

DETEKSI KESALAHAN BENTUK OBJEK BERBASIS PENGOLAHAN CITRA

Object shape error detection based on image processing

Abraham Caesar Yanuar Putra¹, Suci Aulia, S.T., M.T.², Atik Novianti, S.ST., M.T.³

Prodi D3 Teknologi Telekomunikasi, Fakultas Ilmu Terapan, Telkom University

¹bramoktaviandi@student.telkomuniversity.ac.id, ²suciaulia@tass.telkomuniversity.ac.id,
³atiknovianti@tass.telkomuniversity.ac.id

Abstrak

Pada pabrik produksi kue bukan hanya rasa yang nikmat yang dibutuhkan, tetapi bentuk kue yang sempurna juga, akan tetapi terkadang mendapatkan bentuk kue yang tidak sempurna sehingga mengurangi kualitas produksi dari kue tersebut. Permasalahan yang terjadi pada pabrik produksi kue yaitu mendeteksi produk secara manual, membutuhkan banyak pekerja terkadang tidak sesuai dengan harapan dan membuat kualitas produksi jadi menurun sehingga proses produksi kue menjadi terhambat.

Dengan adanya teknologi *morphological image* tersebut maka akan berencana menggabungkan dengan suatu kamera Webcam merupakan sebuah *device* yang dapat digunakan sebagai sensor dalam mendeteksi sebuah benda bergerak melalui proses pengolahan citra. Sehingga bisa mendeteksi kesalahan bentuk kue yang dapat meningkatkan produksi dan mengurangi kesalahan pada penyeleksian manual, pada sistem yang saya rancang menggunakan format gambar .JPG dan .BMP.

Sistem yang akan dibuat dapat mengurangi kesalahan pada penyeleksi manual mendapatkan bentuk yang sempurna sehingga dapat meningkatkan kualitas dari produksi kue tersebut.

Kata Kunci: *Matlab, ImageProcessing, Segmentation, Citra Digital*

Abstract

The cake production plant is not only a delicious taste that is needed, but also the perfect shape of the cake, but sometimes it gets an imperfect cake form which reduces the quality of the cake production. The problems that occur in the cake production plant are manually detecting the product, requiring many workers sometimes not in line with expectations and making the quality of production declining so that the cake production process becomes hampered.

With the *morphological image* technology, it will plan to combine with a Webcam camera, a *device* that can be used as a sensor to detect a moving object through image processing. So that it can detect cookie-shaped errors that can increase production and reduce errors in manual selection, on systems that I designed using the .JPG and .BMP image formats.

The system to be created can reduce errors in manual screening and get the perfect shape so as to improve the quality of the cake production.

Keywords : *Matlab, ImageProcessing, Segmentation, Citra Digital*

1. Pendahuluan

Pada pabrik produksi kue bukan hanya rasa yang nikmat yang dibutuhkan, tetapi bentuk kue yang sempurna juga, akan tetapi terkadang mendapatkan bentuk kue yang tidak sempurna sehingga mengurangi kualitas produksi dari kue tersebut. Permasalahan yang terjadi pada pabrik-pabrik produksi kue yaitu

mendeteksi produk masih menggunakan manual, sehingga pemilihan kue yang baik dan jelek membutuhkan waktu yang lama, membutuhkan banyak pekerja, dan terkadang tidak sesuai dengan harapan sehingga membuat kualitas produksi jadi menurun. Maka dari itu dibutuhkan sebuah alat yang digunakan untuk mendeteksi kesalahan bentuk kue pada pembuatan kue tersebut agar mendapatkan hasil maksimal.

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra, Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer[8]. Dengan adanya teknologi tersebut maka penulis berencana menggabungkan dengan suatu kamera *Webcam* merupakan sebuah *device* yang dapat digunakan sebagai sensor dalam mendeteksi sebuah benda bergerak melalui proses pengolahan citra. *Webcam* juga dapat digunakan dalam pengambilan gambar (*capture image*)[6] sehingga bisa mendeteksi kesalahan bentuk kue yang dapat meningkatkan produksi dan mengurangi kesalahan pada penyeleksian manual. Pada sistem yang saya rancang menggunakan format gambar .JPG dan .BMP karena lebih mudah menampatkan file besar dengan keluaran berkualitas tinggi. Jika mengeskor grafik dengan kualitas tertinggi, kemungkinan besar sedikit aman yang penting ukuran file tidak terlalu besar dan resolusi .jpg 300 dpi [4]. Telah ada penelitian yang membahas tentang pendeteksi kesalahan objek dengan menggunakan RGB yaitu objek tertentu dapat dideteksi dengan menggunakan pengolahan citra digital ini. Salah satu metode yang digunakan adalah berdasarkan segmentasi warna. Normalisasi RGB adalah salah satu metode segmentasi warna yang memiliki kelebihan yaitu mudah, proses cepat dan efektif pada obyek *traffic sign*, maupun aplikasi untuk *face detection*[7] penelitian tersebut hanya menggunakan pendeteksi warna maka di penelitian kali ini ditambahkan dengan pendeteksi bentuk sehingga pemrosesan objek menggunakan 2 tahap warna dan bentuk.

Dengan adanya sistem ini diharapkan dapat dapat mengurangi kesalahan pada penyeleksian manual, hasil pembuatan kue pada pabrik yang bisa mendapatkan bentuk yang sempurna sehingga dapat meningkatkan kualitas dari kue tersebut.

2. Dasar Teori

2.1 Pengertian Citra Digital

Citra dapat berbentuk foto hitam putih atau berwarna, sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetic. Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada ranah waktu atau bidang dan untuk menyatakan nilai keabuan atau warna suatu citra.

Citra adalah gambar bidang dua dimensi. Dan citra memiliki fungsi yang berkesinambungan dari intensitas cahaya pada bidang dua dimensi[2]. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari sumber cahaya. Pantulan dari sumber cahaya ditangkap oleh alat optic, seperti pada kamera, mata manusia, dan lain-lain sehingga bayangan objek dalam citra dapat terekam[1]

2.2 Pengolahan citra (*Image Porcessing*)

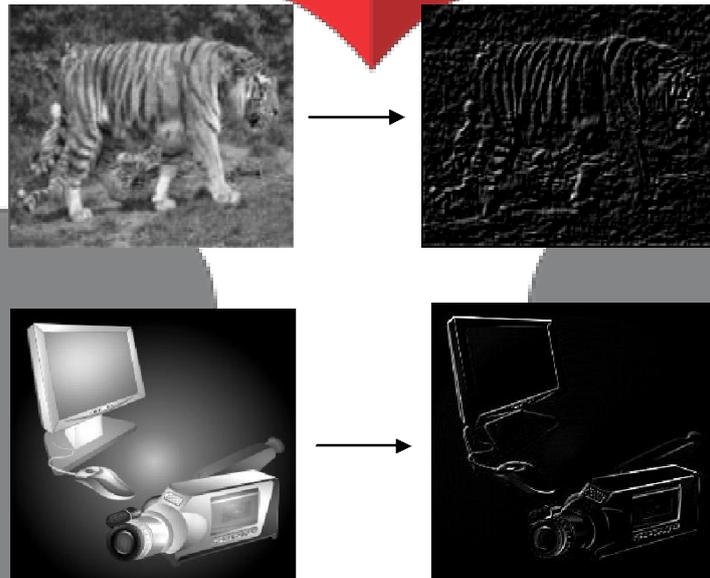
Pengolahan citra atau *Image Processing* adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmuilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*[9].

2.3 Deteksi Tepi

Deteksi tepi merupakan salah satu proses pengolahan yang sering dibutuhkan pada analisis citra yang bertujuan untuk meningkatkan penampakan garis pada citra prosesnya mempunyai sifat diferensiasi atau memperkuat komponen frekuensi tinggi. Tujuan deteksi tepi adalah

1. Untuk menandai bagian yang menjadi detail citra
2. Untuk memperbaiki detail citra yang kabur, yang terjadi kaeran error atau adanya efek dari proses akuisisi citra[5].

Suatu titik (x,y) dikatakan sebagai tepi bila titik tersebut mempunyai perbedaan nilai piksel yang tinggi dengan nilai piksel tetangganya. Gambar 2.3 menunjukkan salah satu model tepi untuk dimensi.



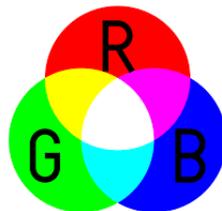
Gambar 2.3 Hasil beberapa deteksi tepi

Pada gambar 2.3 terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi[10].

2.4 RGB (*Red Green Blue*)

Untuk citra berwarna maka digunakan model RGB (*Red-Green-Blue*). Satu citra berwarna dinyatakan sebagai 3 buah matrik grayscale yang berupa matrik untuk *Red* (R-layer), matrik *Green* (G-layer) dan matrik untuk *Blue* (B-layer). R-Layer adalah matrik yang menyatakan derajat kecerahan untuk warna merah (misalkan untuk skala keabuan 0-255, nilai 0 menyatakan gelap (hitam) dan 255 menyatakan merah. G-layer adalah matrik yang menyatakan derajat kecerahan untuk warna hijau, dan B-layer adalah matrik yang menyatakan derajat kecerahan untuk warna biru. Daru definisi tersebut, mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB[9].

Seperti diketahui bahwa tiap citra yang berwarna memiliki nilai indeks warna RGB yang bermacam-macam. Perbedaan prosentase indeks RGB membuat suatu citra menjadi berwarna merah, hijau, biru, kuning dan sebagainya. Semakin tinggi indeks warnanya maka citra tersebut akan semakin terang. Begitu pula sebaliknya, semakin kecil nilai indeks warnanya, maka citra akan semakin gelap[3]



Gambar 2.4 Komposisi Warna RGB

2.4.1 Derajat Keabuan (*Grayscale*)

Dimana proses awal yang sering dilakukan dalam image processing adalah mengubah citra berwarna menjadi citra *grayscale*, proses ini digunakan untuk menyederhanakan model citra. Seperti dijelaskan sebelumnya bahwa citra berwarna terdiri dari tiga layer yaitu *R-layer*, *G-layer*, *B-layer*. Sehingga ketika melakukan proses selanjutnya hanya dapat memperhatikan tiga layer tersebut. Grayscale atau abu-abu pada sebuah image digital adalah image yang pada setiap pixelnya hanya berisikan informasi intensitas warna putih dan hitam. Image Grayscale memiliki banyak variasi nuansa abu-abu sehingga berbeda dengan image hitam-putih. Grayscale juga disebut monokromatik karna tidak memiliki warna lain selain variasi intensitas putih dan hitam. Sebuah image yang dijadikan Grayscale akan terkesan berbeda bila dibandingkan dengan image berwarna



Gambar 2.5 Pantai Warna RGB



Gambar 2.6 Pantai Warna *Grayscale*

2.4.2 BW (*Black And White*)

Sebuah image dengan foto pemandangan atau suatu barang yang berwarna-warni adalah hal yang biasa, tetapi bila image tersebut dijadikan warna hitam-putih dapat terkesan lebih berseni).

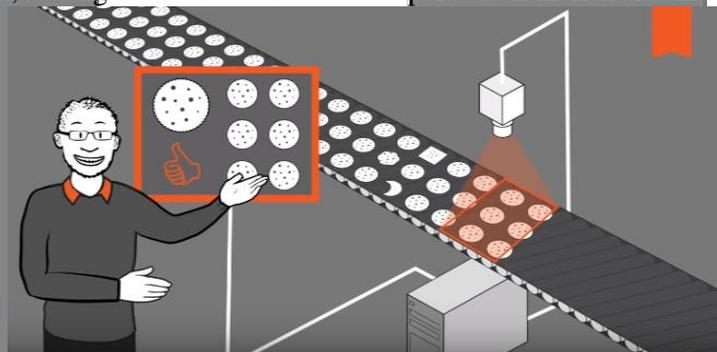


Gambar 2.4.1 Pantai Warna RGB

Gambar 2.7 Pantai Warna *Grayscale*

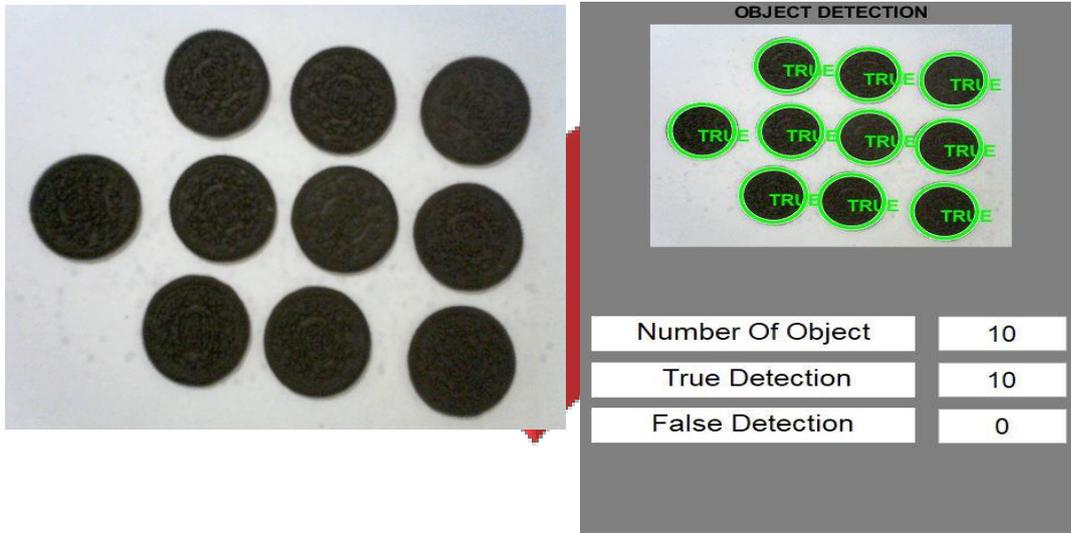
3. Perencanaan

Sistem yang di buat menggunakan *image processing*, *Matlab*, *Kamera webcam*. *Image processing* akan mendeteksi objek dan mengambil gambar objek yang dengan menggunakan kamera yang tersedia. Sistem ini menggunakan kamera webcam Logitech dan menggunakan laptop yang di gabungkan sehingga hasil yang di dapatkan berupa video, 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. Pada saat program di jalankan, maka kamera akan menangkap setiap citra yang ada, setiap citra yang tertangkap kamera akan dilakukan proses analisis citra, yaitu deteksi tepi. Tujuan dilakukan proses deteksi tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek, informasi tentang koordinat garis batas tersebut akan disimpan untuk proses selanjutnya. Proses selanjutnya adalah perbandingan antara citra pertama dan citra selanjutnya. Pada citra yang kedua dilakukan proses yang sama dengan proses yang pertama, yaitu proses deteksi tepi dari proses kedua ini didapatkan informasi tentang koordinat garis batas yang kedua. Koordinat ini akan dibandingkan apabila ada perbedaan maka sistem akan langsung mengaktifkan alarm, sedangkan bila tidak ditemukan perbedaan maka sistem akan tetap merekam.



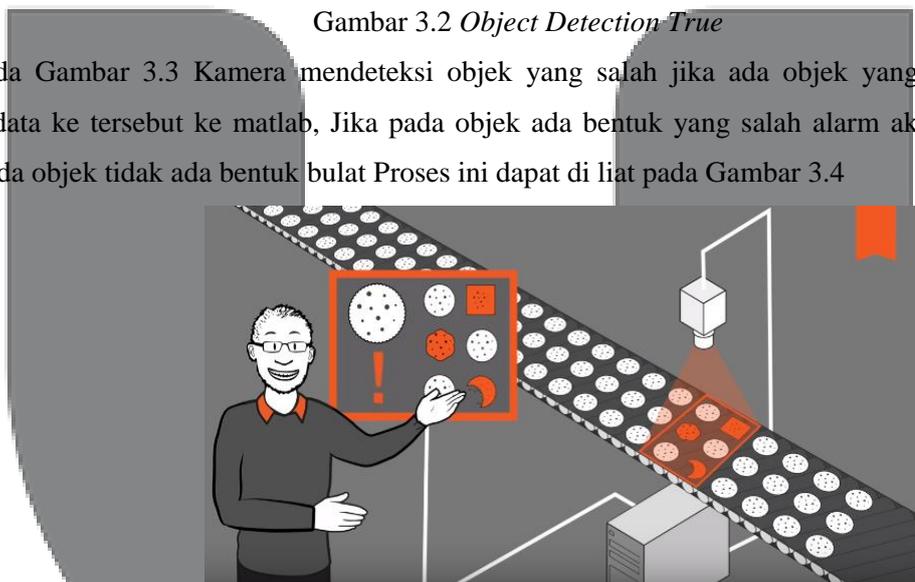
Gambar 3.1 Mendeteksi Objek Tidak Ada Yang Salah

Pada Gambar 3.1 dapat dijelaskan bahwa tujuan system deteksi objek berbasis pengolahan citra adalah untuk membantu mendeteksi kesalahan pada kue yang tidak bulat. Pada gambar 3.1 kamera mendeteksi objek yang benar kemudian hasil capture objek tersebut di save pada folder yang sudah disiapkan. Dari hasil yang capture tadi di dapat objek kue yang masih berupa RGB (*Red Green Blue*) dari hasil RGB dirubah menjadi BW (*Black and White*). Proses ini dapat di lihat pada Gambar 3.2 dan Gambar 3.3

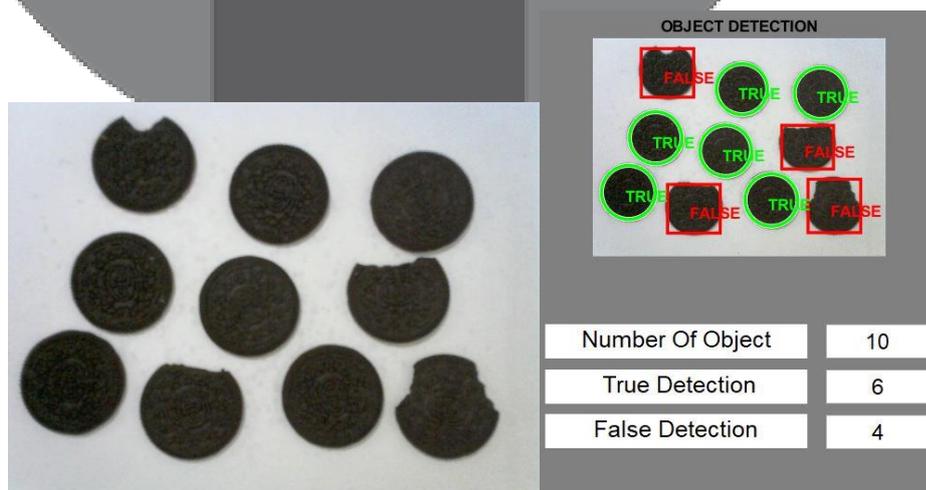


Gambar 3.2 Object Detection True

Pada Gambar 3.3 Kamera mendeteksi objek yang salah jika ada objek yang salah dan langsung mengirim data ke tersebut ke matlab, Jika pada objek ada bentuk yang salah alarm akan menyala/warning, dan jika pada objek tidak ada bentuk bulat Proses ini dapat di liat pada Gambar 3.4



Gambar 3.3 Mendeteksi Objek Ada Yang Salah



Gambar 3.4 Object Detection False

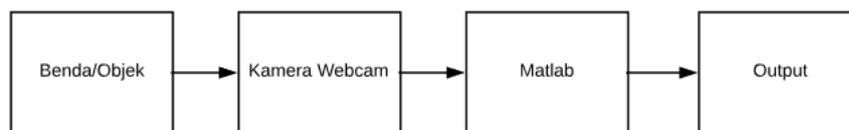
3.1 Deskripsi Proyek Akhir

Proyek Akhir ini dilakukan dengan cara gambar diambil menggunakan webcam dengan jarak objek dengan kamera 20-30 cm, lalu gambar tersebut dikirim ke laptop yang sudah terintegraris oleh matlab, kemudian pada proses di matlab dilakukan proses perubahan warna objek dari RGB, putih abu-abu, BW. Pada proses RGB ke putih abu-abu, objek yang di tangkap dari warna asli dirubah menjadi putih abu-abu kemudian dari warna putih-abu diubah menjadi hitam putih pada proses (BW) berfungsi untuk memperhalus gambar agar gambar yang keditek oleh kamera ke matlab hanya objek yang di ujcobakan,

jika gambar tersebut sesuai pencocokan akan ditampilkan di GUI sedangkan jika pada saat pencocokan gambar tidak sesuai alarm akan menyala dan akan di tampilkan kesalahan objeknya itu menandakan objek tersebut tidak sesuai

3.2 Blok Diagram Koneksi Kamera Webcam ke Matlab

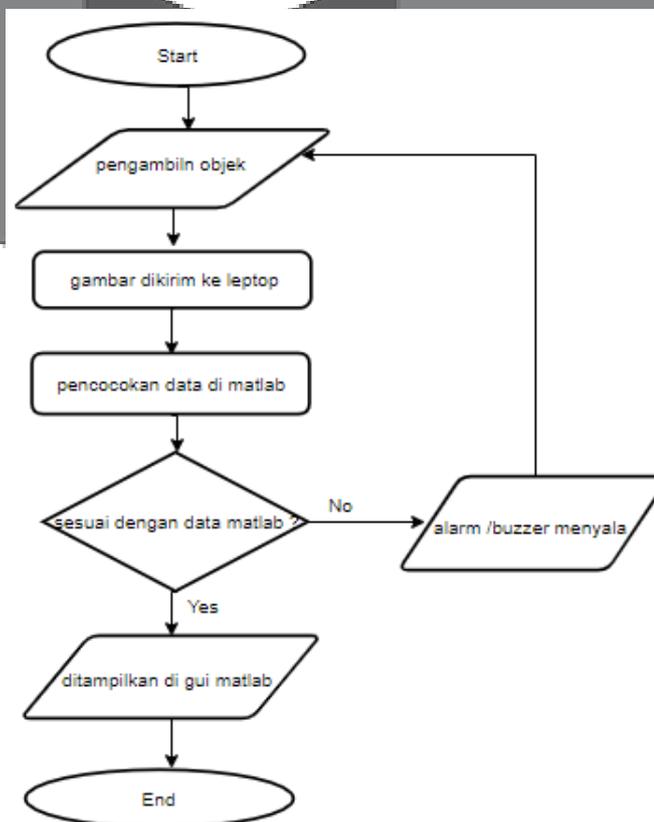
Pada bab ini akan dijelaskan mengenai perancangan sistem monitoring pada objek berbasis *image processing* dengan menggunakan *Matrix Laboratory (MATLAB)* dan *Alarm*. Sistem yang akan dibuat merupakan sistem yang mampu mendeteksi gerakan berdasarkan citra yang tertangkap oleh kamera *webcame*. Melalui sistem yang akan dibuat setiap objek/ benda yang bergerak akan di monitoring oleh kamera webcam jika objek/benda yang di monitoring tidak bulat buzzer/alarm akan menyala dan jika pada objek berbentuk bulat buzzer tidak akan menyala



Gambar 3.5 Blok Diagram Koneksi Kamera Webcam ke Matlab

3.3 Flowchart System Proses Deteksi Objek

Berikut adalah flowchart perancangan system yang menjelaskan proses pada perancangan system. Data yang diproses adalah data hasil objek yang di capture.



Gambar 3.6 Flowchart Perancangan System Proses Deteksi Kamera Matlab

Pada Gambar 3.6 menjelaskan urutan proses pada flowchart perancangan system dimulai dengan kamera start/realtime. Setelah kamera realtime, kamera mengambil objek atau mengcapture objek yang di monitoring, kemudian hasil capture di kirim ke laptop dengan cara save pada folder yang sudah dibuat. Sampai proses data yang di save akan di proses pencocokan data pada matlab data masih berupa *RGB (Read Green Blue)* diubah *Greyscale (menjadi putih abu-abu)* diubah lagi menjadi *BW (Black and White)*. Proses selanjutnya deteksi objek berbentuk lingkara tidak lingkaran, jika objek yang deteksi berbentuk lingkaran akan ditampilkan di GUI dan jika objek yang deteksi tidak berbentuk lingkaran akan di tampilkan di GUI dan alarm menyala

3.4 Tahapan Rancangan.

3.4.1 Akuisisi Citra

Merupakan proses menangkap (*capture*) suatu citra analog sehingga diperoleh citra digital. Beberapa factor yang memperhatikan dalam proses akuisis citra antara lain adalah: jenis alat akuisis, resolusi kamera, Teknik pencahayaan, perbesaran atau zooming, jarak, dan sudut penambilan citra .

3.4.2 Akuisisi Capture dan Save

Capture adalah suatu gambar yang diambil oleh komputer untuk merekam tampilan yang tampak di layar atau peranti keluaran visual lainnya. Biasanya ini adalah suatu gambar digital yang ditangkap oleh sistem operasi inang atau perangkat lunak yang dijalankan pada komputer, walaupun dapat pula dihasilkan oleh kamera atau peranti yang menangkap keluaran video dari komputer. Cuplikan layar dapat digunakan untuk mendemonstrasikan suatu program, suatu masalah yang dihadapi, atau secara umum sewaktu keluaran komputer perlu ditunjukkan pada orang lain atau diarsipkan.

Save adalah penyimpanan data yang di lakukan pada file (current document) sedangkan save as adalah menyimpan yang dilakukan pada tempat nama dan tipe data yang ditentukan

3.4.2 Preprocessing

Pada tahap ini data citra masukan akan diubah menjadi data citra yang lebih sesuai untuk diproses. Preprocessing yang dilakukan meliputi beberapa tahap mulai dari *input* data citra, konversi RGB ke greyscale ke *BW (black and white)*, *resize* dan *cropping* menghasilkan citra sesuai dengan standar yang diharapkan dan dapat memudahkan proses selanjutnya.

1. Pengubah warna

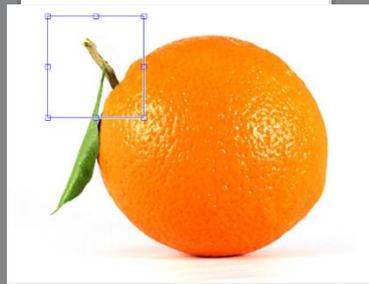
Pengubah warna adalah mengubah warna citra yang awalnya citra RGB diubah menjadi citra *grayscale*, setelah menjadi *grayscale* akan diubah menjadi citra *BW*



Gambar 3.7 RGB GRSCALE BW

1. Cropping

Cropping adalah proses penghapusan bagian sudut dari suatu gambar untuk memotong/mengambil/mengeluarkan sebagian isi dari gambar guna memperoleh hasil yang diinginkan.



Gambar 3.8 Cropping

2. Labeling

Labeling adalah proses memberikan label objek yang ada di dalam citra untuk memisahkan masing-masing objek yang di deteksi. Sebelum memberikan label, sebelum memberikan label terlebih dahulu menentukan jumlah objek yang ingin di deteksi.

4. Analisis

Simulasi Deteksi Objek ini terbagi atas dua skenario, pengujian cahaya serta pengujian parameter jarak. Skenario 1 pengujian cahaya dilakukan dengan deteksi objek dengan cahaya yang terang dan kurang terang, dan skenario 2 pengujian parameter jarak dilakukan dengan melakukan deteksi objek dengan jarak yang berbeda-beda dan melakukan analisis tingkat akurasi setiap jarak. Hasil simulasi dari kedua skenario ini akan dibandingkan dan dianalisis.

4.1 Hasil Analisis Pengujian Cahaya

Berdasarkan table 4.1 hasil analisis pengujian didapatkan 15 kali percobaan, jarak yang dipakai untuk mendeteksi objek 20cm, 25cm, 30cm, 35cm, 40cm, dengan lux 32, lux 58, dan lux 325. Pengaruh cahaya terhadap objek yang di deteksi tidak tapi berpengaruh, tetapi jika lux cahaya kurang dari 20lux kamera akan gelap untuk deteksi objek dan jika cahaya terlalu terang atau terlalu dekat dengan objek akan menimbulkan bayangan objek yang di deteksi.

Tabel 4. 1 Hasil analisis Pengujian Cahaya

No	Jarak Kamera	Cahaya		
		Lux 32	Lux 58	Lux 325
1	20cm	Baik	Baik	Baik
2	25cm	Baik	Baik	Baik
3	30cm	Baik	Baik	Baik
4	35cm	Baik	Baik	Baik
5	40cm	Baik	Baik	Baik

4.2 Hasil Analisis Pengujian Parameter Jarak

Berdasarkan table 4.2 hasil analisis pengujian didapatkan 15 kali percobaan, dari 15 kali percobaan objek maksimal deteksi hanya 6 objek, jarak yang dipakai untuk mendeteksi objek 20cm, 25cm, 30cm, 35cm, 40cm, dalam pengujian dampak bayangan cahaya objek tidak tapi berpengaruh proses perubahan warna dari dari RGB (*Read Green Blue*) ke Putih abu-abu (*Gryscale*), BW (*Black and White*), pada proses Putih abu-abu ke BW akan dampak bayangan objek akan mempengaruhi perubahannya.

Dilihat dari table 4. 2 jarak 20cm akurasi deteksi jumlah area objek lumayan bagus karena semakin banyak objek akurasi lumayan bagus, Jarak 25cm akurasi deteksi jumlah area objek bagus karena semakin banyak objek yang di deteksi jumlah area yang di deteksi akurasi yang di dapatkan bagus, jarak 30cm akurasi deteksi jumlah area objek kurang bagus karena semakin banyak objek akurasi deteksi kurang bagus, jarak 35cm akurasi deteksi jumlah area objek kurang bagus, karena semakin banyak deteksi objek akurasi kurang bagus, jarak 40cm akurasi deteksi jumlah area objek jelek karena jarak objek antara kamera jauh, semakin banyak objek yang di deteksi akurasi jelek, jarak 45cm akurasi deteksi jumlah area objek sangat sangat jelek karena jarak objek dan kamera terlalu jauh, semakin banyak objek yang di deteksi sangat jelek. Setelah dilakukan pengujian terhadap jarak 20cm, 25cm, 30cm, 35cm, 35cm, 40cm, dan 45cm dengan masing-masing pengujian sebanyak 15 kali didapatkan hasil terbaik adalah di jarak 25cm karena objek yang di dapatkan dari hasil pengujian di jarak

Tabel 4. 2 Hasil Analisis Pengujian Parameter Jarak

No	Jarak Kamera	Jumlah Objek	False	True	Jumlah Area 1	Jumlah Area 2	Jumlah Area 3	Jumlah Area 4	Jumlah Area 5	Jumlah Area 6
1	20cm	2	1	1	55684	61222				
2	20cm	4	2	2	59628	61299	54665	60193		
3	20cm	6	3	3	60459	62917	55178	50270	52817	43029
4	25cm	2	1	1	38515	41807				
5	25cm	4	2	2	33510	37221	42403	413646		
6	25cm	6	3	3	31659	38842	33295	39828	40861	43064
7	30cm	2	1	1	27614	29264				
8	30cm	4	2	2	27298	28095	22788	25919		
9	30cm	6	3	3	21298	27128	22336	27015	26833	28549
10	35cm	2	1	1	19044	17995				
11	35cm	4	2	2	13696	13585	11805	13772		
12	35cm	6	3	3	10659	13774	13620	13512	11744	13697
13	40cm	2	1	1	12987	14041				
14	40cm	4	2	2	13696	13585	11805	13772		
15	40cm	6	3	3	10659	13774	13620	13512	11744	13697

5. Penutup

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan maka didapatkan kesimpulan yaitu:

- a. Sistem melakukan *capture* secara manual pada system yang dibuat, hasil citra yang sudah di *capture* kemudian di proses software matlab, di software matlab memproses hasil citra yang di dapat dengan merubah warna dari *rgb-gayscale-bw* kemudian ditampilkan *false and true* di *gui* yang sudah dibuat.
- b. Pengaruh cahaya tidak tapi berpengaruh, tetapi jika cahaya 20lux kamera gelap dan tidak bisa mendeteksi objek dan jika cahaya terlalu terang akan menimbulkan bayangan, bayangan objek juga mempengaruhi dampak diteksi objek karena itu di perlukan cahaya 58 Lux yang terang agar cahaya bayangan objek tidak terlihat.
- c. Pengaruh jarak kamera terhadap objek adalah jika objek semakin dekat dengan kamera maka akan mendapatkan nilai luas area yang besar dan sebaliknya jika jarak kamera terhadap objek semakin jauh maka nilai luas area yang didapatkan semakin kecil. Penguji menggunakan jarak pengujian 20cm, 25cm, 30cm, 35cm, 40cm, 45cm. Jarak 25cm adalah jarak terbaik dimana dapat mengambil jumlah objek yang lumayan banyak dan analisis sesuai yang diharapkan.
- d. Pengaruh *noise* atau benda-benda kecil juga berpengaruh pada saat deteksi objek, karena pada saat perubahan warna putih abu-abu (*Gayscale*) ke *BW* (*Black and White*) di deteksi otomatis.

5.2 Saran

Pada proyek akhir ini terdapat kekurangan pada aplikasi sehingga dapat dilakukan pengembangan untuk proyek akhir ini :

- a. Untuk diteksi objek selanjutnya disarankan bisa menditeksi kue kotak
- b. Jarak diteksi objek selanjutnya disarankan 15 cm dan 45cm

Daftar Pustaka

- [1]. Adisty, R., & Muslim, M. A. (2016). *Deteksi dan Klasifikasi Kendaraan menggunakan Algoritma Backpropagation dan Sobel*, 1(2), 65–73 Universitas Brawijaya, Malang, Indonesia.
- [2]. Ali, J. (2016). *SISTEM SECURITY WEBCAM DENGAN MENGGUNAKAN MICROSOFT VISUAL*, 1(2), 46–58 Panam- Pekanbaru.
- [3]. Bustomi, M. A., & Dzulfikar, Z. (2014). *Analisis Distribusi Intensitas RGB Citra Digital untuk Klasifikasi Kualitas Biji Jagung menggunakan Jaringan Syaraf Tiruan. Fisika Dan Aplikasinya*, 10(3), 127–132. https://doi.org/10.1007/978-3-319-40216-1_2 Institut Teknologi Sepuluh Nopember (ITS), Jl. Arief Rahman Hakim, Surabaya.
- [4]. Divya, M. (2015). *International Journal of Emerging Trends in Science and Technology*, 3285–3290 University of Newcastle United Kingdom, England.
- [5]. Ervika Pramu Shinta, Imam Santoso, R. R. I. (2015). *Aplikasi Webcam Untuk Mendeteksi Gerakan Suatu Objek*, 1–7 Universitas Diponegoro Fakultas Teknik Jurusan Teknik Elektro Semarang.
- [6]. Asep Nana, M. Ichwan, I Made Santika Putra (2015) *Informatika*, J. T., Industri, F. T., Vision, C., & Warna, D. (n.d.). *Segmentasi citra untuk deteksi objek warna pada aplikasi pengambilan bentuk citra rectangle*, 1–10 Jurusan Teknik Informatika, Fakultas Teknologi Industri Institut Teknologi Nasional Bandung.
- [7]. Kusumanto, R. D., & Tomponu, A. N. (2011). *PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB*, 2011(Semantik) Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS), Jl. Arief Rahman Hakim, Surabaya.
- [8]. Kusumanto, R. D., & Tomponu, A. N. (2016). *PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN NORMALISASI RGB*, (January). Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang 30139 Jl Sriwijaya Negara Bukit Besar, Palembang.
- [9]. Mulyawan, H., Samsono, M. Z. H., & Setiawardhana. (2011). *Identifikasi dan Tracking Objek Berbasis Image Processing Secara Real Time*. Jurusan Telekomunikasi - Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember (ITS) Surabaya Kampus PENS-ITS, Keputih, Sukolilo, Surabaya
- [10]. SAA Bowo, A Hidayatno, RR Isnanto, R. R. I. (2011). *Analisis deteksi tepi untuk mengidentifikasi pola daun*. Undergraduate Thesis, Diponegoro University, (June), 1–7 Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro, Semarang

