PEMBIDIK TANK OTOMATIS MENGGUNAKAN DETEKSI GERAK BERBASIS RASPBERRY PI

(AUTO TARGETING TANK USE MOTION DETECTION BASED ON RASPBERRY PI)

Aris Munandar, Iswahyudi Hidayat, Rizki Ardianto
Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
arismunandar@students.telkomuniversity.ac.id iswahyudihidayat@telkomuniversity.ac.id
rizkia@telkomuniversity.ac.id

Abstrak

Tank adalah kendaraan lapis baja yang bergerak menggunakan roda berbentuk rantai, serta memiliki meriam besar. Dalam peperangan tank bisa menjadi ancaman yang besar karena memiliki daya hancur yang besar dan mobilitas yang tinggi. Untuk mengatasi ancaman tersebut dibutuhkan sebuah sistem yang dapat mendeteksi keberadaan tank. Salah satu metode yang bisa digunakan adalah *motion detection*. Metode ini bisa mendeteksi sebuah pergerakan objek dengan melihat perubahan atau pergeseran nilai pixel dalam gambar yang ditangkap oleh kamera.

Dalam perancangan sistem ini menggunakan raspberry pi sebagai pengolahan citra (*image processing*). Raspberry pi digunakan karena memiliki waktu *booting* yang cepat, daya yang digunakan kecil dan tidak mudah rusak jika terjadi *power failure*. *Output* dari raspberry pi berupa koordinat tank yang akan dijadikan *input motor driver* untuk menggerakkan motor ke arah koordinat tersebut.

Hasil dari perancangan, sistem memiliki akurasi terhadap objek bergerak mencapai 100% dan akurasi terhadap objek yang menjadi target mencapai 76%. Sistem ini membutuhkan waktu untuk memroses citra sampai mengirim koordinat ke motor driver ±0,4s.

***Kata kunci : Motion detection*, Tank, Raspberry Pi**

Abstract

Tank is armored vehicle that moves with wheel shaped chain, and it has a big cannon. On war tank can become big problem, because it has weapon with big destruction power and high mobility. To overcome that problem, a sytem that can detect the presence of tank is needed. One of method that can detect the presence of tank is motion detection. This method can detect a moving object by looking at the change or shift pixel values in an image captured by camera.

In its design, raspberry pi is used for manage image processing. Raspberry pi is used because fast on booting, low in power usage and durable to power failure. Output from raspberry pi is used for input motor driver.

The result from design, system have 100% accuracy to moving object and up to 76% accuracy to targeted object. This system have time ±0,4s processing.

***Keywords : Motion detection*, Tank, Raspberry Pi**

1. Pendahuluan

Motion detection merupakan metode deteksi yang cukup sederhana namun memiliki banyak kegunaan, salah satunya adalah bisa digabungkan dengan object tracking untuk menghasil sebuah sistem pembidik tank otomatis pada penelitian ini. Digunakannya *motion detection* karena memiliki waktu proses yang cepat sekitar 0.4s serta sensitifitas yang tinggi terhadap gerak yang mencapai 100% ataupun bisa juga diatur oleh pengguna.

2. Pembidik Tank otomatis menggunakan *Motion detection*

A. *Motion detection*

Motion detection adalah salah satu hasil dari pemrosesan citra berfungsi untuk mendeteksi adanya pergerakan (orang, mobil, dan lain-lain). Sistem pendeteksi gerakan ini dapat menganalisa citra video dan

menentukan apakah telah terjadi pergerakan di area yang telah ditentukan oleh pengguna. Sensifitas sistem pendeteksi gerakan ini dapat disesuaikan dengan kondisi yang ada.

Salah satu metode yang dapat digunakan untuk membuat sebuah sistem *motion detection* adalah dengan melakukan pengurangan antar citra atau mencari perbedaan antar citra. Hal ini dapat dilakukan dengan merubah format video menjadi sebuah matriks tentunya ukurannya akan sama karena berasal dari video yang sama. Setelah diubah menjadi matriks, lakukan pengurangan atau bandingkan kedua matriks sehingga didapatkan perbedaan jika ada pergerakan di setiap citra. Jika tidak ada pergerakan objek pada setiap citra, maka nilai citra akan sama dan jika dikurangkan maka hasilnya nol (dalam format gambar hanya terlihat warna hitam).

B. *Object Tracking*

Object tracking juga merupakan hasil pemrosesan citra yang berfungsi untuk mendeteksi keberadaan sebuah objek. Salah satu metode yang dapat digunakan untuk *objeck tracing* adalah dengan melihat kontur.

Deteksi kontur adalah salah satu metode untuk mendeteksi objek berdasarkan intensitas warnanya. Untuk pendeteksian objek yang akurat, benda yang akan dideteksi harus berwarna putih dengan latar belakang hitam. Oleh karena itu sebelum menggunakan metode kontur diperlukan perubahan format warna citra dari RGB/BGR ke *Black and White* (BW). Pada proses perubahan format warna, digunakan nilai *threshold* 5,255. Maksud dari nilai tersebut adalah jika ada nilai *pixel* kurang dari sama dengan 5 makan diubah menjadi 0 dan jika lebih dari 5 diubah menjadi 255. Digunakan nilai tersebut agar objek yang ingin dideteksi terlihat lebih jelas dan sedikit *noise* yang terdeteksi.

C. Pengurangan citra

Pengurangan citra adalah salah suatu metode untuk mendeteksi gerak, cara kerja metode ini adalah dengan mengurangkan nilai setiap *pixel* dari gambar baru dengan gambar yang sebelumnya. Jika dalam gambar tidak ada pergerakan maka nilai *pixel* baru dengan sebelumnya pasti sama, maka jika dikurangkan akan menghasilkan nilai nol (warna hitam). Sedangkan jika ada pergerakan maka nilai *pixel* dari dua gambar akan ada perbedaan (tidak sama persis), dikarenakan sebelum diterapkannya metode ini gambar telah diubah ke format BW. Oleh karena itu jika ada perbedaan nilai pixel akan menyebabkan muncul warna putih, area putih yang muncul mengindikasikan adanya pergerakan.

D. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara waktu nyata, yang dibuat oleh Intel. Pustaka ini berada dalam naungan sumber terbuka (*open source*) dari lisensi BSD. Pustaka ini merupakan pustaka lintas *platform*. Pustaka ini didedikasikan sebagian besar untuk pengolahan citra secara waktu nyata. Jika pustaka ini menemukan pustaka *Integrated Performance Primitives* dari intel dalam sistem komputer, maka ia akan menggunakan rutin ini untuk mempercepat proses kerjanya secara otomatis.

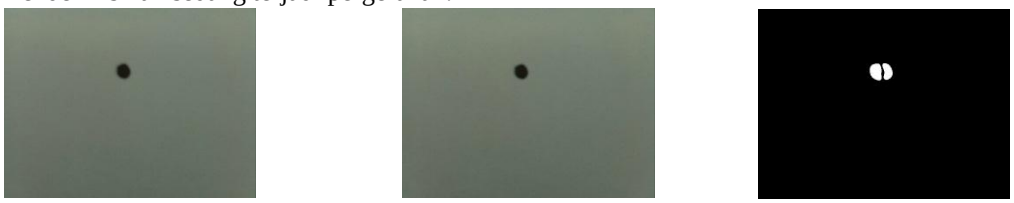
3. Simulasi Dan Implementasi pada Rapberry Pi

Simulasi dilakukan dengan mencoba beberapa metode sehingga didapatkan metode yang terbaik. Kemudian dari metode terbaik diuji menggunakan objek pengujian berupa titik, gambar tank dan miniatur tank. Hasil simulasi harus bisa mendeteksi objek bergerak dan mengirim koordinat tengah dari target ke *motor driver*.

3.1 Pengujian Waktu Respon Sistem Terhadap Gerakan

3.1.1 Pengurangan Citra dan Deteksi *Pixel* Putih

Metode ini adalah mengurangkan dua buah citra yang diambil secara waktu nyata menggunakan fungsi *capture* pada raspberry. Keluaran dari metode ini adalah sebuah citra hasil pengurangan dimana jika tidak ada pergerakan maka citra yang dihasilkan hanya berwarna hitam, sedangkan jika ada pergerakan akan menghasilkan sebuah citra dengan beberapa bagian berwarna putih (*pixel* putih) dengan latar belakang hitam. Kemudian *pixel* putih dideteksi dengan cara mencari nilai *pixel* yang tidak sama dengan nol. Jika ada maka sistem akan mendefinisikan sedang terjadi pergerakan.



Gambar (a) Gambar (b) Gambar hasil pengurangan (a) – (b)

Gambar 1. Pengurangan dua citra

3.1.2 Pengurangan Citra dan Deteksi *Pixel* Putih

Dalam metode ini sama seperti metode sebelumnya namun dalam pengambilan gambar secara waktu nyata dengan menggunakan mode *port video* pada opencv, sehingga bisa mengambil gambar lebih cepat.

3.1.3 Pengurangan Citra dan Deteksi Kontur

Metode ini menggunakan mode *port video* pada opencv untuk mengambil gambar, kemudian fungsi *contour* untuk mendeteksi area putih hasil pengurangan citra yang menandakan adanya sebuah pergerakan. Jika ada yang terdeteksi oleh fungsi *contour*, maka dapat disimpulkan ada sebuah pergerakan.

3.1.4 Hasil Pengujian

Berikut hasil pengujian dari metode-metode deteksi gerak :

Tabel 1. Hasil pengujian deteksi gerak

No.	Metode	Waktu respon (s)
1	Pengurangan Citra dan deteksi *pixel* putih.	± 2,9
2	Pengurangan Citra dan deteksi *pixel* putih.	± 0,5
3	Pengurangan Citra dan deteksi kontur	± 0,3

Dapat dilihat dari tabel 1. bahwa metode yang memiliki respon tercepat dalam mendeteksi gerak adalah metode pengurangan citra dan deteksi kontur. Oleh karena itu dalam tugas akhir ini metode tersebut yang digunakan.

3.2 Pengujian Deteksi Objek dan Pengenalan Objek

3.2.1 Membandingkan Citra Hasil Deteksi Gerak dengan *Database*

Dalam metode ini dicari persamaan nilai *pixel* dari citra waktu nyata hasil deteksi gerak dengan citra dari *database*. Jika ada nilai yang sama maka nilai *counter* kecocokan akan bertambah 1, kemudian nilai *counter* kecocokan akan dibandingkan dengan nilai *threshold* (ditentukan pengguna). Jika nilai *counter* kecocokan lebih besar atau sama dengan nilai *threshold* maka objek didefinisikan sebagai tank.

Gambar waktu nyata Gambar *database*

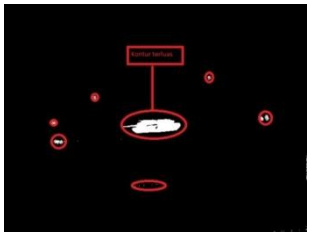
Gambar 2. Perbandingan citra waktu nyata dengan *database*

3.2.2 Membandingkan Nilai *Histogram* Citra Waktu nyata dengan *Database*

Untuk metode ini penulis memanfaatkan fungsi *cv2.CalcHist* untuk mencari nilai histogram dari citra dan *cv2.CompareHist* untuk mencari kesamaan dari kedua *histogram*. Jika nilai keluaran dari *cv2.CompareHist* lebih besar atau sama dengan *threshold* (ditentukan pengguna) maka objek didefinisikan sebagai tank.

3.2.3 Deteksi Kontur dan Asumsi Area Kontur Terluas sebagai Target

Pada metode ini menggunakan fungsi *cv2.FindContour* dan *cv2.ContourArea* untuk mendeteksi kontur yang ada pada citra dan luas dari kontur tersebut. Kontur terluas yang dideteksi akan diasumsikan sebagai objek yang ditargetkan.



Gambar 3. Deteksi kontur

3.2.4 Hasil Pengujian

Berikut tabel 2. berisi hasil pengujian metode deteksi objek :

Tabel 2. Metode *Object Detection*

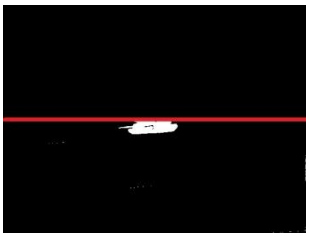
No.	Metode	Waktu proses(s)	Akurasi (%)
1	Membandingkan Citra Hasil Deteksi Gerak dengan *Database* (1 gambar)	± 3,8	≤ 60
2	Membandingkan Nilai *Histogram* Citra Waktu nyata dengan *Database* (1 gambar)	± 0,31	≤ 50
3	Mendeteksi Kontur kemudian Mengasumsikan Area Kontur Terluas adalah objek yang ditargetkan	± 0,38	≤ 76

Dari data tabel 2. metode yang memiliki waktu relatif cepat dan akurasi terbesar adalah metode mendeteksi kontur kemudian mengasumsikan area kontur terluas adalah objek yang ditargetkan. Oleh karena itu pada tugas akhir ini yang digunakan adalah metode tersebut.

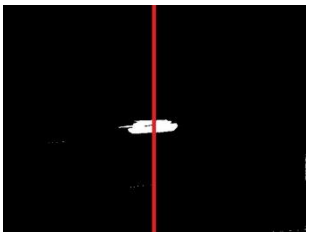
3.3 Pengujian Ketepatan Koordinat Arah Gerak Objek

3.3.1 Membagi Citra Menjadi 4 ROI (*Region of Image*) dan membandingkan nilai *pixel* putih dari ke-4 ROI

Dalam proses ini citra hasil deteksi gerak akan dibagi menjadi empat bagian. Kemudan bagian atas dengan bawah dibandingkan, yang memiliki nilai terbesar menentukan objek bergerak ke arah tersebut begitu juga dengan bagian kiri dan kanan.



ROI atas dan bawah



ROI kiri dan kanan

Gambar 4. ROI

Dari gambar 4. maka sistem akan mendefinisikan objek bergerak ke arah kiri-bawah.

3.3.2 Mendeteksi Koordinat Kontur Terluas

Pada proses ini memanfaatkan fungsi *cv2.ContourArea* untuk mendapatkan koordinat dari objek yang terdeteksi. Koordinat yang terdeteksi adalah bagian kiri-atas dari objek.

Gambar 5. Ilustrasi penentuan koordinat dari objek

3.3.3 Hasil Pengujian

Tabel 3. Data uji metode deteksi arah gerak

No.	Metode	Waktu Proses	Presisi dalam menentukan koordinat objek
1	Membagi Citra Menjadi 4 ROI (*Region of Image*) kemudian membandingkan nilai pixel putih dari ke-4 ROI	0,1	0%
2	Mendeteksi Koordinat Kontur Terluas	0,05	± 75%

Tabel 3. menunjukan hasil pengujian dari metode-metode penentu koordinat dari objek bergerak. Dari data tersebut metode yang memiliki waktu proses tercepat dan presisi dalam menentukan koordinat adalah metode mendeteksi koordinat kontur terluas.

3.4 Hasil Pengujian Sistem Keseluruhan Menggunakan Metode terbaik

Dalam pengujian ini objek yang berbentuk titik digerakkan ke kiri, kanan, atas dan bawah sejauh nilai pixel pergeseran. Sistem diharapkan dapat mendeteksi titik tersebut, hasil pengujiannya adalah sebagai berikut :

Tabel 4. Pengujian menggunakan 1 titik

No.	Pixel pergeseran	Arah	Benar	Kurang	Salah	Jumlah pergeseran
1	40	Kiri	7	0	1	8
2	40	Bawah	3	2	1	6
3	40	Kanan	6	1	1	8
4	40	Atas	4	2	0	6
5	80	Kiri	3	1	0	4
6	80	Bawah	3	0	0	3
7	80	Kanan	3	0	1	4
8	80	Atas	3	0	0	3
	Total		32	6	4	42

Dari data tabel 4. bisa dihitung, akurasi sistem = (32/44) x 100% = 76%

Tabel 5. Pengujian menggunakan 1 titik dengan *noise*

No.	Pixel pergeseran	Arah	Benar	Kurang	Salah	Jumlah pergeseran
1	40	horizontal	11	3	2	16
2	40	Vertikal	7	4	1	12
3	80	horizontal	5	3	0	8
4	80	vertikal	4	0	2	6
	Jumlah		27	10	5	42

Tabel 5. adalah hasil pengujian sistem terhadap sebuah titik yang dikelilingi oleh *noise* (berupa titik dengan ukuran yang lebih kecil dari titik utama). Sistem diharapkan tetap mengenali objek utama atau tidak terganggu oleh *noise*.

Dari data tabel 5. bisa dihitung, akurasi sistem = (27/44) x 100% = 61%

Tabel 6. Pengujian menggunakan 1 gambar tank (sebanyak 20 kali)

Benar	Salah	Total
16	4	20

Tabel 6. merupakan hasil pengujian pengenalan sistem terhadap sebuah gambar tank yang digerakkan ke kiri-kanan dan ke atas-bawah.

Gambar 6. Objek uji sebuah gambar tank

Tabel 7. Pengujian menggunakan 1 gambar tank (sebanyak 20 kali) dengan arah menjauh/mendekat.

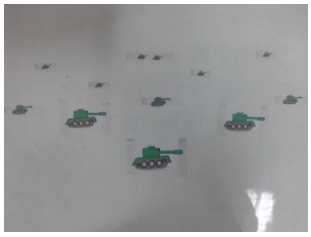
Benar	Salah	Total
12	8	20

Tabel 7. adalah hasil pengujian dari sistem menggunakan objek uji seperti pada gambar 4.1 dengan menjauhkan dan mendekatkan objek pada kamera.

Tabel 8. Pengujian menggunakan 11 gambar tank (sebanyak 20 kali)

Ukuran tank (cm^2)	Jumlah tank	Total terdeteksi
32	1	8
18	2	7
8	3	5
2	5	0

Tabel 8. merupakan hasil pengujian sistem terhadap 11 gambar tank dengan ukuran yang berbeda-beda. Dari data diatas bisa disimpulkan bahwa sistem dapat mendeteksi dengan benar jika tidak ada *noise* atau objek serupa yang memiliki ukuran 0,06 dari objek yang menjadi target utama.



Gambar 7. Objek uji 11 tank dengan empat ukuran berbeda

Tabel 9. pengujian menggunakan 1 miniatur tank

No.	Pixel pergeseran	Arah	Benar	Kurang	Salah	Jumlah pergeseran
1	20	Kiri	5	0	3	8
2	20	Kanan	5	0	3	8
3	40	Kiri	6	2	0	8
4	40	Kanan	8	0	0	8
	Jumlah		24	2	6	32

Tabel 9. merupakan hasil pengujian sistem dengan objek sebuah miniatur tank yang digerakan ke kanan dan ke kiri didepan kamera dengan pergeseran sebanyak 20 dan 40 pixel.

Dari data tabel 9. bisa dihitung, akurasi sistem = (24/32) x 100% = 75%

4. Kesimpulan

Berdasarkan hasil pengujian pada penelitian ini, sistem memiliki sensitifitas terhadap objek yang bergerak mencapai 100% dan akurasi terhadap objek yang menjadi target mencapai 76%. Waktu proses yang dibutuhkan sistem mulai dari pengambilan gambar sampai mengirim koordinat ke *motor driver* dengan menggunakan metode yang paling optimal adalah ±0,4s.

Daftar Pustaka:

[1] Rakhman, Edi, Faisal Candrasyah dan Fajar D.Sutera. 2014. RaspberryPI Mikrokontroler Mungil yang Serba Bisa. Yogyakarta:ANDI.

[2] Rosebrock, Adrian. (2014) Citra Search Engine Resource Guide. Tersedia di : https://www.getdrip.com/forms/1113813/submissions [diakses 10 Agustus 2015]

[3] Jaya P et al. "Contour Based Object Tracking" (IJCSIT) International Journal of Computer Science and Information Technologies. Vol. 5 (3) , 2014, 4128-4130

[4] http://opencv-python-tutroals.readthedocs.org/en/latest/index.html [diakses 10 Agustus 2015]

[5] https://electrosome.com/uart-raspberry-pi-python/ [diakses 10 Agustus 2015]