

IMPLEMENTASI *IMAGE PROCESSING* PADA PROSES PENGISIAN AIR TEH

Prasetyo Adi Nugroho¹

¹Dosen Teknik Informatika, Universitas Darma Persada

Abstrak

Implementasi Image processing sudah banyak ragamnya, misalnya absensi sidik jari, robot vision, pemantauan cuaca berdasarkan warna dan lain-lain. Dalam tulisan ini akan dijelaskan salah satu implementasi Image Processing pada proses pengisian air teh ke dalam botol. Input yang dimasukkan berupa citra real time (video) dalam format 24 bit ukuran 320 x 240 piksel. Citra kemudian diperkecil ukurannya menjadi 160 x 120 piksel dengan tujuan untuk mempercepat proses Image Processing dan kemudian diubah menjadi citra grayscale (8 bit).

Citra kemudian di filter dengan filter median untuk mereduksi noise dan dibuat menjadi citra biner. Deteksi tepi selanjutnya dikenakan pada citra dan hasil pendeteksian tepi digunakan untuk memberikan perintah kepada mikrokontroler AT89S51 melalui port serial (RS 232). Selanjutnya mikrokontroler akan mengontrol kerja dari motor pengisi air teh.

Kata Kunci : *Image Processing, mikrokontroler AT89S51, Port serial (RS 232).*

I. PENDAHULUAN

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi [4].

Citra digital dapat diolah dengan komputer karena berbentuk data numeris. Analisis citra digital (*digital image analysis*) menghasilkan suatu keputusan atau suatu data, termasuk didalamnya adalah pengenalan pola (*pattern recognition*).[1]

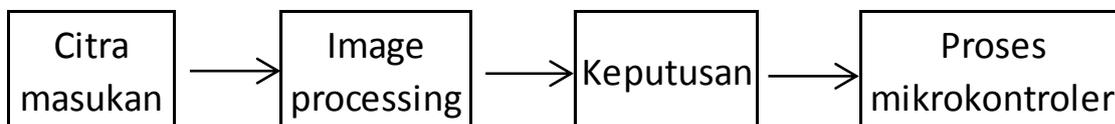
Implementasi dari *image processing* salah satunya dapat diterapkan pada proses pengisian air teh ke dalam botol. Citra yang diolah adalah citra *real time* (Video) yang diambil menggunakan webcam. Jumlah frame citra yang dihasilkan oleh webcam 30 fps (*frame per second*), tetapi untuk mempercepat proses pengolahan citra maka frame yang diolah adalah 1 fps.

Citra keluaran dari webcam adalah citra *true color* (warna) format 24 bit yang selanjutnya diubah ke dalam format bitmap dan diperkecil, kemudian dikonversi ke format 8 bit (*grayscale*). Untuk mereduksi *noise* maka proses *filtering* dikenakan pada citra yang dilanjutkan dengan proses *binerisasi* citra. Proses pendeteksian tepi kemudian dilakukan pada citra hasil *binerisasi*. Hasil deteksi tepi ini selanjutnya digunakan untuk memberikan keputusan ke mikrokontroler AT89S51 melalui port serial (RS 232). Mikrokontroler kemudian mengontrol kinerja motor pengisi air teh berdasarkan data yang diterima dari komputer.

II. PERANCANGAN SISTEM

Terdapat empat proses utama dari sistem yang dibuat, yaitu proses *capturing* botol teh yang diisi menggunakan webcam dan digunakan sebagai citra masukan, kemudian *image processing*, keputusan, proses mikrokontroler.

Yang harus diperhatikan dalam proses *image processing* adalah ukuran citra yang diolah dan tingkat pencahayaan pada waktu proses *capturing*. Ukuran citra sangat berpengaruh pada kecepatan pemrosesan data dan tingkat pencahayaan sangat menentukan hasil pembacaan tepi objek. Keluaran dari proses *image processing* digunakan untuk pengambilan keputusan, keputusan kemudian dikirim melalui port serial (RS 232) ke mikrokontroler AT89S51 yang selanjutnya digunakan untuk mengontrol motor pengisi.



Gambar 1. Blok Diagram

2.1. Citra Masukan

Citra yang digunakan sebagai masukan (*input*) didapat dari sebuah webcam dengan ukuran 320x240 pixel 24 bit. Agar citra bisa diolah, maka citra dikonversi kedalam format BMP. Obyek diambil dengan penerangan yang cukup dengan posisi webcam menghadap lurus ke depan (*frontal view*) dan tidak tertutupi atau terhalangi oleh obyek-obyek lainnya.

2.2. Image Processing (pengolahan citra)

Image Processing yang digunakan pada tulisan ini dibagi menjadi beberapa bagian yaitu proses penskalaan citra masukan, proses konversi dari *image* ke matrik (*array*), proses *greyscale*, reduksi *noise*, pencarian nilai *threshold* yang sesuai dengan citra masukan, proses *thresholding* dan deteksi tepi (*edge detection*). Penskalaan citra dilakukan dengan membentuk citra baru yang mempunyai ukuran setengah lebih kecil dari citra masukan. Proses *Image2Array* dilakukan dengan cara *scanning* tiap piksel citra dan hasilnya disimpan dalam bentuk matrik, seperti yang dilakukan [3]. Pada proses *grayscale*, citra masukan yang sudah diskala dijadikan citra dengan piksel format 8 bit yang tiap piksel memiliki kemungkinan warna maksimal 255. Proses *grayscale* dilakukan dengan membaca nilai RGB dari tiap piksel citra dan kemudian dirata-ratakan seperti yang dilakukan [3].

$$\text{Grayscale} = \frac{R + G + B}{3} \dots\dots\dots(1)$$

atau

$$\text{Grayscale} = (R*0.24)+(G*0.32)+(B*0.28)\dots\dots\dots(2)$$

Proses reduksi *noise* dilakukan dengan metode statistik median. Nilai-nilai keabuan dari titik-titik didalam citra diurutkan dari urutan terkecil sampai yang terbesar kemudian dicari mediannya. Nilai median merupakan nilai yang terletak paling tengah. Operasi median dapat menggunakan *mask / templete* tanpa bobot dengan ukuran yang dikehendaki, misalnya 3x3, 5x5, 7x7 atau bahkan bisa juga menggunakan yang tidak berbentuk bujur sangkar seperti yang dilakukan [1]. Perlu diingat dalam penentuan ukuran *mask* yang digunakan sangat berpengaruh pada ketajaman citra yang dihasilkan dan kecepatan proses *Image Processing*-nya.

Citra *output* dari proses reduksi selanjutnya menjadi *input* dari proses selanjutnya, yaitu pencarian nilai *threshold*. Nilai *threshold* awal dicari dengan rerata intensitas tiap piksel. Nilai awal ini digunakan sebagai patokan untuk mencari nilai *threshold* yang akan digunakan. Proses pencarian nilai *threshold* berdasarkan citra masukannya dilakukan dengan metode *iterasi* seperti yang dilakukan [2].

Dengan menggunakan nilai *threshold* dari metode iterasi selanjutnya dilakukan proses binerisasi (*thresholding*) pada citra dengan cara membagi citra menjadi dua bagian.

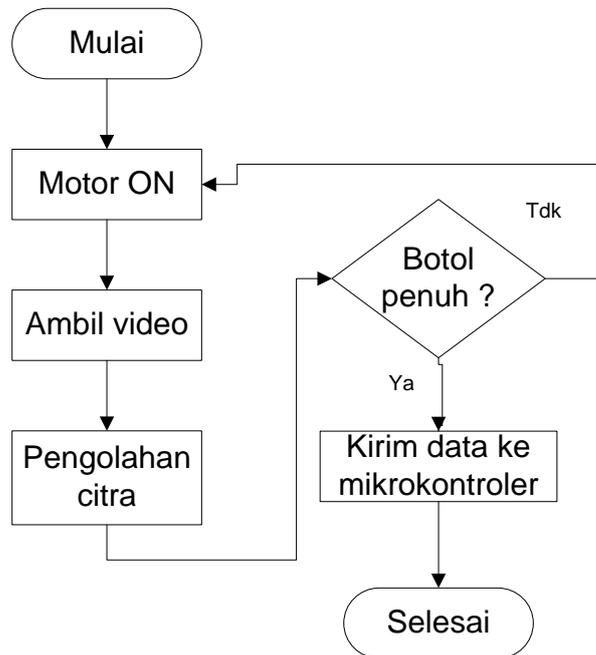
$$h_o(x,y) = \begin{cases} 1, & \text{jika } h_i(x,y) \geq T \text{ iterasi} \\ 0, & \text{kebalikannya} \end{cases} \dots\dots\dots(3)$$

dimana : h_o adalah citra hasil,
 h_i adalah citra *input* hasil reduksi *noise*,
 T_{iterasi} adalah nilai *threshold* iterasi

Proses selanjutnya adalah menemukan tepi objek hasil proses *thresholding*, operator yang digunakan adalah operator Robert karena tepi yang diperoleh paling tipis jika dibandingkan dengan operator yang lain. Hasil deteksi tepi biasanya berupa nilai yang kecil, sehingga cenderung untuk menghasilkan daerah yang berwarna hitam sebagai latar belakang dan garis yang terbentuk berwarna putih. Untuk membalik hasilnya, maka dilakukan operasi *negasi* (pembalikan) dengan cara mengurangkan hasil deteksi tepi dengan 255.

2.3. Keputusan

Hasil deteksi tepi selanjutnya digunakan untuk mengambil keputusan. Keputusan diambil dengan cara *scanning* bagian tengah citra, dimulai dari citra bagian atas. *Scanning* akan berhenti sampai menemukan ada salah satu piksel citra pada kolom yang sudah ditentukan yang bernilai 255 (warna hitam). Posisi piksel yang berwarna hitam selanjutnya dibandingkan dengan tinggi citra (160 x 120 piksel), jika posisi piksel ≤ 18 , maka komputer akan mengirim karakter "A" ke mikrokontroler, jika posisi piksel ≤ 94 dan ≥ 19 maka komputer akan mengirim karakter "A" ke mikrokontroler, jika posisi piksel ≥ 95 maka komputer akan mengirim karakter "B" ke mikrokontroler.

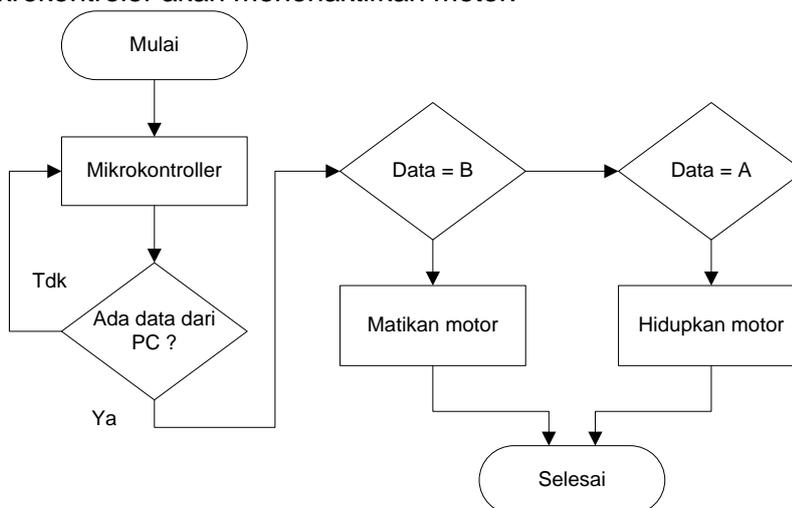


Gambar 2. Flow process system secara umum

2.4 Proses Mikrokontroler

Proses awal yang dilakukan oleh mikrokontroler adalah melakukan sinkronisasi kecepatan pengiriman dan penerimaan data dengan komputer. Setelah itu dilakukan proses *checking register SBUF* untuk mengetahui apakah ada data yang diterima oleh mikrokontroler.

Jika ada data yang diterima maka data akan di *compared* dengan data yang sudah dimasukkan dalam program. Jika data yang diterima adalah *character "A"*, maka mikrokontroler akan menghidupkan motor pengisi, tetapi jika data yang diterima adalah "B", maka mikrokontroler akan menonaktifkan motor.



Gambar 3. Flow process mikrokontroler

III. HASIL DAN PEMBAHASAN

Eksperimen dilakukan sebanyak 10 kali dengan intensitas pencahayaan yang berbeda-beda dan objek percobaan tidak terhalangi oleh benda lain. Citra *real time* diambil menggunakan webcam merk Logitech C170, sedangkan intensitas pencahayaan diukur menggunakan KRISBOW KW06-291 4 in 1 Environment Meter.

Berdasarkan hasil pengujian yang dilakukan terhadap beberapa citra yang dijadikan bahan uji, didapat data sebagai berikut :

Tabel 1. Hasil Pengujian

No	Intensitas Cahaya (Lux)	Keterangan
1	5	Perintah tidak sesuai
2	10	Perintah tidak sesuai
3	20	Perintah sesuai yang diharapkan
4	35	Perintah sesuai yang diharapkan
5	50	Perintah sesuai yang diharapkan
6	60	Perintah sesuai yang diharapkan
7	100	Perintah sesuai yang diharapkan
8	250	Perintah sesuai yang diharapkan
9	500	Perintah sesuai yang diharapkan
10	600	Perintah sesuai yang diharapkan

Dari hasil pengujian dengan menggunakan rancangan sistem ini, didapatkan hasil 8 citra dapat menghasilkan perintah yang benar dan 2 citra tidak dapat memberikan perintah dengan benar (tingkat akurasi 80%), hal ini disebabkan karena :

1. Tingkat pencahayaan yang agak gelap.
2. Posisi peletakan kamera.

IV. KESIMPULAN

Dari hasil eksperimen yang dilakukan dengan menggunakan model rancangan sistem ini, dapat disimpulkan bahwa :

1. Beberapa tahapan pengolahan citra (*image processing*) dapat digunakan untuk meningkatkan kualitas citra masukan.
2. Intensitas cahaya yang diterima oleh objek uji sangat berpengaruh terhadap hasil keputusan yang dikirim oleh komputer ke mikrokontroler.
3. Program yang dibuat mampu merespon tingkat pencahayaan yang diberikan dengan mengirimkan hasil yang akurat sebesar 80%.
4. Untuk lebih meningkatkan tingkat akurasi, maka dapat dicoba untuk meningkatkan kontras citra dengan berpatokan pada citra input dan memberikan proses reduksi noise dan *thresholding* beberapa kali .

V. DAFTAR REFERENSI

- [1] Ahmad, Balza, Ir, M.Sc.E, Firdausy, Kartika, ST., M.T, 2014, "*Teknik Pengolahan Citra Digital Menggunakan Delphi*", Yogyakarta, Ardi Publishing.
- [2] Ahmad, Usman, 2005, "Pengolahan Citra Digital & Teknik Pemrogramannya", Yogyakarta, Graha Ilmu
- [3] Susilo, Djoko, 2016, "Grafika Komputer Dengan Delphi", Yogyakarta, Graha Ilmu
- [4] Munir, Rinaldi, 2004, "*Pengolahan Citra Digital dengan pendekatan algoritmik*", Bandung, Penerbit Informatika