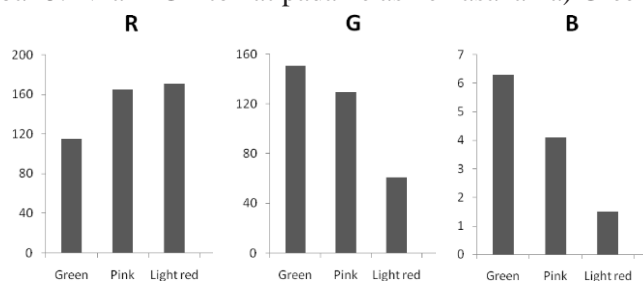
Color Parameter	Value
R Mean	173.007
G Mean	62.036
B Mean	1.472

Color Parameter	Value
R Mean	196.463
G Mean	29.842
B Mean	4.175

B.

C.

Gambar 5. Nilai RGB tomat pada kelas kematangan a) Green, b) Light red, dan c) Red.



Gambar 6. Nilai R, G, dan B tomat pada berbagai kelas kematangan

Penentuan Kualitas Tomat Menggunakan CVS

Untuk menentukan akurasi program dalam menentukan parameter kualitas, maka perlu dilakukan uji akurasi, dengan membandingkan antara nilai yang diperoleh dari program yang dibuat dengan nilai parameter kualitas dari pengukuran secara destruktif (kimiawi) di laboratorium. Gambar 8 merupakan grafik hubungan antara nilai Brix, vitamin C, asam sitrat, dan gula total prediksi dan aktual. Agar diperoleh nilai prediksi yang sama atau mendekati nilai aktual maka dibuat nilai kalibrasi. Untuk nilai Brix, maka nilai aktualnya diperoleh dari persamaan $y = 12,16x - 26,46$ dengan x adalah nilai Brix prediksi. Sedangkan kadar vitamin C, asam sitrat, dan gula total aktual secara berturut-turut diperoleh dari $y = 1,09x - 3,13$, $y = 7,35x - 19,44$, dan $y = 1,58x - 0,18$, dengan x adalah nilai vitamin C prediksi, asam sitrat prediksi, dan gula total prediksi.

B. KESIMPULAN

Sebuah *Computer Vision System* (CVS) yang terdiri dari kotak obyek, webcam, komputer, sistem penerangan, dan perangkat lunak analisis citra dengan jaringan syaraf tiruan dikembangkan untuk menentukan kualitas tomat, yaitu Brix, asam sitrat, vitamin C, dan gula total. Arsitektur jaringan dibentuk dengan 3 lapisan terdiri dari 1 lapisan input dengan 3 sel syaraf masukan, 1 lapisan tersembunyi dengan 14 sel syaraf berfungsi aktivasi logsig dan 5 lapisan output dengan fungsi aktivasi purelin menggunakan algoritma pelatihan *backpropagation*.

DETEKSI MANUSIA MENGGUNAKAN WEBCAM PADA APLIKASI BERBASIS KECERDASAN BUATAN

DI TULIS OLEH :

KANDA IRAWAN

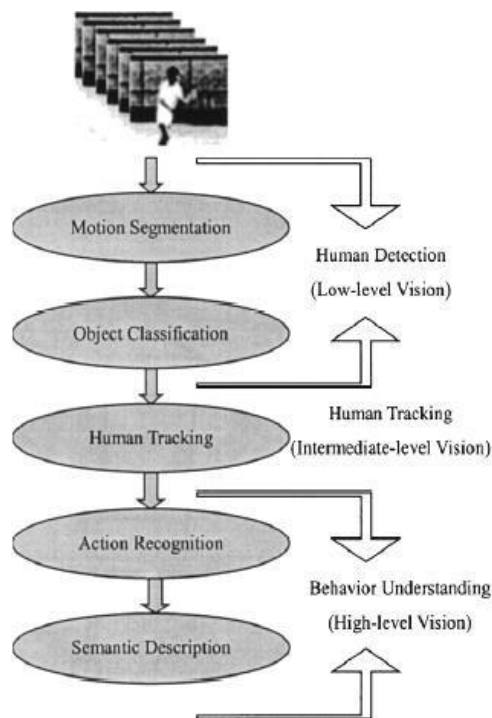
A. ISI

Deteksi gerak (*motion detection*) merupakan komponen penting pada sistem pengawasan ruang (*surveillance*). Hampir setiap vendor kamera atau webcam menyertakan perangkat lunak yang memiliki fasilitas *motion detection* sehingga dapat digunakan untuk memantau suatu area atau ruangan. Jika terdapat suatu objek yang bergerak pada ruangan yang dipantau, maka sistem akan memberikan sinyal berupa suara alarm.

yang timbul pada sistem pengawasan ruang berbasis kamera atau webcam tersebut adalah pendeteksian yang umum sehingga objek apa pun yang terdeteksi pada suatu ruangan yang dipantau akan menyebabkan alarm berbunyi.

Kecerdasan buatan (AI = *Artificial Intelligence*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu sistem atau entitas buatan. Sistem ini umumnya adalah komputer. Kecerdasan diciptakan dan dimasukkan ke dalam komputer yang bertujuan agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

Pendeteksian manusia merupakan dasar ke arah *behavior understanding*. Walaupun bersifat dasar, deteksi manusia merupakan hal yang rumit bila diimplementasikan pada computer.



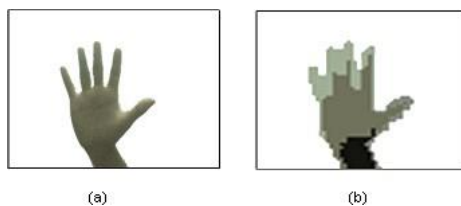
Gambar 1 Tahap analisis perilaku

proses pendeteksian manusia secara garis besar dapat dilakukan dengan melalui tiga tahap yaitu (1) memisahkan *foreground* dan *background* melalui proses segmentasi citra yang berasal dari video atau

webcam; (2) proses *tracking* terhadap pergerakan objek setiap *frame*; (3) klasifikasi terhadap objek yang berhasil dideteksi. Namun ketiga tahap tersebut tidak mutlak harus dilakukan untuk mendeteksi objek karena disesuaikan dengan kebutuhan. Pada laporan tugas akhir ini, penulis hanya menggunakan dua tahap yaitu segmentasi dan klasifikasi.

Mean Shift

Metode segmentasi ini dapat digunakan untuk *feature analysis* yaitu ekstraksi fitur untuk aplikasi pendeteksian dan pengenalan bentuk (*shape detection and recognition*), analisis tekstur citra, dan lain sebagainya. Konsep dasar dari metode ini adalah menggunakan pola vektor dimensi p . Di mana $p = 1$ untuk citra *grayscale* dan $p = 3$ untuk citra berwarna karena tiap *pixel* merupakan kombinasi dari intensitas nilai RGB.



Gambar 2 Segmentasi menggunakan *mean shift*

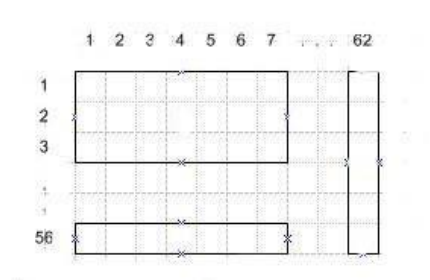
Secara umum sistem deteksi manusia ini memiliki input *real time* dari webcam dan 2 buah pemrosesan yaitu segmentasi dan klasifikasi sehingga menghasilkan output berupa kesimpulan bahwa objek yang terdeteksi adalah manusia.

Analisis JST

Mengingat bahwa manusia bersifat dinamis, maka arsitektur JST yang akan dibangun merupakan tugas yang kompleks. Oleh karena itu diperlukan “mesin penghitung” yang dapat melakukan proses iterasi dengan jumlah neuron bergantung pada percobaan (*trial and error*).

Jumlah Neuron input

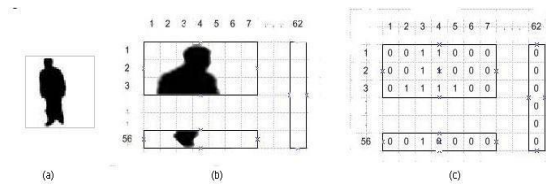
Input JST berasal dari proses segmentasi $62 \times 56 \text{ pixel} = 3472$ atau hasil dari proses yang ditampilkan pada kamera virtual segmentasi.



Gambar 6 Matriks untuk input neuron Jumlah output layer

Lapisan output terdiri dari 3472 buah neuron dengan *pixel* aktif ditentukan sesuai dengan pola inputnya yaitu ukuran pattern $62 \times 56 \text{ pixel}$. Semua neuron yang keluar akan disesuaikan bobotnya dengan pola tubuh manusia pada neuron output. Yaitu untuk menghasilkan klasifikasi “manusia”.

Jumlah lapisan tersembunyi (*hidden layer*) Jumlah hidden layer ditentukan sebanyak 1500 untuk meringankan kinerja prosesor. Misalnya untuk melatih satu citra.



Gambar 7 Visualisasi nilai pola pada matriks

Pada Gambar 7-a merupakan data hasil *thresholding* (citra statis); Gambar 7-b perbesaran pixel, Gambar 7-c nilai pixel *bilevel*.

JST ditentukan oleh 3 hal, yaitu: (1) Arsitektur jaringan; (2) Metode pembelajaran; dan (3) Fungsi aktivasi

Arsitektur Jaringan

Pola hubungan antar neuron pada JST disebut arsitektur jaringan. Secara umum terdapat tiga buah arsitektur JST yaitu: *perceptron (single layer)*, *MLP (multilayer perceptron)* yang memiliki minimal satu buah layer tersembunyi (*hidden*) antara lapisan input dan output serta *recurrent network* yang memiliki *feedback loop*.

Metode pembelajaran

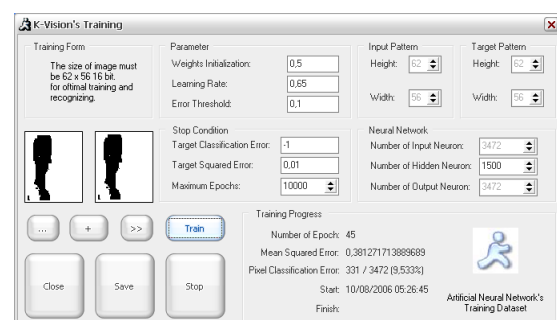
Metode pembelajaran (*learning*) atau pelatihan adalah metode untuk menentukan bobot penghubung (*weight*.) Salah satu metode pembelajaran yang digunakan dalam pengenalan pola atau klasifikasi objek adalah *backpropagation*. Metode *backpropagation* merupakan pelatihan JST dengan supervisi (*supervised*) yaitu terdapat sejumlah pasangan data (masukan – target keluaran) yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan.

Fungsi Aktivasi

Fungsi aktivasi menentukan apakah neuron akan mengalami aktivasi atau tidak. Argumen fungsi aktivasi adalah *net* masukan (kombinasi linier masukan dan bobotnya.) Jika $net = \sum x_i w_i$, maka fungsi aktivasinya adalah $f(net) = f(\sum x_i w_i)$. Salah satu fungsi aktivasi adalah *sigmoid bipolar*.

Implementasi dan Pengujian Sistem

Pola untuk pelatihan JST diambil secara konvensional dari area proses segmentasi pada aplikasi yang *dicapture* kemudian *resize* karena ukuran dari citra yang akan diproses harus sama dengan jumlah panjang dan lebar dari input neuron, yaitu 62 x 56. Gambar 11 merupakan beberapa citra (bilevel) untuk pola pelatihan JST.



Gambar 11 Citra untuk pola pelatihan

Gambar 12 Proses pelatihan jaringan

Pelatihan JST dilakukan menggunakan “mesin penghitung” seperti pada Gambar 12. dengan parameter JST sebagai berikut:

Bobot awal : 0,5

Learning rate : 0,65

Error threshold : 0,1

Epoch : 10000

MSE : 0,01

B. KESIMPULAN

Berdasarkan hasil pengujian sistem dan analisis secara menyeluruh maka dapat ditarik beberapa kesimpulan sebagai berikut:

Posisi terbaik yang dapat dideteksi oleh sistem adalah berjarak maksimal 2 meter dari webcam. Hal ini ditunjukkan pada percobaan kedua, keempat, ketujuh dan kesembilan, sistem dapat mendeteksi manusia dengan waktu maksimal selama 2 menit.

Mengingat bahwa persentase keberhasilan sebesar 40%, sistem pendeteksian manusia ini kurang berhasil dari segi kecepatan oleh karena itu perlu diadakan kajian ulang mengenai jumlah neuron dan pelatihannya serta proses segmentasi yang dilakukan.

Jika jarak objek (manusia) sangat dekat dengan webcam, maka sistem akan menampilkan pesan error yang mengacu pada file HD.dll, ini disebabkan karena rutin pada file *dynamic link library* menggunakan OpenCV yang direkomendasikan untuk prosesor intel. Hal ini dapat dilihat pada hasil pengujian pertama di mana jarak objek (manusia) ke webcam adalah 0,1 meter.

JURNAL 3

MENGHITUNG KECEPATAN MENGGUNAKAN COMPUTER VISION

DI TULIS OLEH :

DANNY AGUS WAHYUDI; IMAN H. KARTOWISASTRO

A. ISI

Computer vision adalah bagian dari ilmu komputer yang membahas bagaimana sebuah komputer dapat ‘melihat’ seperti manusia, oleh karena sangat erat kaitannya dengan penglihatan, pencahayaan menjadi faktor yang juga penting dalam hal ini.

Pencahayaan sangat mempengaruhi kualitas hasil dari *computer vision*, karena seperti yang diketahui bahwa cahaya yang semakin banyak akan membuat kontras gambar semakin naik begitu juga kalau kekurangan cahaya, gambar akan semakin buruk.

Kemampuan di bidang *vision* dapat di kembangkan, contohnya sebuah robot menggunakan *vision* untuk seakan-akan dapat melihat, dan lengan robot di pabrik yang dapat membuat lingkaran presisi atau meletakkan sesuatu. Pengukuran jarak yang terintegrasi juga menggunakan bantuan *vision*. Dari pengembangan *vision* tersebut, penulis tertarik untuk membuat topik ini, yaitu menghitung kecepatan benda dengan bantuan *computer vision*. Studi ini dilakukan untuk mengetahui lebih jauh lagi kemampuan *computer vision*, dan mempelajari bagaimana proses memperkirakan kecepatan sebuah benda yang melaju dari satu titik ke titik lainnya. Topik ini dapat juga di kembangkan lebih lanjut dalam kehidupan sehari-hari seperti dalam memberikan informasi kedatangan kendaraan (estimasi waktu).

Kalibrasi dan Sampling Time

Kalibrasi adalah di mana terjadi penyesuaian ukuran sebenarnya terhadap ukuran yang berada pada gambar, kalibrasi digunakan untuk menghitung kecepatan, karena pada kecepatan digunakan ukuran sebenarnya bukan ukuran pada pixel. *Sampling time* digunakan untuk mengukur berapa lama sistem harus mengambil gambar, pada percobaan ini diberikan 0.5s untuk tiap pengambilan gambar.

Grayscale

Grayscale adalah proses mengubah derajat gambar yang mulanya RGB menjadi keabuan.

Thresholding

Thresholding adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas.

Reverse Pixel

Mengubah nilai pixel dalam gambar, yang bernilai 1 menjadi 0, dan begitu juga sebaliknya.

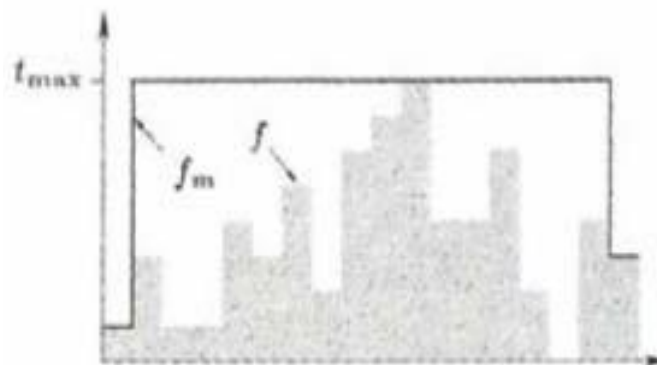
Dengan algoritma: $1\text{-pixel awal} = \text{reverse pixel}$

Removing Pixel

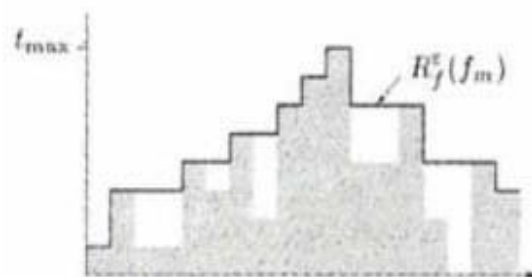
Removing pixel adalah teknik yang digunakan untuk menghilangkan pixel-pixel yang berhubungan dengan jumlah tertentu. Bila ingin menghilangkan pixel dengan jumlah 100, pixel yang berhubungan dengan jumlah 100 akan dihilangkan.

Image Filling Region

Teknik ini digunakan untuk mengisi sebuah daerah yang diinginkan dengan menyamakan jumlah pixel-nya. Proses pengisian daerah (*region*) dilakukan oleh sistem dengan cara mengisi daerah yang ditentukan dengan *pixel* yang sudah ditentukan juga, dengan syarat *pixel* yang diisi tidak melewati sisi luar (*edge*) dari objek tersebut, jadi batas pengisian *pixel* hanya sampai sisi luar objek (*edge*). Gambar 1 di bawah ini adalah bagaimana mengisi/mengubah nilai pixel untuk mengisi sebuah daerah, dengan cara menghilangkannya secara ‘paksa’ atau mengubah set nilai pixel (Gambar 2).



Gambar 1. Cara mengisi/mengubah nilai pixel.



Gambar 2. Cara menghilangkan atau mengubah set nilai pixel.

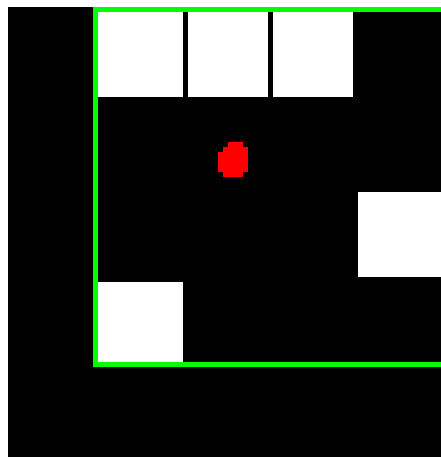
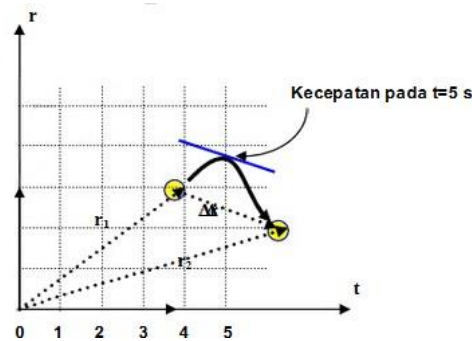
Mencari Centroid

Untuk mendapatkan kecepatan sebuah benda yang bergerak, yang dibutuhkan adalah informasi posisi benda, melalui perubahan posisi benda baru dapat diberikan informasi kecepatannya. Oleh sebab itu, diperlu dicari informasi dari benda tersebut menggunakan algoritma untuk mencari titik tengah dari

benda tersebut. Gambar 3 di bawah ini memperlihatkan bagaimana titik tengah adalah pixel merah, titik tengah di cari berdasarkan daerah yang sudah ditentukan. Pada gambar tersebut, daerah yang ditentukan adalah bounding box yang mempunyai tepi berwarna hijau.

Speed dan Velocity

Speed adalah perubahan jarak terhadap perubahan waktu, sementara *velocity* adalah perubahan posisi terhadap perubahan waktu. *Speed* menggunakan *distance*, sementara *velocity* menggunakan *displacement*. speed satuannya skalar jadi tidak ada informasi tentang arah di dalamnya, sementara velocity satuannya vektor sehingga mempunyai arah. Perhatikan gambar berikut ini.



Gambar 3. Titik tengah dan bounding box.

Karena pada studi ini gambar yang ditangkap dalam bentuk 2D, akan didapatkan informasi dari perubahan posisi tersebut. Untuk menghitung velocity-nya, kita akan menggunakan rumus berikut:

$$\Delta \mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$$

$$\mathbf{v}_r = \frac{\Delta \mathbf{r}}{\Delta t} = \frac{\mathbf{r}_2 - \mathbf{r}_1}{t_2 - t_1}$$

Di mana r berisi informasi x dan y , dan apabila dinyatakan dalam vektor satuan, maka:

$$\mathbf{v}_r = \frac{\Delta x \mathbf{i} + \Delta y \mathbf{j}}{\Delta t} = \frac{\Delta x}{\Delta t} \mathbf{i} + \frac{\Delta y}{\Delta t} \mathbf{j}$$

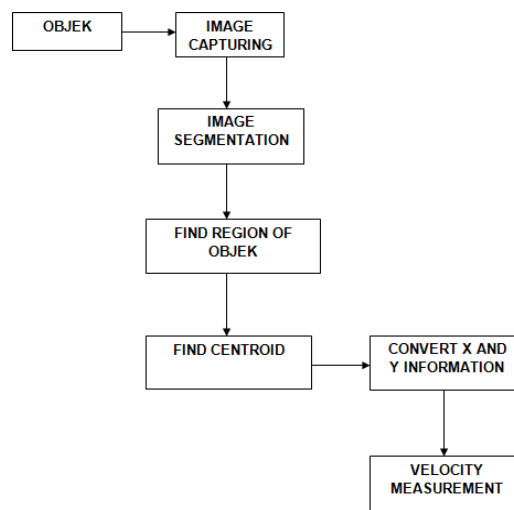
$$\mathbf{v}_r = v_{rx} \mathbf{i} + v_{ry} \mathbf{j}$$

$$|\mathbf{v}_r| = v_r = \sqrt{v_{rx}^2 + v_{ry}^2} \quad \theta = \tan^{-1} \left(\frac{v_{ry}}{v_{rx}} \right)$$

dan besar kecepatan rata-rata dan arah kecepatan rata-rata dapat dihitung dengan rumus:

Blok Diagram Sistem

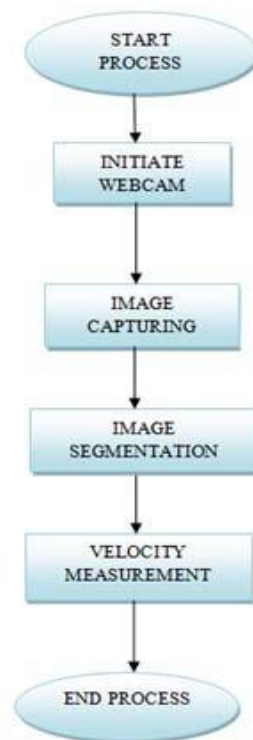
Gambar 4 di bawah ini menunjukkan alur dari sistem secara keseluruhan. Objek yang di tangkap oleh kamera, akan diproses dan sistem akan mengeluarkan informasi dari kecepatan benda tersebut.



Gambar 4. Blok diagram sistem.

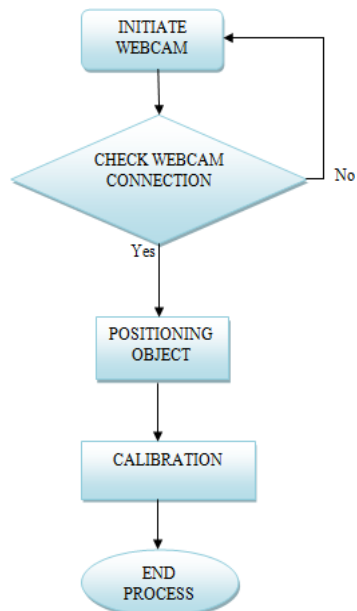
Diagram Alir Sistem

Gambar 5 berikut menunjukkan alir sistem secara keseluruhan.



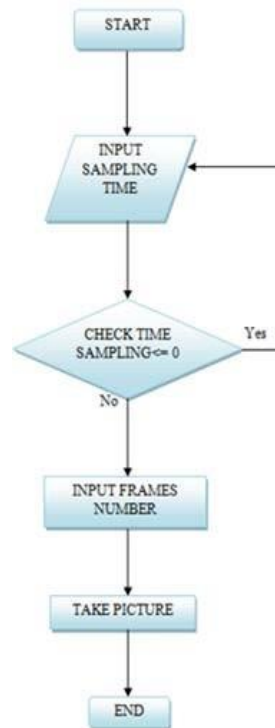
Gambar 5. Diagram alir sistem.

Gambar 6 menunjukkan diagram alir dari inisiasi *webcam* yang di pakai.



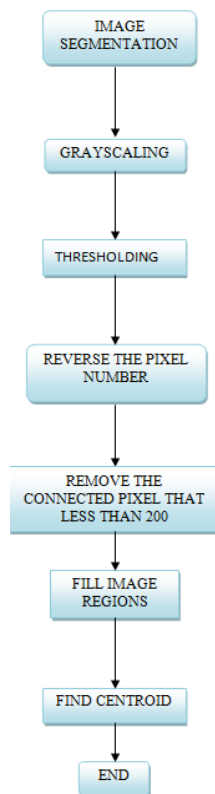
Gambar 6. Diagram alir *initiate webcam*.

Gambar 7 menunjukkan pengambilan gambar yang dilakukan oleh *webcam*. Dengan *interval* waktu yang dimasukkan, sistem akan mengambil gambar selama t second.



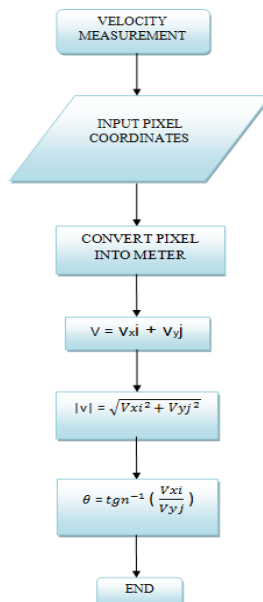
Gambar 7. Diagram alir pengambilan gambar.

Gambar 8 menunjukkan proses segmentasi yang dilakukan oleh sistem, bagaimana sistem memproses image sehingga menghasilkan titik tengah yang nantinya akan di analisis dan menghasilkan informasi kecepatan benda yang di amati.



Gambar 8. Diagram alir segmentasi citra.

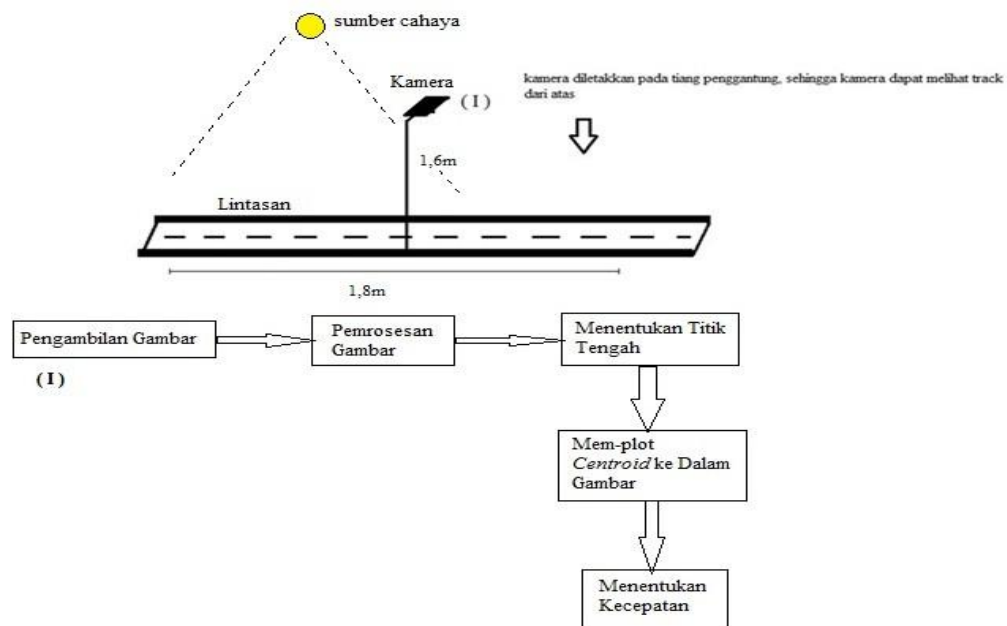
Gambar 9 menunjukkan proses perhitungan kecepatan terhadap informasi yang telah diterima, yaitu titik tengah, dan hasil dari perhitungan akan di tampilkan dan dapat di lihat oleh *user*.



Gambar 9. Diagram alir menghitung kecepatan.

Rancang Bangun

Berikut adalah rancang bangun dari sistem secara keseluruhan (Gambar 10).



Gambar 10. Rancang bangun dari sistem secara keseluruhan.

Spesifikasi Sistem

Spesifikasi sistem yang digunakan yaitu: webcam Logitech c270, *software* Matlab, *regionproperties* untuk mencari *centroid*, sistem operasi Windows 7 pada PC, Tamiya dengan dimensi (14cm x 8cm), track lintasan dengan panjang 1,8m, *background* putih, lampu kamar sebagai sumber pencahayaan dengan spesifikasi lampu: 23watt (2 buah) dan 18watt (1 buah), penyangga dengan tinggi 2,20m, dan kamar dengan dimensi panjang 3,5m dan lebar 2,5m.

HASIL DAN PEMBAHASAN

Implementasi yang dilakukan ditujukan untuk menguji sistem yang telah dirancang, sistem dikatakan stabil apabila sistem dapat mempertahankan output yang stabil dan kinerja sistem yang stabil. Pada perancangan ini sistem diuji dengan pencahayaan yang masuk ke dalam sistem. Berikut adalah hasil percobaan dari sistem (Tabel 1 – 4).

Tabel 1. Hasil Percobaan 1

No.	Lokasi	Pencahayaannya	Sistem – Velocity Measurement		Sistem Manual – (Stopwatch) (m/s)	Kondisi Baterai	Error rate (%)
			Magnitude (m/s)	Sudut Deviasi (°)			
1	Indoor	23 watt	1,09m/s	0,5	1,1m/s	10 – 15 kali pakai	0,9%
2	Indoor	23 watt	1,17m/s	0,8	1,2m/s	10 – 15 kali pakai	2,5%
3	Indoor	23 watt	0,75m/s	0,7	0,74m/s	10 – 20 kali pakai	1,35%
4	Indoor	23 watt	0,66m/s	0,6	0,7m/s	10 – 20 kali pakai	5,7%
5	Indoor	23 watt	0,72m/s	0,6	0,71	10 – 20 kali pakai	1%
6	Indoor	23 watt	0,73m/s	0,6	0,69	10 – 20 kali pakai	4%
7	Indoor	23 watt	0,7m/s	0,6	0,71	10 – 20 kali pakai	1%
8	Indoor	23 watt	0,73m/s	0,5	0,72	10 – 20 kali pakai	1%
9	Indoor	23 watt	0,7m/s	0,5	0,7	10 – 20 kali pakai	0%
10	Indoor	23 watt	0,72m/s	0,6	0,73	10 – 20 kali pakai	1%
Rata – rata			0,797m/s	0,6	0,8m/s		1,935%

Tabel 2. Hasil Percobaan 2

No.	Lokasi	Pencahayaannya	Sistem – Velocity Measurement		Sistem Manual – (Stopwatch) (m/s)	Kondisi Baterai	Error rate (%)
			Magnitude (m/s)	Sudut Deviasi (°)			
1	Indoor	18 watt	2,25m/s	0,3	2,23m/s	1 – 10 kali pakai	0,8%
2	Indoor	18 watt	2,28m/s	0,2	2,3m/s	1 – 10 kali pakai	2%
3	Indoor	18 watt	2,3m/s	0,3	2,3m/s	1 – 10 kali pakai	0%
4	Indoor	18 watt	2,39m/s	0,3	2,36m/s	1 – 10 kali pakai	1,27%
5	Indoor	18 watt	2,33m/s	0,5	2,35m/s	1 – 10 kali pakai	0.8%
6	Indoor	18 watt	2,3m/s	0,4	2,3m/s	1 – 10 kali pakai	0%
7	Indoor	18 watt	2,4m/s	0,1	2,36m/s	1 – 10 kali pakai	1,69%
8	Indoor	18 watt	2,36m/s	0,2	2,3m/s	1 – 10 kali pakai	2,6%
9	Indoor	18 watt	2,29m/s	0,9	2,23m/s	1 – 10 kali pakai	2,6%
10	Indoor	18 watt	2,26m/s	0,5	2,2m/s	1 – 10 kali pakai	2,7%
Rata – rata			2,313m/s	0,37	2,293m/s		1,446%

Tabel 3. Hasil Percobaan 3

No.	Lokasi	Pencahayaannya	Sistem – Velocity Measurement		Sistem Manual – (Stopwatch) (m/s)	Kondisi baterai	Error rate (%)
			Magnitude (m/s)	Sudut Deviasi (°)			
1	Indoor	23 watt +18 watt	1,55m/s	0,3	1,56 m/s	10 - 15 kali pakai	0,6%
2	Indoor	23 watt +18 watt	1,5 m/s	0,2	1,49 m/s	10 - 15 kali pakai	0,6%
3	Indoor	23 watt +18 watt	1,6m/s	0,3	1,58 m/s	10 - 15 kali pakai	1,2%
4	Indoor	23 watt +18 watt	1,3m/s	0,3	1,3 m/s	10 - 15 kali pakai	0%
5	Indoor	23 watt +18 watt	1,4m/s	0,4	1,4m/s	10 - 15 kali pakai	0%
6	Indoor	23 watt +18 watt	1,4m/s	0,5	1.42m/s	1 – 10 kali pakai	1,4%
7	Indoor	23 watt +18 watt	2,4m/s	0,6	2,37m/s	1 – 10 kali pakai	1,26%
8	Indoor	23 watt +18 watt	2,36m/s	0,7	2,35m/s	1 – 10 kali pakai	0,42%
9	Indoor	23 watt +18 watt	2,3m/s	0,7	2,3m/s	1 – 10 kali pakai	0%
10	Indoor	23 watt +18 watt	2,4m/s	0,7	2,34m/s	1 – 10 kali pakai	2,5%
Rata – rata			1,821m/s	0,47	1,81m/s		0,858%

Tabel 4. Hasil Percobaan 4

No.	Lokasi	Pencahayaannya	Sistem – Velocity Measurement		Sistem Manual – (Stopwatch) (m/s)	Kondisi Baterai	Error rate (%)
			Magnitude (m/s)	Sudut Deviasi (°)			
1	Indoor	23 watt + 23 watt + 18 watt	1,63m/s	0,3	1,62 m/s	10 - 15 kali pakai	0,6%
2	Indoor	23 watt + 23 watt + 18 watt	1,6m/s	0,2	1,6 m/s	10 - 15 kali pakai	0%
3	Indoor	23 watt + 23 watt + 18 watt	1,6m/s	0,3	1,59 m/s	10 - 15 kali pakai	0,6%
4	Indoor	23 watt + 23 watt + 18 watt	1,62m/s	0,3	1,64 m/s	10 - 15 kali pakai	1,2%
5	Indoor	23 watt + 23 watt + 18 watt	1,45m/s	0,4	1,48m/s	10 - 15 kali pakai	2%
6	Indoor	23 watt + 23 watt + 18 watt	1,1m/s	0,6	1,15m/s	15 - 20 kali pakai	4,34%
7	Indoor	23 watt + 23 watt + 18 watt	1m/s	0,5	1m/s	15 - 20 kali pakai	0%
8	Indoor	23 watt + 23 watt + 18 watt	1,2m/s	0,6	1,18m/s	15 - 20 kali pakai	1,69%
9	Indoor	23 watt + 23 watt + 18 watt	1,14m/s	0,4	1,2m/s	15 - 20 kali pakai	5%
10	Indoor	23 watt + 23 watt + 18 watt	1,2m/s	0,3	1,2m/s	15 - 20 kali pakai	0%
Rata – rata			1,354m/s	0.39	1.366m/s		1,543%

Pada percobaan di ruang yang tidak terdapat cahaya, sistem tidak dapat mengambil gambar apapun. Pada keadaan ini sistem dianggap tidak dapat bekerja secara maksimal.

Terdapat beberapa poin yang didapatkan pada implementasi ini, yaitu: (1) perhitungan dari sistem dibandingkan dengan perhitungan sistem lain agar dapat membuktikan kestabilan sistem; (2) pencahayaan tidak merata/homogen mengakibatkan adanya *noise*; (3) pada ruangan yang gelap sistem tidak dapat bekerja karena tidak ada cahaya; (4) ruangan yang dipakai mempengaruhi pencahayaan, pemantulan cahaya terhadap latar dinding.

B. KESIMPULAN

Pencahayaan yang tidak merata menyebabkan *noise*, dan tingkat pencahayaan yang tinggi ataupun rendah juga menyebabkan *noise*. *Noise* yang muncul akibat pencahayaan dapat dihilangkan dengan algoritma *remove pixel*. Akan tetapi, saat *noise* memiliki jumlah yang lebih besar dari obyek, sistem tidak dapat mengatasinya lagi, contohnya pada saat kamar gelap. Gambar tidak lagi dapat diproses karena tidak ada obyek apapun yang tertangkap.