

SISTEM PENDETEKSIAN TANDA PENGENAL DI SEBUAH GEDUNG UNTUK MENENTUKAN SASARAN TEMBAK MUSUH (MENGIDENTIFIKASI POSISI MUSUH) BERBASIS TEMPLATE MATCHING

Riza Hasbi Ash Shiddieqy, Rahmadwati dan Panca Mudjirahardjo

Jurusan Teknik Elektro, Universitas Brawijaya Malang
Jl. Veteran, Ketawanggede, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145, Indonesia

**) E-mail: rhasbi@student.ub.ac.id, rahmadwati@ub.ac.id, panca@ub.ac.id*

Abstrak

Pengenalan dari berbagai sudut pandang atau *point of view* untuk mendeteksi keberadaan seseorang dengan memanfaatkan pengolahan citra digital dapat dijadikan pembeda dari berbagai karakteristik dalam suatu objek gambar yang ada. Pencarian dan pembuatan *database* citra adalah salah satu komponen yang berperan sebagai informasi *visual*. Pengenalan citra dari bentuk dan posisi objek yang akan dideteksi merupakan hasil uji yang akan merepresentasikan seberapa cocok dengan input gambar terdeteksi dengan benar. Objek akan diberikan *noise* serta halangan atau *obstacle* contoh ada beberapa orang yang menutupi objek atau gedung dan pohon. Dalam penelitian ini menerapkan *template matching* merupakan sebuah teknik pengolahan citra *digital* untuk menemukan bagian kecil dari gambar yang sesuai dengan template gambar. Aplikasi ini dirancang menggunakan matlab 2019b sebagai software pembantu dan membuat database pengolahan citra.

Kata kunci: Template Matching, Sasaran Tembak

Abstract

Recognition from various points of view or points of view to detect the presence of a person by utilizing digital image processing can be used as a differentiator from various characteristics in an existing image object. Searching and creating an image database is one component that acts as visual information. Image recognition from the shape and position of the object to be detected is a test result that will represent how well it matches the detected image input correctly. Objects will be given noise and obstacles, for example there are several people covering objects or buildings and trees. In this study, applying template matching is a digital image processing technique to find a small part of the image that matches the image template. This application is designed using MATLAB 2019b as a supporting software and creates an image processing database.

Keywords: Matching Template, Shoot Target

1. Pendahuluan

Industri 4.0 mempunyai kecenderungan menuju kolaboratif dari berbagai aspek digital computation, teknologi informasi, dan saling bertukar informasi dalam hal teknologi digital. Hal ini dapat dimanfaatkan untuk meningkatkan pengamanan dan pertahanan kedaulatan NKRI TNI Angkatan darat di setiap daerah. Pemanfaatan teknologi ini dapat berupa pengolahan citra sebagai contoh untuk sasaran tembak musuh dan jarak sasaran tersebut. Tingkat keakurasian dapat ditingkatkan dengan adanya bantuan dari pengolahan citra dalam kesalahan pengumpulan data di lapangan penentuan suatu sasaran. Terdapat dua tipe metode pendeteksian sasaran dengan cara manual dan terkomputerisasi. Proses komputerisasi dilakukan dengan menentukan objek secara otomatis

berdasarkan informasi gambar yang diterima oleh pasukan pengintai [1].

Pendeteksian sasaran musuh pada satuan Artileri Medan memiliki masalah kinerja ketika terjadi miss prediction, yaitu keterbatasan prajurit Artileri Medan pada saat menentukan sasaran musuh dan jarak terhadap musuh sebagai informasi yang akan dilaporkan ke komando atas untuk dilakukan penembakan. Misalkan, ketika respon penentuan sasaran musuh yang lambat dan penentuan jarak tidak akurat, sehingga satuan penembak Artileri Medan tidak dapat menembakan alat sistem senjatanya dengan optimal. Beberapa masalah kinerja yang bersifat sewaktu-waktu muncul seperti *miss prediction* dan gangguan pandangan teropong untuk menentukan sasaran musuh yang selalu tinggi. Masalah peninjau dalam pengkoreksian

sasaran yang terjadi berulang kali dapat mengindikasikan bahwa peninjau sasaran sedang berada pada tahap tidak baik. Tujuan dari penelitian ini untuk memudahkan para prajurit armed dalam melaksanakan tugas latihan maupun pertempuran akan tetapi dalam pelaksanaannya masih kurang terhadap sasaran tembak musuh.

2. Metode

Pemrosesan awal mengusulkan kriteria komparasi gambar untuk membantu memahami dan membandingkan seluruh proses image processing dengan menggunakan metode *template matching*. Kriteria tersebut diantaranya yaitu :

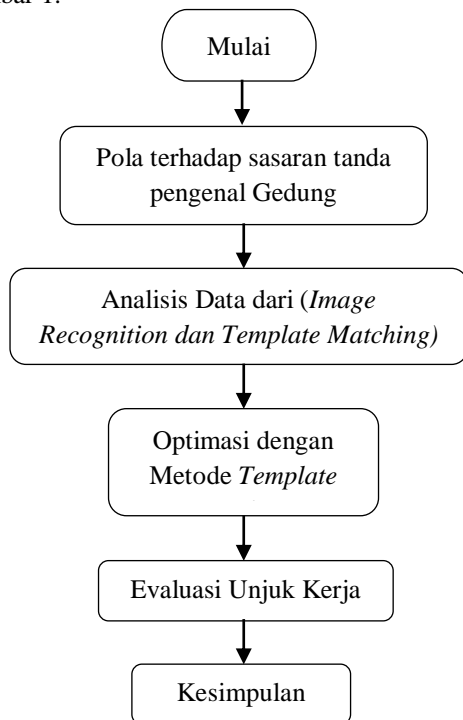
1. Mampu beradaptasi dengan perubahan gambar, yaitu perubahan pencahayaan atau kabut.
2. Dapat belajar dari pengalaman, yaitu pola *image recognition* dan *image processing*.

Metode penelitian memberikan gambaran, arah dan tahapan-tahapan yang akan dilakukan pada proses penelitian ini, sehingga proses yang dilakukan sesuai dengan tujuan dari penelitian.

Penelitian ini dibagi menjadi tiga kerangka konsep penelitian utama yaitu :

1. Proses *Image Processing*.
2. Proses *Template Matching*.
3. Hasil

Adapun alir tahapan kegiatan dari penelitian seperti pada Gambar 1.



Gambar 1. Bagan alir tahapan kegiatan penelitian

2.1. Image processing

Proses yang dilakukan dengan membandingkan kedekatan nilai jarak dari dua buah variabel yaitu antara citra uji dengan citra acuan untuk mencari nilai jarak terdekat.

Tahapan-tahapan pengolahan citra digital sebagai berikut:

2.1.1. Pre-Processing (Pengolahan)

Pada proses ini matriks citra uji akan disamakan ukurannya. Format file citra yang digunakan yaitu JPEG (.jpg), menurut [11] merupakan format yang sangat umum digunakan untuk transmisi dan menyimpan citra hasil kompresi. Selain itu dalam proses ini menghilangkan *noise* dan peningkatan citra gambar dilakukan oleh filter median.

2.1.1.1. Median Filter

Untuk menghilangkan *noise* menggunakan median filter, yang merupakan teknik *filtering digital non linier*. Median filter juga membantu menjaga tepi selama proses penghilangan *noise*.

$$\text{Median}[A(x) + B(x)] \neq \text{Median}[A(x)] + [B(x)]$$

2.1.1.2. Sobel Edge Detection

Matriks gradien *horizontal* dan *vertical* yang dimensinya 3X3 satu gambar menunjukkan respons vertikal dan gambar lainnya menunjukkan respons *horizontal* sedangkan nilai *threshold* digunakan untuk mendeteksi piksel tepi.

Tabel 1. Masks dari sebuah gambar

Gx			Gy		
+1	+2	+1	-1	0	+1
0	0	0	2	0	+2
1	2	1	-1	0	+1

Besarnya gradien dihitung dengan rumus :

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

Perkiraan besarnya dihitung dengan rumus :

$$|G| = \sqrt{G_x + G_y} \quad (2)$$

2.1.2. Segmentasi

Proses segmentasi adalah mempartisi citra digital menjadi beberapa segmen (yaitu kumpulan piksel) yang tersegmentasi. Konsep segmentasi adalah memodifikasi representasi suatu citra menjadi bermakna dan lebih mudah untuk dianalisis.

2.1.2.1. Region Growing

Metode ini bergantung pada piksel tetangga dengan nilai piksel yang sama di satu wilayah.

$$\sqrt{(R - R^*)^2 + (G - G^*)^2 + (B - B^*)^2} \leq d \quad (3)$$

Dimana: R, G, B adalah komponen warna dari piksel yang diuji, R*, G*, B* adalah komponen warna dari warna rata-rata daerah pembuatan dan d adalah parameter yang sangat penting untuk hasil segmentasi.

2.1.2.2. Histogram

Rentang data dibagi menjadi tempat sampah dengan ukuran yang sama untuk membangun histogram.

$$h_i = \frac{n_i}{n} \quad (4)$$

Dimana :

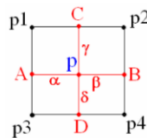
n_i = jumlah pixel yang memiliki derajat keabuan i

n = jumlah seluruh pixel di dalam citra

$i = 0, 1, \dots, L - 1$

2.1.3. Resize

Adalah penyamaan ukuran citra uji agar mempermudah dalam proses. Berdasarkan data yang ada semua diubah untuk disamakan dengan ukuran sesuai dengan citra acuan. Agar matriks citra uji yang dicocokkan dengan metode template matching akan sama dengan data acuan.



Gambar 2. Interpolasi bilinear

Titik $p_1, p_2, p_3,$ dan p_4 pada Gambar 4.8 adalah pixel yang diketahui warnanya sedangkan p adalah pixel yang ingin dicari warnanya dengan pendekatan interpolasi bilinear. Diambil titik A, B, C, dan D yang merupakan proyeksi titik p ke garis p_1 - p_3, p_2 - p_4, p_1 - $p_2,$ dan p_3 - p_4 . Didefinisikan $\alpha, \beta, \gamma,$ dan δ sebagai berikut :

$$\alpha = \frac{|AP|}{|AB|}, \beta = \frac{|BP|}{|AB|}, \gamma = \frac{|CP|}{|CD|}, \delta = \frac{|DP|}{|CD|} \quad (6)$$

$|AB|$ merupakan panjang AB, hal yang sama juga berlaku terhadap $|AP|,$ dan lain-lain. Jarak antara dua buah pixel terdekat dianggap satu satuan. Dalam hal ini juga berlaku

$$\alpha + \beta = \gamma + \delta = 1. \quad (7)$$

2.1.4. Grayscale

Pada proses ini semua citra uji diubah menjadi warna keabuan, agar mempermudah untuk tahap selanjutnya. Grayscale sendiri merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, atau dengan kata lain bagian Red = Green = Blue dalam suatu citra merupakan intensitas, pada tahapan ini masing-masing piksel RGB (red green blue) dari citra diambil nilainya, dihitung rata-ratanya dari ketiga nilai RGB [11]. Konversi dapat dilakukan dengan mengambil rata-rata dari nilai r, g dan b sehingga dapat dituliskan menjadi:

$$S = \frac{r+g+b}{3} \quad (8)$$

2.1.5. Thresholding

Thresholding adalah proses yang digunakan untuk menghasilkan citra biner, yaitu citra yang memiliki hanya dua warna hitam dan putih. Nilai pixel yang awalnya lebih kecil diinisialisasi dengan nilai 1, sedangkan yang besar berupa background diinisialisasi dengan nilai 0 [11].

Proses thresholding ini pada dasarnya adalah proses perubahan kuantisasi pada citra, sehingga untuk melakukan thresholding dengan derajat keabuan dapat digunakan rumus:

$$x = b. \text{int} \left(\frac{w}{b} \right) \quad (5)$$

Dimana :

w = nilai derajat keabuan sebelum thresholding

x = nilai derajat keabuan setelah thresholding

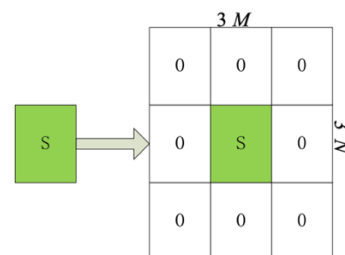
2.2. Template Matching

Persiapan Proses Object Recognition untuk dapat mengenali objek diperlukan suatu metode pengolahan citra. Pada penelitian ini menggunakan metode Template Matching yaitu metode pencarian kedekatan nilai jarak dari dua buah variabel. Penelitian ini merupakan pengembangan dan berfokus pada citra, pengenalan objek dan keluaran yang telah kaliberasi. Untuk dapat melakukan hal-hal tersebut memerlukan perangkat yang mendukung pengakuisisian citra, pengolahan citra dan komputasi, selain itu membutuhkan software Matlab R2019b.

Dalam pencocokan gambar *template*, perlu untuk menghitung jumlah pixel dan jumlah kuadrat dari setiap sub-matriks dari pencocokan gambar sasaran secara *real-time*. Berikut ini, kita akan mengambil matriks jumlah kumulatif dari gambar sasaran sebagai contoh untuk menggambarkan metode perhitungan.

2.2.1. Matrix Extension

Luas gambar dari gambar sasaran (S) yang akan dicocokkan dengan ukuran T yang sama adalah diekspansi menjadi matriks K $3M \times 3N$. Untuk kenyamanan, area gambar yang dipilih masih ditandai sebagai S. Metode ekspansi adalah dengan menempatkan S di tengah matriks K, dan mengatur semua nilai sekitar 8 daerah menjadi nol, seperti yang ditunjukkan pada gambar 3.



Gambar 3. Matriks ekspansi dari pencocokan gambar S

2.2.2. Perhitungan pada matriks K :

Langkah 1. Perhitungan matriks K1: Mulai dari baris kedua K, elemen setiap baris ditambahkan ke elemen kolom yang sama pada baris berikutnya untuk memperoleh:

$$K1(i,j) = K(i-1,j) + K(i,j)$$

Langkah 2. Perhitungan matriks K2: Dari baris pertama K1 sampai akhir baris 2N, dihitung sebagai berikut:

$$K2(i,j) = K1(i+1,j) - K1(i,j)$$

Langkah 3. Perhitungan matriks K3: Untuk K2, mulai dari kolom kedua, ditambahkan elemen setiap kolom dengan kolom sebelumnya dari baris yang sama sehingga diperoleh:

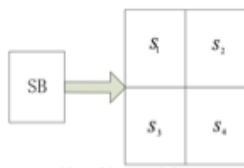
$$K3(i,j) = K2(i,j-1) + K2(i,j)$$

Langkah 4. Perhitungan matriks K4: Dari kolom pertama K3 sampai akhir kolom 2M, dihitung sebagai berikut:

$$K4(i,j) = K3(i,j+Mb) - K3(i,j)$$

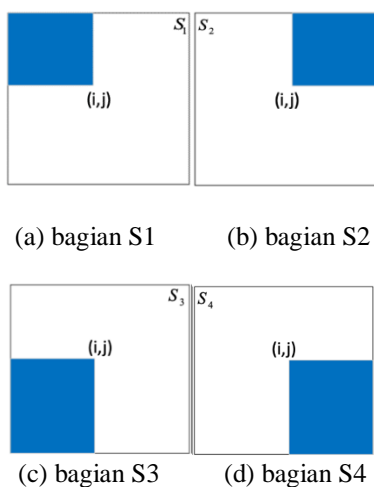
2.2.3. Matriks jumlah kumulatif SB

Matriks dari jumlah kumulatif SB menjadi sub-matriks dengan ukuran 2M*2N di sudut kiri atas K4, seperti yang ditunjukkan pada gambar 4.



Gambar 4. Komposisi matriks penjumlahan kumulatif SB

Adapun setiap titik (i, j), S1 (i,j), S2(i,j), S3(i,j) dan S4(i,j), mewakili jumlah beberapa elemen dalam sub-matriks S1,S2,S3 dan S4 masing-masing, seperti yang ditunjukkan pada bagian yang diarsir dari gambar 5.



Gambar 5. Diagram matriks penjumlahan kumulatif

Nilai S135(i,j), S2(i,j), S3(i,j) dan S4(i,j) memiliki hubungan yang sesuai dengan jumlah kumulatif di bawah ini :

$$\sum_{i,j} f(u+i, v+j) = \begin{cases} S1(u+m, v+n), u < 0, v < 0 \\ S2(u+m, v), u < 0, v \geq 0 \\ S3(u, v+n), u \geq 0, v < 0 \\ S4(u, v), u \geq 0, v \geq 0 \end{cases}$$

2.2.4. Perhitungan similarity dalam image matching

Perhitungan pencocokan gambar kesamaan adalah sebagai berikut:

Koefisien pencocokan γ dihitung sebagai berikut:

$$\gamma = \sqrt{(m * n - 1) * \text{std}(T)}$$

Dimana std(T) adalah varians dari semua elemen dalam gambar template T

Menghitung penyebut dari ukuran korelasi:

$$\alpha(i,j) = \gamma * \sqrt{\max(2SB(i,j) - \frac{SB(i,j)^2}{M*N}, 0)} \quad (9)$$

Menghitung molekular dari ukuran korelasi:

$$\beta(i,j) = r(i,j) - \frac{SB(i,j) * \text{sum}(T)}{m * n} \quad (9)$$

Dimana jumlah(T) adalah jumlah semua elemen dalam gambar templat T, dan r(i,j) adalah nilai suku yang bersesuaian dalam matriks konvolusi r.

Dalam kesamaan pencocokan template akhir matriks C, nilai setiap item dihitung sebagai berikut:

$$c(i,j) = \frac{\beta(i,j)}{\alpha(i,j)} \quad (10)$$

Derajat kemiripan dua citra adalah nilai elemen maksimum dari matriks C, dan jika koordinat maksimum adalah (im,jm), maka offset relatif setelah pencocokan citra adalah (im-m,jm-n).

Agar penelitian ini dapat selesai sesuai dengan tujuan penelitian maka peneliti menyusun langkah-langkah penelitian sebagai berikut:

2.2.4.1. Mengumpulkan Data

Pada tahap ini peneliti mengumpulkan data citra foto dalam kondisi terang dengan berbagai posisi objek sasaran area gedung arah depan, ke samping, terlipat arah ke depan, terlipat ke samping, terlipat dan terhalang arah ke depan, terlipat dan terhalang ke samping dengan background gedung, jarak pemasangan kamera dengan objek yang diawasi 10 meter. Masing – masing kondisi berisi 3 foto.jpeg yang berbeda dengan resolusi gambar 1920 x 1080 pixel.

2.2.4.2. Analisis Data

Setelah data citra gambar yang merepresentasikan 6 kondisi sudah terkumpul maka selanjutnya peneliti melihat

ukuran sasaran area dan objek gedung pada saat lurus arah depan, ke samping, terlipat dan terhalang dengan blob analysis pada matlab.

2.2.4.3. Merancang Template Matching

Setelah diketahui besar pixel pada area dari setiap objek sasaran Gedung pada saat lurus arah depan, ke samping, terlipat dan terhalang maka langkah selanjutnya peneliti merancang *template* untuk 6 kondisi.

2.2.4.4. Merancang Background Subtraction

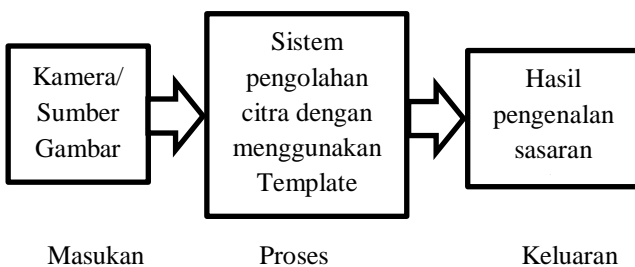
Untuk merancang background subtraction langkah-langkah yang akan dilakukan peneliti sama seperti yang dilakukan seperti ekstraksi foto, normalisasi citra, *background subtraction*, operasi morfologi, deteksi.

2.2.4.5. Pengujian Deteksi Objek Sasaran Dengan Template Matching

Pada penelitian ini pengujian yang dilakukan untuk mendeteksi sasaran menggunakan *template matching* dan *background subtraction* untuk masing – masing 6 kondisi.

2.3. Diagram Blok Sistem

Pada gambar 6 diperlihatkan diagram blok dari sistem pendeteksian gedung sasaran musuh. Sistem ini terdiri dari 6 blok bagian. Dalam blok pertama merupakan gambar sumber yang ingin dideteksi. Blok kedua merupakan blok proses dimana proses pengolahan citra dilakukan menggunakan Matlab R2019 sebagai aplikasi untuk mengolah citra yang sebelumnya sudah di siapkan. Blok bagian terakhir yang merupakan bagian terakhir merupakan blok *output* menggunakan PC sebagai media interface antara user dengan sistem. output yang didapat dari sistem ini adalah hasil deteksi berupa frame berbentuk kotak yang mendeteksi gedung sasaran musuh.



Gambar 6. Gambaran umum sistem

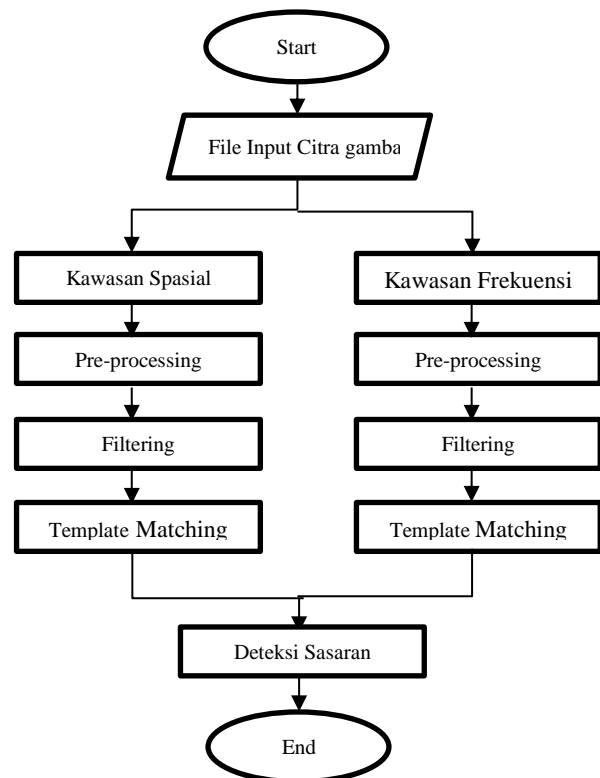
2.4. Algoritma Sistem

Aplikasi pendeteksian gedung sasaran musuh ini dibuat menggunakan software Matlab R2019. Gambar 7 memperlihatkan proses dari sistem tersebut. Proses dimulai dengan memasukkan citra gedung yang telah diambil pada bagian input dengan menggunakan kamera dengan resolusi

baik ke dalam aplikasi yang telah dibuat. Selanjutnya, algoritma dari sistem dengan menggunakan metode *template matching* yang telah dibuat adalah sebagai berikut :

1. Input gambar.
2. *Pre-processing* Citra Digital
 - a. *Cropping* bagian gedung yang ingin dideteksi
 - b. Ubah citra RGB menjadi Citra *Grayscale*
 - c. Ubah citra *Grayscale* menjadi citra biner
 - d. Menghapus bagian citra biner yang bukan karakter
3. Proses Deteksi
 - a. Hasil *pre-processing* yang sudah dilakukan konversi
 - b. *Resize* citra target dengan ukuran citra lebih kecil untuk dibandingkan dengan gambar asli pada proses *template matching*.
 - c. Sistem akan membandingkan citra hasil *resize* gedung target dengan sumber gambar.
 - d. Membuat *Bounding box* dari target gedung berdasarkan nilai korelasi yang paling besar
4. Gedung Sasaran teridentifikasi
5. Sistem akan melakukan deteksi dengan tanda *bounding box* pada aplikasi.

Algoritma sistem yang dibuat membaca gambar yang sudah di capture dan dikirim ke bagian *pre-processing* untuk diproses. Tugas utama dari bagian *pre-processing* adalah untuk mengurangi gangguan dari citra dan juga menyesuaikan citra hasil dengan citra pada sumber gambar agar dapat diproses dengan algoritma *template matching*.



Gambar 7. Flowchart Proses

2.5. Proses Filter Domain Spasial

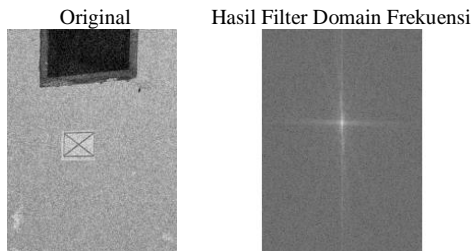
Proses filter spasial merupakan proses melewatkan atau menghambat suatu frekuensi dari komponen dalam hal ini yang dimaksud adalah komponen spasial. Filter *domain* spasial akan melakukan proses manipulasi kumpulan pixel dari sebuah citra untuk menghasilkan citra baru. Pada gambar 8 menunjukkan hasil proses dari filter menggunakan *domain* spasial.



Gambar 8. Filter *Domain Spasial*

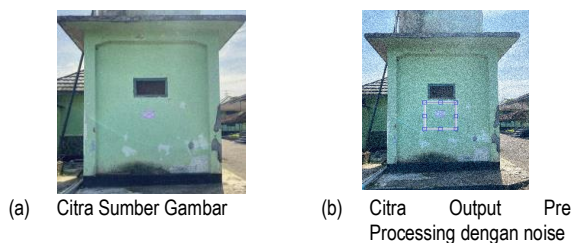
2.6. Proses Filter Domain Frekuensi

Pada gambar 9 menunjukkan hasil dari filter menggunakan domain frekuensi dimana tahapan proses dimulai dari input citra digital untuk gedung diinput dengan format gambar JPEG / PNG. Lalu kemudian dilakukan proses transformasi fourier dari citra input menggunakan FFT. Terakhir lakukan proses invers transformasi fourier menggunakan *invers FFT* untuk memperoleh hasil citra dengan filter domain Frekuensi



Gambar 9. Filter *Domain Frekuensi*

Akurasi dari pendeteksian sistem ini dipengaruhi oleh keberadaan noise pada citra yang akan dideteksi. Semakin tinggi *noise*, maka akurasi pendeteksian akan berkurang dan sebaliknya jika semakin kecil *noise* maka akurasi pendeteksian akan semakin besar. Dalam bagian *pre processing noise* dari citra akan dikurangi dengan adanya bantuan filtering gambar.



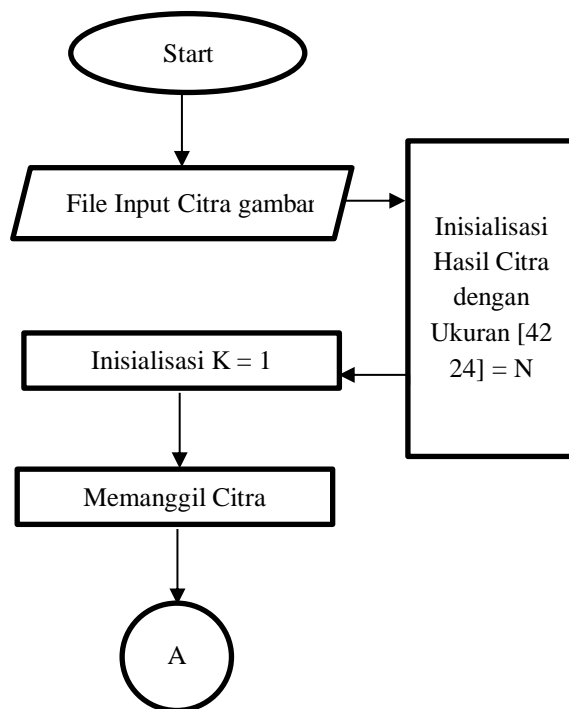
Gambar 10. Proses Citra sebelum dan sesudah

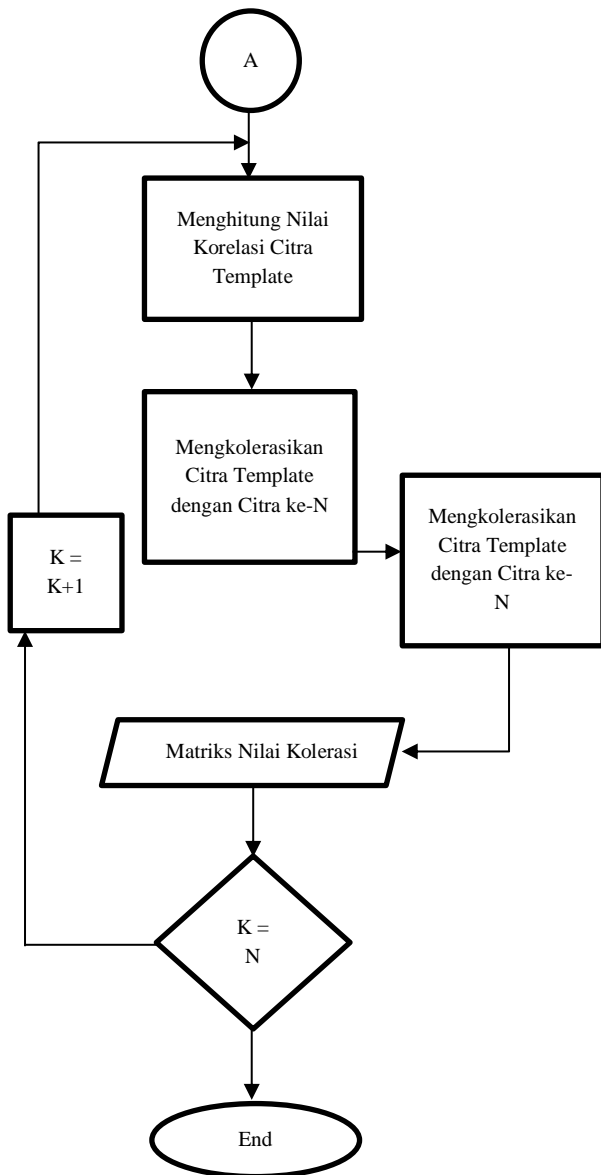
Gambar 10 memperlihatkan citra sebelum dan sesudah diproses pada bagian *preprocessing*. Sehingga sistem dengan template matching hanya melakukan proses deteksi pada objek itu saja. Citra dari object gedung diperoleh melalui proses deteksi objek yang sebelumnya sudah ditentukan. Deteksi target gedung yang dilakukan oleh sistem dilakukan pada proses *template matching* ditunjukkan pada gambar 9.



Gambar 11. Hasil Deteksi Gedung

Ada beberapa kriteria yang perlu diperhatikan agar proses *template matching* dapat berjalan optimal, hal tersebut adalah ukuran citra target dengan ukuran citra sumber gambar yang akan dibandingkan harus sama. Hal ini dikarenakan proses yang terjadi pada algoritma template matching adalah membandingkan dua buah citra pada pixel yang berada pada koordinat yang sama. Gambar 9 merupakan flowchat proses *template matching* yang terjadi pada sistem.





Gambar 12. Flowchart *Template Matching*

Pada dasarnya proses template matching adalah membandingkan nilai pixel dari citra gambar dalam basis data template. Citra target dibandingkan dengan seluruh citra dalam basis data template. Output dari proses template matching adalah munculnya *bounding box* pada gedung target sasaran.

3. Hasil dan Analisa

Citra uji yang digunakan sebanyak 60 citra uji yang terdiri dari 6 kondisi area gedung sebagai sasaran area pada saat lurus arah ke depan, ke samping, terlipat dan terhalang dengan pola gambar silang berwarna merah sebagai objek sasaran.

Selanjutnya data citra uji tersebut diolah menggunakan algoritma template matching. Dalam penelitian ini,

background subtraction akan memisahkan background dengan pola tanda pengenalan sasaran.

3.1. Proses pengambilan gambar

Pengambilan gambar merupakan tahapan awal yang harus dilakukan dalam penelitian ini. Adapun sampel uji yang digunakan adalah objek sasaran yang terdiri dari 4 kondisi area sasaran. Sampel uji ini terdiri dari sasaran area pada saat lurus arah ke depan, ke samping, terlipat dan terhalang, 15 buah sasaran objek lurus arah kedepan, 15 buah sasaran objek lurus arah samping, 15 buah sasaran objek lurus arah terlipat dan 15 buah sasaran objek lurus arah terlipat dan terhalang obstacle. Citra yang diperoleh formatnya dalam bentuk JPG atau PNG. Adapun contoh sampel dari sasaran sebuah Gedung ditunjukkan pada gambar 13.



Gambar 13. Contoh sasaran pada Gedung

3.2. Template Matching

Pada tahapan ini dilakukan pencocokan antara template dan image (gambar) pada metode template matching dengan melakukan perhitungan rata-ratanya. Batas untuk nilai pada metode ini adalah 1 sampai -1 hasil yang paling cocok maka memiliki nilai 1, dan sebaliknya. Untuk menentukan letak template pada sebuah image dilakukan dengan perhitungan menggunakan persamaan di bawah ini :

$$R(x', y') = \sum (T'(x', y') \cdot I'(x + x', y', y + y')) \quad (11)$$

dimana :

T' = template

I' = gambar yang dicocokkan

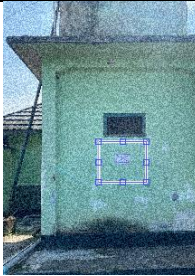




(x, y) = koordinat gambar yang dicocokkan


(x', y') = koordinat template

3.3. Pengujian terhadap pendeteksian target

Dalam bagian ini terdapat pengujian performansi deteksi gambar menggunakan template matching dengan berbagai serangkaian kondisi tertentu yang dapat mempengaruhi keefektifan kinerja sistem pendeteksian target gedung tersebut. Untuk menguji performa deteksi ini gambar data yang telah diinputkan digunakan Kembali untuk pengujian lain. Dapat dilihat pada Tabel 2 skema pengujian performansi.

Tabel 2. Uji performansi deteksi target

Pengujian 1 dengan 15 kali iterasi				
No	Posisi target gambar	Terdeteksi	Tidak Terdeteksi	Presentase Keberhasilan
1	 Lawan tampak depan	10	5	66,67%
Pengujian 2 dengan 15 kali iterasi				
2	 Lawan tampak samping	11	4	73,33%
Pengujian 3 dengan 15 kali iterasi target terlipat				
3	 Lawan tampak depan terlipat	10	5	66,67%
Pengujian 4 dengan 15 kali iterasi target menyamping dan terlipat				
4	 Lawan tampak samping terlipat	9	6	60%
Pengujian 5 dengan 15 kali iterasi target terlipat dan terhalang				
5	 Lawan tampak depan terlipat + obstacle	12	3	80%

Pengujian 6 dengan 15 kali iterasi target menyamping terlipat dan terhalang				
6	 Lawan tampak samping terlipat + obstacle	11	4	66,67%

Pengujian telah dilakukan dengan berbagai kondisi dan disudut tampak depan untuk mendeteksi target pada gambar secara keseluruhan. adapun cara untuk mengukur persentase keberhasilan yaitu menggunakan rumus persamaan (2) berikut :

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah berhasil}}{\text{Jumlah Percobaan}} \times 100\%$$

Dalam pengujian 15 kali iterasi dengan 2 sesi yang berbeda menghasilkan target lawan tampak depan dengan gambar yang diberikan noise ada beberapa perbedaan.

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Berhasil}}{\text{Jumlah Percobaan}} \times 100\%$$

$$\text{Persentase Keberhasilan} = \frac{10}{15} \times 100\%$$

$$\text{Persentase Keberhasilan} = 66,7 \%$$

Berdasarkan persamaan (2), Pada sesi 1 pada gambar tersebut presentase keberhasilan sekitar 66,7%. Untuk sesi yang kedua gambar lawan tampak depan yang tidak diberikan *noise* menghasilkan presentasi keberhasilan sekitar 73,3% hal ini dikarenakan pemberian *noise* pada citra sebelum mendeteksi target mempengaruhi hasil presentasi keberhasilan.

Pengujian untuk gambar yang tidak diberikan noise akan terlihat lebih jelas dan memiliki presentase keberhasilan lebih tinggi. Hal ini ditunjukkan dengan hasil tabel 4 dimana dengan dilakukannya 15 kali iterasi dan 2 kali sesi yang berbeda tetap memiliki angka presentase keberhasilan yang sama dan lebih tinggi dari sebelumnya (gambar yang telah diberikan *noise*).

4. Kesimpulan

Dari hasil pengujian performansi dengan metode template matching untuk mendeteksi target lawan yang telah dilakukan dengan penambahan gangguan berupa noise, obstacle, dan terlipat memiliki hasil presentase keberhasilan sekitar 70% untuk gambar yang memiliki gangguan berupa noise kemudian untuk gambar yang diberikan obstacle dan target terlipat memiliki presentase

keberhasilan sekitar 80% dengan 30 kali percobaan. Hal ini merupakan masih dapat digunakannya metode template matching untuk mendeteksi target atau objek tertentu.

Referensi

- [1]. Prema T. Akkasaligar & Sumangala Biradar (2020). Selective medical image encryption using DNA cryptography. *Information Security Journal: A Global Perspective*.
- [2]. Krishnamurthy A, Joshua V, Haj Hensvold A, et al. (2016). Identification of a novel chemokine-dependent molecular mechanism underlying rheumatoid arthritis-associated autoantibody-mediated bone loss. *Ann Rheum Dis*. 2016 Apr;75(4):721-9. Epub 2015 Nov 26.
- [3]. Sutoyo Edi Mulyanto. (2016). *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- [4]. Fuad, M. & Suciati, Nanik. (2018). Klasifikasi Multilabel Motif Citra Batik Menggunakan Boosted Random Ferns. *JUTI: Jurnal Ilmiah Teknologi Informasi*. 16. 79. 10.12962/j24068535.v16i1.a673.
- [5]. F. Nakul, R. Pardede, B. Budiana, R. Mahdaliza, and H. Wijanarko. (2021). Sistem Inspeksi Label Produk Menggunakan Metode Golden Template Comparison dan Konveyor Pemilah. *JAEE*, vol. 5, no. 1, pp. 20-25.
- [6]. Amilia Khoiro Masruri dan Budi Setiyono. (2014). Penggunaan Metode Template Matching Untuk Mendeteksi Cacat Pada Produksi Peluru. Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS).
- [7]. R. Rismanto, A. Prasetyo, and D. Irawati. (2020). Optimalisasi Image Thresholding pada Optical Character Recognition Pada Sistem Digitalisasi dan Pencarian Dokumen.
- [8]. Putra, Darma. (2010). *Pengolahan Citra Digital*. Yogyakarta: CV. ANDI OFFSET.
- [9]. N, Siko Andriyanto, dan R, Bhetta Pandu. (2010). *Pengolahan Citra Program Treshold dengan metode Otsu*.
- [10]. Khairi Ibnutama, Zaimah Panjaitan, dan Erika Fahmi Ginting. (2019). Modifikasi Metode Template Matching pada OCR Untuk Meningkatkan Akurasi Deteksi Plat Nomor Kendaraan.
- [11]. V. Ramya, M. Kokila Sundari, dan T. Akilan. (2014). Product Label Detection and Recognition System using SIFT and Line Detection Algorithm.