

Design and Implementation Automatic Scoring Computation System for Shooting Sport with Template Matching Algorithm

Rancang Bangun Sistem Penghitung Skor Otomatis Olahraga Menembak Menggunakan Algoritma *Template Matching*

Hari Harmaen

Fakultas Ilmu Komputer
Universitas Bina Nusantara
Jakarta
Indonesia

Abstract

This paper presents an Automatic Scoring System (ASS) using image processing method for processing the image of shooting target and bullet trail. Compared with other methods, such as acoustic method, image processing method is more efficient. Grab Cut and Template Matching Algorithm will be used in the background segmentation and positioning the bullet trail. prototype ASS applications built using IP Camera, EMGU CV and C#, is divided into two modules, namely camera calibration module and score module. Based on the results of testing on the target paper 10m air rifle image, compared with manual / visual by officers, ASS application provide better score. 1.67% scoring discrepancy occurred due to the poor quality of shooting target paper used.

Key Words: automatic scoring system, grab cut algorithm, Template Matching algorithm

Abstrak

Makalah ini mempresentasikan aplikasi Penghitung Skor Otomatis (PSO) menggunakan metode pengolahan citra (image processing) dalam mengolah citra target tembak dan jejak peluru. Dibandingkan dengan metode lain, seperti metode *acoustic*, metode *image processing* lebih efisien. Algoritma Grab Cut dan algoritma *Template Matching* akan digunakan pada proses background segmentation dan proses pencarian posisi jejak peluru. *prototype* aplikasi PSO dibangun menggunakan Personal Computer, IP Camera, EMGU CV dan C#, aplikasi terdiri dari dua modul, yaitu modul kalibrasi kamera dan modul penghitung skor. Berdasarkan hasil uji coba pada citra kertas target tembak 10m *air rifle*, dibandingkan dengan cara manual / visual oleh petugas, aplikasi PSO memberikan hasil penilaian skor menembak yang lebih baik. Kesalahan pemberian skor hanya sebesar 1.67

Kata kunci: penghitung skor otomatis, algoritma *Grab Cut*, algoritma *Template Matching*

1. PENDAHULUAN

Mahalnya harga perangkat Penghitung Skor Otomatis (PSO) menyebabkan penggunaannya di Indonesia sangat kurang, tercatat hanya dua daerah yang melengkapi venue menembaknya dengan perangkat PSO, yaitu DKI Jakarta dan Sumatera Selatan.

Perkembangan teknologi informasi (TI) telah mendorong pengembangan industri terkait lainnya.

Efisiensi pelatihan olahraga dapat meningkat, jika sistem pelatihan dikembangkan secara terintegrasi antara TI dengan teknologi otomatis lainnya. Keterampilan olahraga dari atlet akan meningkat, jika dalam suatu pelatihan olahraga, informasi hasil latihan memberi umpan balik yang tepat [1]. Dengan demikian penggunaan perangkat PSO dalam latihan menembak akan sangat membantu untuk mengevaluasi dan meningkatkan kinerja atlet menembak, sekaligus untuk memodernisasi venue menembak yang ada.

Penggunaan metode pengolahan citra dalam sistem PSO telah lama diteliti, penelitian ditujukan untuk mendapatkan tingkat akurasi penghitungan skor yang tinggi. Dibandingkan dengan metode lain

*Corresponding Author. Tel: +6221-5340660

Email: hari.harmaen@gmail.com

Received: 16 Oct 2012; revised: 8 Nov 2012; accepted: 18 Nov 2012

Published online: 26 Nov 2012

© 2012 INKOM 2012/13-NO187

seperti metode acoustic [2], metode pengolahan citra lebih efisien terutama dalam hal biaya selain tingkat akurasi yang semakin baik [3]. Isi makalah ini dibagi menjadi beberapa bagian antara lain: bab 2 menjelaskan permasalahan dalam rancang bangun suatu sistem PSO, bab 3 menjelaskan mengenai metode yang dilakukan pada penelitian ini, bab 4 menampilkan hasil uji coba dan bab 5 simpulan.

2. DESKRIPSI MASALAH

Algoritma *Template Matching* telah banyak digunakan untuk mengidentifikasi pola-pola dalam citra yang memiliki kerumitan rendah, namun demikian algoritma ini cukup rentan terhadap perbedaan orientasi antara citra acuan (*template*) dengan citra yang akan diidentifikasi, yang meliputi: ukuran, posisi dan kualitas citra, sehingga tahap pre-processing citra menjadi sangat penting.

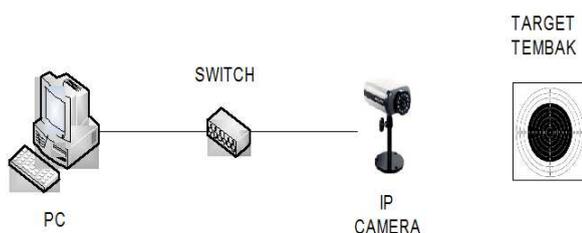
Bayangan pada citra jejak peluru menjadi masalah tersendiri, karena dapat mengurangi akurasi dalam penilaian skor menembak, sehingga diperlukan suatu cara untuk mengurangi atau bahkan menghilangkan dampak bayangan ini.

Penelitian ini diarahkan untuk menyelesaikan masalah dalam rancang bangun suatu sistem PSO pada cabang olahraga menembak dengan mengolah citra target tembak dan jejak peluru hasil tangkapan kamera menggunakan algoritma *Template Matching* secara real-time.

3. METODOLOGI PENELITIAN

3.1 Bahan penelitian

Penelitian ini akan menggunakan bahan penelitian berupa citra dari target tembak 10m *air rifle* dan jejak peluru yang ditangkap oleh kamera sebanyak 2 x 60 tembakan (120 citra). Sebagai alat bantu dalam pengujian sistem PSO akan digunakan komputer dengan sistem operasi Microsoft Windows 7 32-bit, C# dan EMGU CV (OpenCV wrapper for .NET). Untuk konfigurasi peralatan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Konfigurasi peralatan

3.2 Perancangan dan pengembangan *prototype* sistem PSO

Desain *prototype* sistem PSO dibagi menjadi 2 tahap, yaitu tahap Kalibrasi Kamera, kemudian tahap Pengenalan dan Penilaian Hasil Menembak seperti pada Gambar 2. Tahap Kalibrasi Kamera bertujuan untuk menghilangkan distorsi geometri [4], pada tahap ini citra papan catur hasil tangkapan dari kamera, diambil, kemudian dilakukan proses pendeteksian posisi sudut dari setiap kotak yang terdapat pada citra papan catur. Posisi sudut yang telah terdeteksi tersebut, kemudian diproyeksi ulang, hingga distorsi geometri pada citra papan catur hilang. Selanjutnya terhadap citra hasil proyeksi ulang tersebut, dilakukan proses transformasi perspektif dengan menggunakan posisi sudut terluar dari kotak papan catur sebagai acuan. Hasil akhir dari tahap Kalibrasi Kamera ini akan disimpan sebagai parameter intrinsik dan ekstrinsik dari kamera yang digunakan. Parameter kamera tersebut akan digunakan pada tahap Pengenalan dan Penilaian.

Pada tahap Pengenalan dan Penilaian Hasil Menembak, ketika peluru telah ditembakkan, citra target tembak dan jejak peluru hasil tangkapan kamera, diambil, kemudian dilakukan pre-processing seperti pada Gambar 3, dengan tujuan untuk memperbaiki kualitas citra, ekstraksi citra target dan citra jejak peluru yang akan digunakan dalam proses background segmentation dengan algoritma *Grab Cut* dan algoritma *Template Matching*, setelah itu dilakukan proses pemberian nilai dan menampilkan hasil akhir dari citra target dan jejak peluru.

3.3 Pengujian dan Evaluasi Sistem

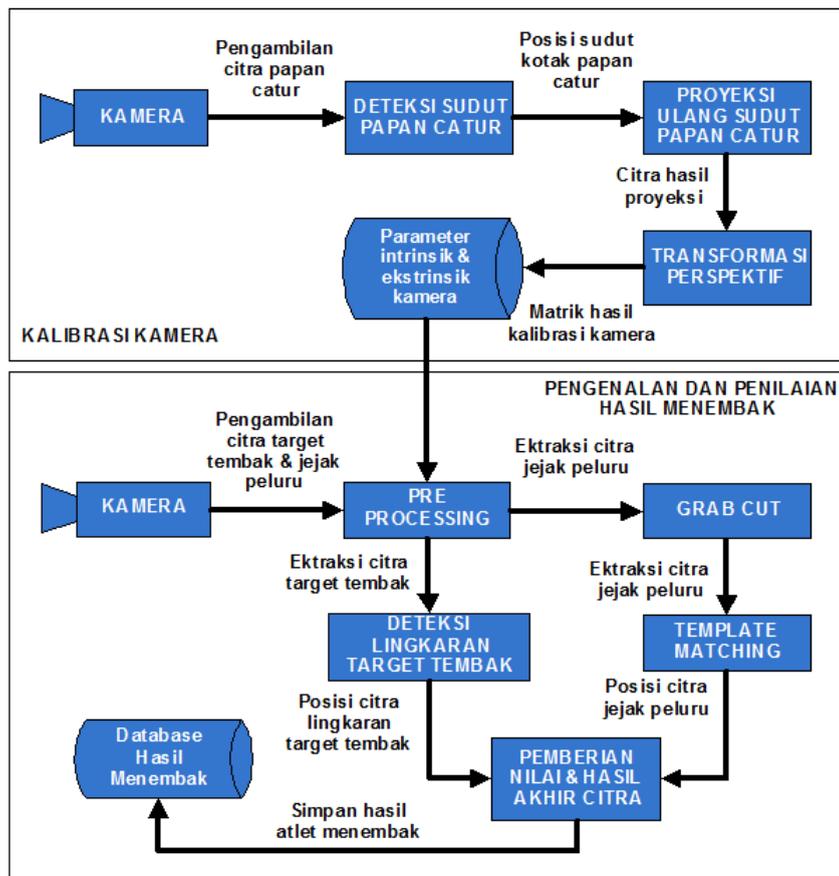
Pengujian dan evaluasi sistem dilakukan untuk mendapatkan waktu proses dan tingkat keakurasian dari metode-metode yang digunakan pada tahap Kalibrasi Kamera maupun tahap Pengenalan dan Penilaian Hasil Menembak dengan cara membandingkan skor hasil dari sistem PSO dengan skor hasil penghitungan manual / visual terhadap 120 citra target tembak dan jejak peluru.

4. HASIL UJI COBA

Pengujian pada *prototype* aplikasi yang dikembangkan dalam penelitian ini, dibagi menjadi dua bagian, bagian pertama pengujian kalibrasi kamera, bagian kedua pengujian pengenalan dan penilaian hasil menembak.

4.1 Pengujian kalibrasi kamera

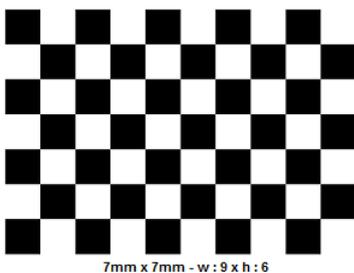
Data penelitian berupa citra papan catur [5] seperti pada Gambar 4, yang terdiri dari lebar 9 sudut, tinggi



Gambar 2. Desain *prototype* sistem PSO

6 sudut, dengan masing-masing kotak berukuran 7 mm x 7 mm.

akan diberi tanda Detected jika sudut terdeteksi dan Invalid jika sudut tidak terdeteksi.



Gambar 4. Citra papan catur

Tujuan dari kalibrasi kamera ini adalah untuk menghilangkan distorsi geometri, yang disebabkan oleh posisi relatif 3 dimensi antara kamera dengan obyek, distorsi lensa atau faktor lainnya [4]. Tahap pengujiannya terdiri dari dua bagian, yaitu pengujian deteksi sudut papan catur dan pengujian koreksi geometri.

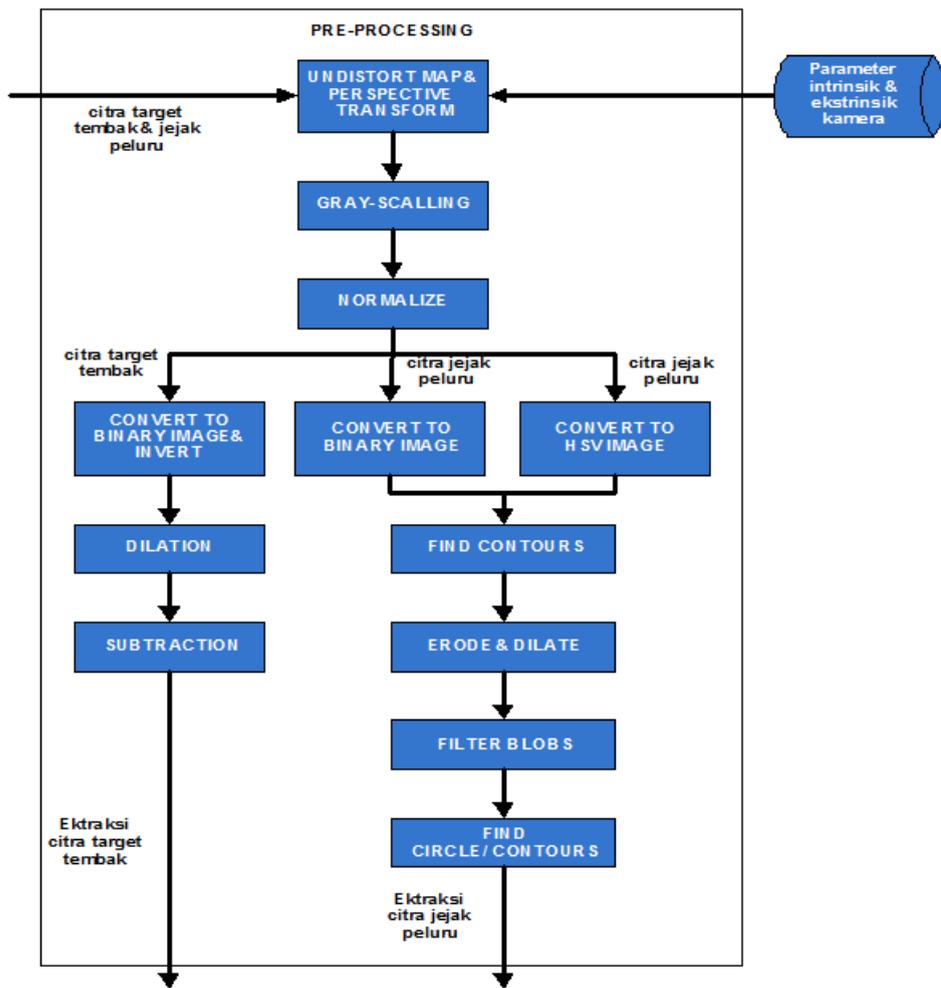
4.1.1 Pengujian deteksi sudut papan catur

Pada tahap ini citra papan catur akan ditangkap kamera sebanyak minimal 15 kali, untuk setiap citra



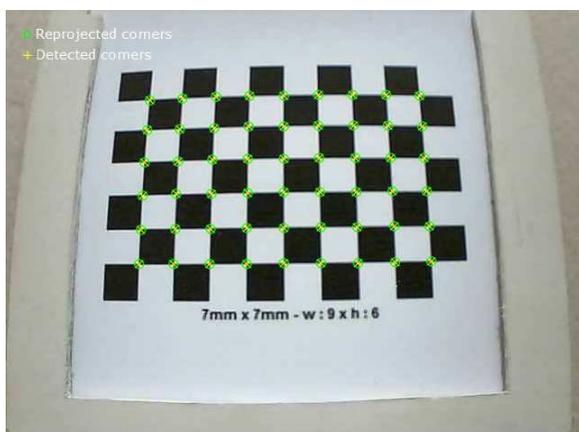
Gambar 5. Proses deteksi sudut papan catur

Citra papan catur pada Gambar 5 ditangkap oleh kamera yang berada di bawahnya, kamera tidak tegak lurus dengan citra, sudut kemiringan disesuaikan hingga citra cukup jelas untuk diproses. Tampak pada citra terdapat distorsi radial, dimana citra papan catur terlihat cembung. Pada tahap



Gambar 3. Pre-processing tahap pengenalan dan penilaian hasil menembak

pengujian ini sudut berhasil dideteksi oleh aplikasi, untuk setiap baris sudut yang terdeteksi diberi warna berbeda dan disimpan posisinya. Hasil dari proses ini ditunjukkan seperti pada Gambar 6.



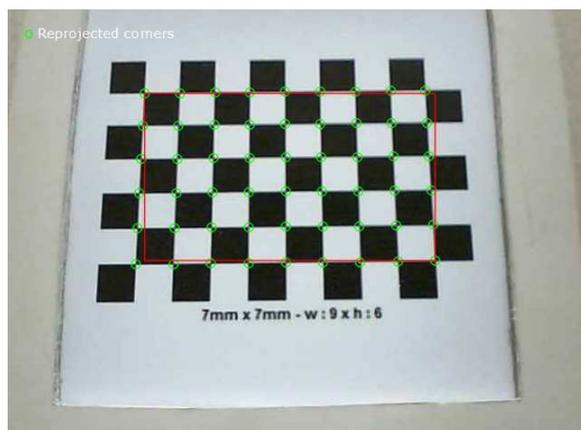
Gambar 6. Hasil deteksi sudut papan catur

4.1.2 Pengujian koreksi geometri

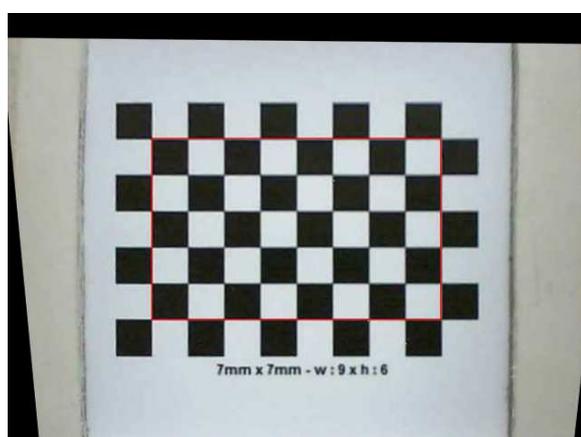
Pengujian koreksi geometri akan menggunakan data hasil proses deteksi sudut papan catur. Setiap titik sudut papan catur akan diproyeksikan ke titik sudut koreksi perspektif, untuk menghilangkan distorsi yang ada. Hasilnya seperti yang ditunjukkan oleh Gambar 7.

Pada Gambar 7 tampak distorsi radial sudah hilang, namun demikian pada citra masih terdapat distorsi geometri lain seperti yang ditunjukkan oleh garis merah pada gambar. Untuk mengkoreksinya, maka sudut terluar dari papan catur akan diproyeksi kembali ke posisi garis merah menggunakan transformasi perspektif dengan tetap mempertahankan aspect ratio dari citra. Gambar 8 menunjukkan hasil dari proses transformasi tersebut.

Pada Gambar 8, terlihat bahwa pada bagian atas citra jadi memanjang dan miring ke kanan, setiap sudut terluar dari papan catur, tepat berada pada garis merah. Dari pengujian kalibrasi kamera



Gambar 7. Hasil proyeksi sudut



Gambar 8. Hasil transformasi perspektif

ini dapat disimpulkan bahwa aplikasi kalibrasi kamera yang dikembangkan, sudah memenuhi tujuan dari proses kalibrasi kamera ini. Hasil dari proses kalibrasi kamera ini akan disimpan sebagai parameter intrinsik dan ekstrinsik dari kamera, dan akan digunakan oleh aplikasi menembak.

4.2 Pengujian pengenalan dan penilai-an hasil menembak

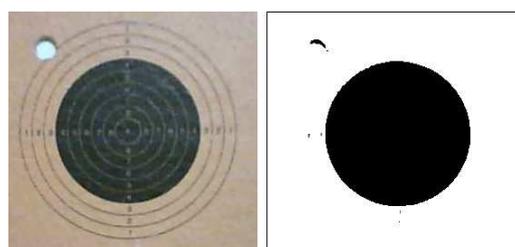
Data penelitian berupa citra dari target tembak 10m *air rifle* dan jejak peluru yang ditangkap oleh kamera sebanyak 2 x 60 tembakan (120 citra). Tujuan dari akhir pengujian ini adalah untuk menilai akurasi aplikasi menembak dalam mengenali citra target tembak dan jejak peluru, sehingga dapat menghasilkan skor menembak yang lebih akurat dibandingkan dengan cara manual / visual. Pengujian dilakukan dalam tiga tahap, yaitu pengenalan target tembak, pengenalan jejak peluru, dan penilaian skor menembak.

4.2.1 Pengujian pengenalan target tembak

Pada tahap ini, aplikasi diuji untuk mengenali citra target tembak, menentukan posisi titik tengah

dan jari-jari lingkaran nomor 4 dari target tembak dengan menggunakan fungsi-fungsi pengolahan citra. Berikut tahap-tahap pengujian yang dilakukan:

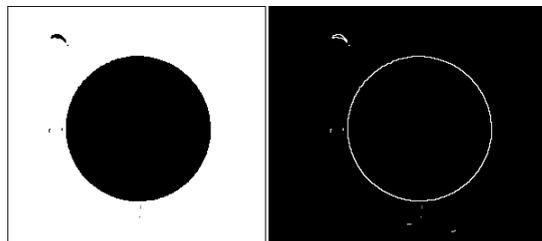
- (1) Konversi citra dari sistem warna RGB (Red, Green, Blue) menjadi citra binari dengan fungsi filter warna (InRange) dan fungsi Not (invert), dengan menggunakan nilai *threshold* minimum $rgb(0, 0, 0)$ dan maksimum $rgb(128, 128, 128)$.



Gambar 9. Hasil konversi RGB ke binari

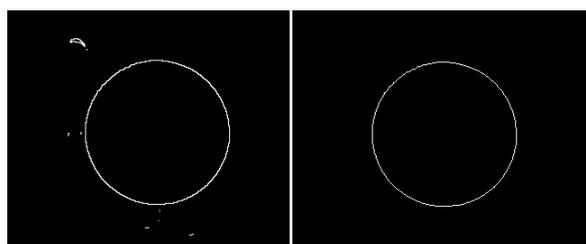
Dari hasil uji coba nilai maksimum $rgb(128, 128, 128)$ merupakan nilai terbaik karena menghasilkan sedikit obyek *shape* / blob pada citra binari.

- (2) Membuat edge / tepi dari obyek *shape* / blob dengan fungsi Dilate dan Subtract

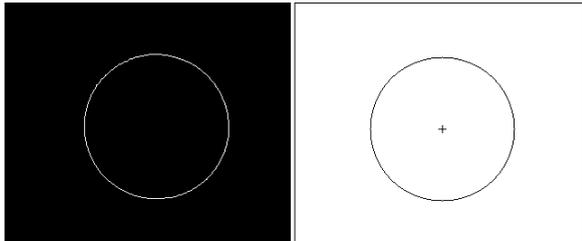
Gambar 10. Hasil pembuatan edge obyek *shape*

Pada Gambar 10 selain obyek *shape* target juga terdapat obyek *shape* lain yang tidak diinginkan, untuk menghilangkannya akan dilakukan proses filter *shape* / blob dengan fungsi *FindContours*.

- (3) Filter *shape* / blob dengan fungsi *FindContours*

Gambar 11. Hasil filter *shapel* blob

shape / blob difilter menggunakan fungsi FindContours, dengan kondisi *shape* / blob harus mempunyai luas area, minimal 16000 dan maksimal 24000, nilai minimal dan maksimal area didapat dari hasil uji coba. Hasil dari proses filter ditunjukkan pada Gambar 11, selanjutnya ditentukan titik tengah dari target seperti pada Gambar 12 dengan menggunakan posisi dan radius contours.



Gambar 12. Titik tengah target

- (4) Deteksi Semua Lingkaran Target Selain berfungsi untuk memfilter *shape* / blob, fungsi FindContours juga menghasilkan posisi (x, y) dan ukuran (lebar, tinggi) *shape* / blob pada citra, sehingga dapat digunakan untuk mencari titik tengah target sekaligus jari-jari lingkarannya. Dengan menggunakan ukuran kertas target 10 m *air rifle* pada Tabel I.

Tabel I. Ukuran kertas target 10m *air rifle*

Ring	Diameter (mm)	Warna
10	0.5 (±0.1)	Hitam
9	0.5 (±0.1)	Hitam
8	0.5 (±0.1)	Hitam
7	0.5 (±0.1)	Hitam
6	0.5 (±0.1)	Hitam
5	0.5 (±0.1)	Hitam
4	0.5 (±0.1)	Hitam
3	0.5 (±0.1)	Putih
2	0.5 (±0.1)	Putih
1	0.5 (±0.1)	Putih

Jari-jari semua lingkaran target dapat diketahui, dari hasil penghitungan didapatkan skala ukuran 1 pixel = 0.25 mm. Hasilnya tahap ini dapat dilihat pada Gambar 13.



Gambar 13. Hasil deteksi semua lingkaran

Dari hasil pengujian pengenalan target menembak di atas, dapat disimpulkan bahwa aplikasi menembak yang dikembangkan telah dapat mendeteksi target tembak dengan baik.

4.2.2 Pengujian pengenalan jejak peluru

Pada tahap ini, aplikasi diuji untuk mengenali citra jejak peluru dan menentukan posisi titik tengah jejak peluru dengan menggunakan fungsi-fungsi pengolahan citra. Diameter peluru untuk 10 m *air rifle* adalah 4.5 mm, dengan demikian jari-jari peluru dapat dicari, menggunakan skala ukuran pixel yang telah didapat dari tahap pengenalan target tembak. Berikut tahap-tahap pengujian yang dilakukan, tahap a dan b merupakan tahap tersendiri bukan berupa urutan proses:

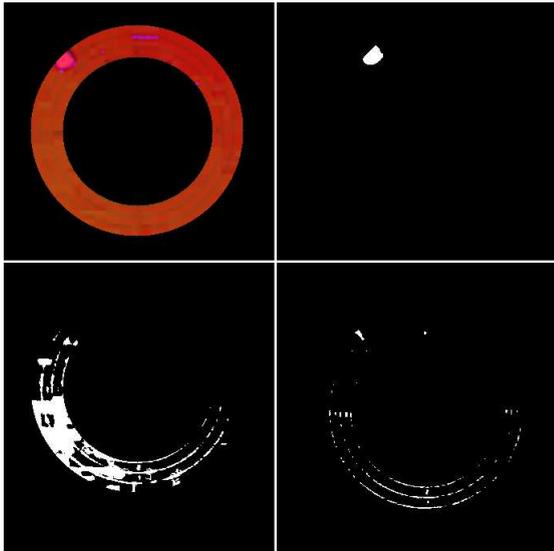
- (1) Konversi citra dari sistem warna RGB ke citra binari menggunakan fungsi *thresholdBinary* dengan nilai *threshold* minimum 128 dan maksimum 255.



Gambar 14. Hasil konversi RGB ke binari

Nilai *threshold* minimum dan maksimum yang digunakan merupakan nilai yang terbaik karena menghasilkan sedikit obyek *shape* / blob. Pada Gambar 14 obyek *shape* / blob peluru hanya muncul sebagian, tidak berupa lingkaran penuh. Hal ini dapat dijelaskan dengan melihat citra sumber, dimana lubang jejak peluru tertutup serat dari kertas target, sehingga mengurangi pantulan cahaya.

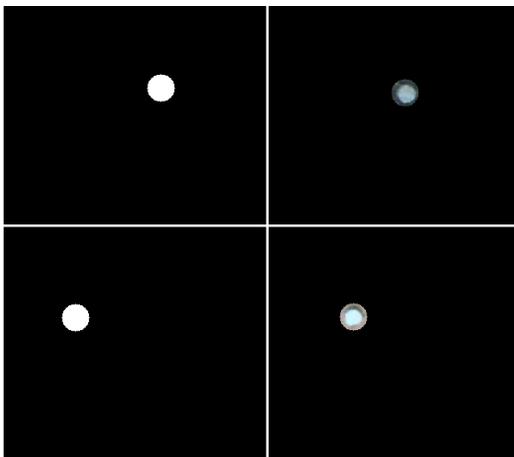
- (2) Konversi citra dari sistem warna RGB ke sistem warna HSV (*Hue, Saturation, Value*), kemudian citra HSV dibagi menjadi kanal *Hue*, kanal *Saturation* dan kanal *Value*, selanjutnya ketiga kanal tadi dirubah menjadi citra binari dengan menggunakan fungsi InRange dengan nilai *threshold* minimum 60 dan maksimum 150. Dari hasil uji coba nilai *threshold* minimum 60 dan maksimum 150 merupakan nilai terbaik karena menghasilkan sedikit obyek *shape* / blob pada citra binari. Berutan searah jarum jam, pada Gambar 15 adalah citra HSV, citra kanal *Hue*, citra kanal *Value* dan citra kanal *Saturation*. Dari 3 kanal yang ada, kanal *Hue* akan digunakan pada proses selanjutnya (tahap c). Konversi dari sistem warna RGB ke sistem



Gambar 15. Hasil konversi RGB ke HSV

warna HSV diperlukan untuk mendeteksi jejak peluru yang berada di luar lingkaran 4 target tembak. Jika menggunakan hasil konversi tahap a, obyek *shape / blob* jejak peluru tidak terdeteksi atau terlalu banyak obyek *shape / blob* yang memiliki luas area yang sama, sehingga akan sulit untuk memilahnya.

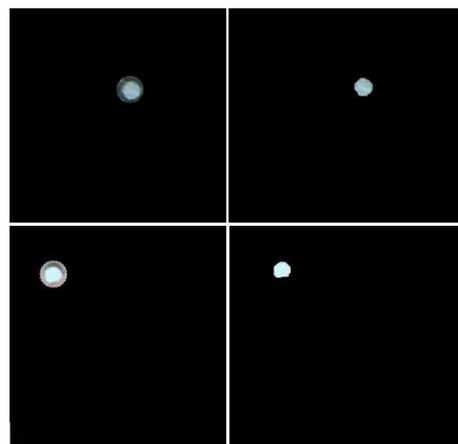
- (3) Deteksi lingkaran jejak peluru dengan fungsi `FindContours` Fungsi `FindContours` digunakan untuk mendeteksi obyek *shape / blob* dari jejak peluru, dengan kondisi *shape / blob* harus mempunyai luas area, minimal 70 dan maksimal 2000, nilai minimal dan maksimal area didapat dari hasil uji coba. Gambar 16 merupakan hasil dari proses deteksi lingkaran jejak peluru.



Gambar 16. Hasil deteksi lingkaran dengan fungsi `FindContours`

Bagian atas Gambar 16 merupakan hasil deteksi jejak peluru dengan menggunakan citra binari hasil dari tahap a, bagian bawah menggunakan citra binari hasil tahap b. Dengan menggunakan posisi (x, y) dan ukuran (lebar, tinggi) yang didapat dari fungsi `FindContours`, obyek *shape / blob* diperbesar seperti yang terlihat pada bagian kiri Gambar 16 dan digunakan sebagai mask untuk menyalin citra sumber, sehingga yang muncul adalah citra dari jejak peluru, seperti yang terlihat pada bagian kanan Gambar 16. Sampai tahap ini dengan membandingkan posisi (x, y) dari jejak peluru dengan posisi (x, y) dari target lingkaran 4, dapat ditentukan posisi dari jejak peluru. Pada kondisi di atas, untuk Gambar 16 bagian atas, jejak peluru berada di dalam lingkaran 4 target tembak, untuk Gambar 16 bagian bawah, jejak peluru berada di luar lingkaran 4 target tembak. Namun titik tengah dari jejak peluru belum dapat ditentukan, karena seperti yang terlihat pada bagian kanan Gambar 16, citra jejak peluru belum tentu berada tepat di tengah-tengah lingkaran.

- (4) Background segmentation Citra jejak peluru hasil dari tahap c, harus diproses lagi untuk memisahkan antara background (target tembak) dengan foreground (jejak peluru), sekaligus untuk mengurangi dampak dari bayangan pada citra jejak peluru. Background Segmentation menggunakan fungsi `GrabCut` yang merupakan implementasi dari model segmentasi yang dikembangkan oleh Rother et al [6]. Gambar 17 merupakan hasil dari proses background segmentation.



Gambar 17. Hasil background segmentation

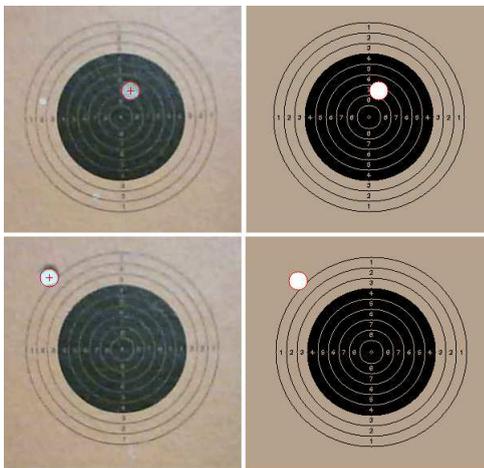
Hasil proses background segmentation akan dijadikan template untuk menentukan posisi yang lebih tepat dari jejak peluru dengan menggunakan fungsi *Template Matching*.

- (5) Pembuatan citra template jejak peluru Dengan menggunakan posisi (x, y) dan ukuran (lebar, tinggi) jejak peluru yang didapat dari proses tahap c, maka citra hasil proses tahap d, di-crop dengan memanfaatkan fungsi Region Of Interest (ROI), hasilnya seperti yang terlihat pada Gambar 18.



Gambar 18. Template jejak peluru

- (6) Mencari posisi titik tengah jejak peluru dengan *Template Matching* Citra template jejak peluru akan digunakan untuk mencari posisi jejak peluru pada citra target tembak dengan menggunakan fungsi MatchTemplate dengan algoritma Normalized Correlation Coefficient (NCC - Fast Normalized Cross-Correlation) yang dikembangkan oleh JP. Lewis [7]. Algoritma ini berdasarkan uji coba memberikan hasil kecocokan yang terbaik, menghasilkan nilai kecocokan = 1.0 (perfect match). Selain mencari kecocokan citra template, fungsi ini juga memberikan nilai balik posisi (x, y) dari citra template, dengan demikian posisi titik tengah dari jejak peluru dapat dicari. Gambar 19 merupakan hasil dari fungsi MatchTemplate ini.



Gambar 19. Hasil fungsi MatchTemplate NCC

Dari hasil pengujian pengenalan jejak peluru di atas, dapat disimpulkan bahwa aplikasi menembak yang dikembangkan telah dapat mendeteksi jejak peluru dengan baik dan membutuhkan waktu proses 5 detik, pada saat aplikasi baru mulai dijalankan, selanjutnya hanya membutuhkan waktu proses 2 detik.

procedure MENGHITUNG SKOR(rp , ARL)

▷ rp adalah jari jari peluru
▷ ARL adalah array jari jari lingkaran

```

 $rl = ARL[1]$ 
 $ab = \text{Jarak}(A, B)$ 
if  $ab < rl$  then
    Skor = 0
else
     $i = 1$ 
    while  $i \leq 10$  do
         $rl = ARL[i]$ 
        if  $ab = rl$  then
            Skor =  $i + 0.99$ 
        else if  $ab < rl$  then
            Skor =  $i + ((rl - ab)/10) + (rp/10)$ 
        else if  $(abrp) < rl$  then
            Skor =  $i + ((rl - ab - rp)/10)$ 
        end if
         $i = i + 1$ 
    end while
    end if
end procedure

```

Gambar 20. Algoritma menghitung skor

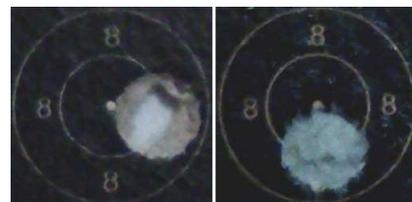
4.2.3 Pengujian penilaian hasil menembak

Setelah mendapatkan posisi (x, y) titik tengah dari target tembak (A) dan jejak peluru (B), selanjutnya adalah mencari jarak antara kedua titik tengah (AB), dengan menggunakan rumus Pythagoras:

$$\text{Jarak}(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (1)$$

Berikutnya adalah mencari skor dari jejak peluru dengan cara seperti Gambar 20.

Dari hasil uji coba penilaian skor menembak dengan menggunakan aplikasi menembak yang dikembangkan, terdapat 2 data dari 120 data yang salah skornya atau secara prosentase 1.67% jika dibandingkan dengan skor secara visual / manual. Kesalahan ini terjadi karena lubang peluru tidak bulat sempurna atau pada permukaan kertas target tembak terdapat sobekan dekat lubang peluru, seperti algoritma menghitung skor pada Gambar 21.



Gambar 21. Lubang peluru rusak

5. KESIMPULAN

Berdasarkan hasil pengujian dan pembahasan yang telah dikemukakan, maka hasil dari penelitian

ini dapat disimpulkan sebagai berikut: kualitas kamera dengan spesifikasi yang lebih tinggi dapat memberikan hasil yang lebih baik pada saat tahap Kalibrasi kamera maupun tahap Pengenalan dan Penilaian Hasil Menembak, proses Kalibrasi kamera harus dilakukan dengan benar karena parameter intrinsik dan ekstrinsik (koefisien distorsi dan transformasi perspektif) dari kamera akan digunakan tahap Pengenalan dan Penilaian Hasil Menembak yang dapat mempengaruhi tingkat akurasi aplikasi menembak dalam mengenali citra target tembak dan citra jejak peluru, kualitas kertas target tembak ikut menentukan tingkat akurasi dari aplikasi menembak dalam memberi skor, dan lampu untuk pencahayaan pada area target tembak harus diposisikan sedemikian rupa untuk mengurangi bayangan pada kertas target tembak.

Selain itu, penggunaan fungsi GrabCut yang bertujuan untuk mengurangi dampak bayangan pada citra jejak peluru, disamping memberikan hasil yang baik jika dibandingkan dengan cara segmentasi warna dengan *threshold* tertentu, fungsi GrabCut juga menambah waktu proses tahap Pengenalan dan Penilaian Hasil Menembak sebesar 2 - 3 detik terutama pada saat pertama kali aplikasi menembak dijalankan, dan pemilihan algoritma *Template Matching Normalized Correlation Coefficient* (NCC - Fast Normalized Cross-Correlation) memberikan hasil yang baik, pada aplikasi menembak ini ditetapkan nilai kecocokan = 1.0 (*perfect match*)

yang artinya tidak boleh ada sedikitpun yang berbeda.

Daftar Pustaka

- [1] S. Ying, W. Gang, and W. Yaojun, "The application of information technology in sports training," in *International Conference on Future Computer Science and Education*, 2011, pp. 210–212.
- [2] B. G. Mobasseri, "Automatic target scoring system using machine vision," *Machine Vision and Applications*, vol. 8, no. 1, pp. 20–30, 1995.
- [3] F. Ali and A. B. Mansoor, "Computer vision based automatic scoring of shooting targets," in *IEEE International Multitopic Conference, INMIC 2008*, 2008, pp. 515–519.
- [4] C. Ye and H. Mi, "The technology of image processing used in automatic target-scoring system," in *Fourth International Joint Conference on Computational Sciences and Optimization*, 2011, pp. 349–352.
- [5] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Computer Vision and Pattern Recognition*, 1997, pp. 1106–1112.
- [6] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut - interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 2004, pp. 309–314.
- [7] J. Lewis, "Fast templatematching," in *Vision Interface 95, Canadian Image Processing and Pattern Recognition Society*, 1995, pp. 120–123.