

PERANCANGAN SISTEM PENDETEKSI *NATURAL LANDMARK* SEBAGAI ALAT BANTU PENGENALAN POSISI PADA ROBOT BERBASIS PENGOLAHAN CITRA DIGITAL

DESIGNING OF NATURAL LANDMARK DETECTION SYSTEM AS AN AID FOR ROBOT BASED ON IMAGE PROCESSING

Gani Gamayudha Sofyan¹

¹Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
¹gani.gamayudha@gmail.com

Abstrak

Robot dibutuhkan untuk mempermudah pekerjaan manusia terutama untuk pelayanan di kantor dan rumah. Dalam pengembangannya, *robot vision* telah menjadi ide untuk beberapa riset, karena para pengembang menginginkan robot yang dapat “melihat”. Dengan penglihatan, robot melakukan berbagai macam hal seperti navigasi otomatis, *localization*, *automatic map construction* dan lain – lain. *Localization* atau pengenalan posisi dapat dilakukan salah satunya dengan cara mengenali suatu *landmark* yang ada di sekitarnya. Pada tugas akhir ini dirancang sistem pendeteksi *natural landmark* pada robot dengan menggunakan pengolahan citra digital. Sistem yang dirancang adalah deteksi *landmark* secara berurutan. Alat yang digunakan adalah sebuah kamera yang disimpan diatas robot beroda. Robot akan bergerak dengan cara dikontrol oleh manusia. Metode – metode yang digunakan pada sistem ini adalah: *grayscale*, *Gaussian Pyramid* dan *Zero Padding* untuk tahap *pre-processing*; *2D FFT* untuk tahap ekstraksi; dan *Normalized Cross Correlation* (NCC) untuk tahap *Pattern Matching*. Sistem ini memiliki keluaran berupa posisi (daerah *landmark* 1, 2, 3, dan 4). Pengujian sistem dilakukan secara *nonreal-time* pada 2 skenario yang berbeda. Akurasi terbaik untuk mendeteksi *landmark* tersebut sebesar 99.67% diperoleh pada saat *threshold* bernilai 0.99. *Landmark* kedua adalah *landmark* yang paling mudah untuk dideteksi dengan nilai akurasi sebesar 100%. Sistem hanya membutuhkan waktu sekitar 6ms untuk melakukan pendeteksian *landmark*.

Kata kunci : robot, deteksi, posisi, pengolahan citra digital, *landmark*, *normalized cross correlation*

Abstract

Robots are needed to facilitate human work, especially for office and home services. In its development, robot vision has become an idea for some research, because developers want a robot that can "see". With vision, robots can do things like automatic navigation, localization, automatic map construction and many more. Localization or position recognition can be done one of them by recognizing a landmark that's in the vicinity. This final project designed natural landmark detection system on robot by using digital image processing. Designed system is the sequential landmark detection. The tool used is a camera that is stored on a wheeled robot. Robot move in a way controlled by human. This system used methods such as: *grayscale*, *Gaussian Pyramid* and *Zero Padding* for pre processing; *2D FFT* for feature extraction; and *Normalized Cross Correlation* (NCC) for *Pattern Matching*. This system has a position output (*landmark* areas 1, 2, 3, and 4). System testing is performed nonreal-time in 2 different scenarios. The best accuracy to detect the landmark is 99.67% obtained when threshold value is 0.99. The second landmark is the easiest landmark to detect with an accuracy value of 100%. System only needs about 6ms to detect a landmark.

Keywords: robot, detection, position, image processing, landmark, normalized cross correlation.

1. Pendahuluan

Pada saat ini, robot semakin dikembangkan, terutama untuk pelayanan di kantor dan rumah karena robot dapat melakukan berbagai macam kegiatan manusia. Persaingan global dan upaya meningkatkan efisiensi membuat para peneliti mengembangkan aplikasi yang tidak dimiliki robot – robot sebelumnya. Aplikasi – aplikasi ini membutuhkan robot yang dapat bergerak dan melakukan tugas secara bersamaan. Robot yang memiliki sistem kecerdasan buatan (*Artificial Intelligence*) dianggap lebih baik dari robot yang hanya memiliki sistem kontrol karena

dapat melakukan pergerakan dan pekerjaan secara otomatis tanpa harus dikendalikan langsung oleh manusia. Salah satu contohnya adalah Egemin Automation yang dapat mengantarkan barang dalam suatu gedung^[10].

Sistem kecerdasan buatan diciptakan dan dimasukkan ke dalam mesin agar dapat melakukan pekerjaan seperti yang dapat dilakukan oleh manusia. Beberapa alat yang dapat digunakan sebagai alat bantu navigasi robot adalah sensor elektronik, kamera, dan *Global Positioning System* (GPS). Navigasi dengan sensor elektronik adalah yang paling populer saat ini, akan tetapi navigasi dengan menggunakan sensor elektronik biasanya membutuhkan banyak sensor yang dipasang pada beberapa bagian robot agar dapat bergerak dengan baik. Selain itu, butuh berbagai macam sensor agar dapat melakukan pekerjaan yang berbeda^[10].

Pada akhirnya, penulis memutuskan untuk melakukan penelitian *localization* robot menggunakan kamera. Kamera yang ditanam pada robot dianalogikan sebagai mata pada manusia, kamera akan mengenali *data image* tertentu yang menonjol, *image* tersebut akan digunakan untuk menentukan posisi robot. Robot akan mengenali posisi tersebut dan memberikan informasi dimana posisi robot itu berada. Dalam pengembangannya, kamera ini tidak hanya digunakan oleh robot untuk mengetahui posisi, tapi juga untuk berbagai macam aplikasi lainnya, seperti membaca, mendeteksi jenis benda, dan lain sebagainya.

2. Dasar Teori dan Metodologi

2.1 Robot^[3]

Robot adalah peralatan manipulator yang mampu diprogram, mempunyai berbagai fungsi, yang dirancang untuk memindahkan barang, komponen-komponen, peralatan, atau alat-alat khusus, melalui berbagai gerakan terprogram untuk pelaksanaan berbagai pekerjaan. Secara mendasar, robot memiliki banyak hal yang sama dengan otomasi internal, mereka memanfaatkan piranti tenaga yang serupa (seperti listrik, hidraulik, atau *pneumatic*) dan mereka dikendalikan melalui urutan-urutan yang telah dikendalikan melalui program, yang memungkinkan mesin tersebut pada posisi yang diinginkan.

Navigasi dapat dideskripsikan sebagai proses penentuan jalur yang cocok dan sesuai antara titik awal dan akhir untuk sebuah robot berjalan^[1]. Navigasi secara visual merupakan cara yang bagus untuk sensor robot dikarenakan kemampuannya untuk mendeteksi dan mengenali ciri dengan warna dan ukuran apapun^[6].

2.2 Citra Digital^[2]

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital^[6]. Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi, $f(x,y)$, dimana x dan y adalah koordinat spasial, dan amplitudo f pada setiap pasang koordinat (x,y) disebut intensitas atau tingkat keabuan dari citra pada titik tersebut. Ketika x,y , dan nilai-nilai amplitudo f itu terbatas, kita dapat menyebutnya sebagai citra digital.

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk monitor komputer, nilai rentangna paling kecil = 0 dan paling besar 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer, dan citra berwarna biasanya memiliki 24 bit (R = 8 bit, G = 8 bit, B = 8 bit). Dengan begitu, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 16.777.216$ jenis warna.

Format warna *grayscale* dibentuk oleh komponen luma pada citra RGB. Berdasarkan Recommendation ITU-R BT.601-7, konstruksi komponen luma adalah sebagai berikut:

$$E'_y = 0.2989 * E'_R + 0.5870 * E'_G + 0.1140 * E'_B, \quad (2.1)$$

Dimana E'_R adalah nilai intensitas pixel pada layer *red*, E'_G adalah nilai intensitas pixel pada layer *green*, dan E'_B adalah nilai intensitas pixel pada layer *blue*.

2.3 Transformasi Fourier

Suatu fungsi dengan periode tertentu dapat dinyatakan dalam deret Fourier. Tetapi, bagaimana dengan fungsi yang memiliki periode tak berhingga atau dengan kata lain tidak periodik. Fungsi tersebut dapat dianggap sebagai fungsi periodik dengan periode tak berhingga dan mengganti penjumlahan pada deret Fourier dengan integral. Metode ini disebut Transformasi Fourier. Sebagai contoh, kita dapat menganalisa sinyal seperti bunyi yang pada awalnya

merupakan fungsi waktu, diubah sebagai fungsi frekuensi dengan memanfaatkan transformasi Fourier. Kita kemudian dapat melihat periodisitas sinyal tersebut setelah sinyal tersebut ditransformasi.

Jika kita memiliki fungsi $f(x)$, transformasi Fourier dari fungsi tersebut adalah:

$$g(k) = \int_{-\infty}^{\infty} f(x)e^{-ikx} dx \tag{2.2}$$

$$f(x) = \int_{-\infty}^{\infty} \frac{g(k)}{2\pi} e^{-ikx} \tag{2.3}$$

2.4 Gaussian Pyramid

Gaussian Pyramid (GP) adalah suatu metode dalam pemrosesan gambar untuk men-*downsampling* dan meng-*upsampling*. GP ini pertama kali diperkenalkan oleh Burt dan Adelson pada tahun 1983. Tujuan dari GP ini adalah membentuk suatu representasi sinyal (sebagai contoh adalah gambar jika dalam 2 dimensi) yang berbasis pada suatu filter yang berfungsi untuk mendekomposisi gambar-gambar kedalam suatu informasi pada beberapa skala, untuk mengekstrak suatu ciri atau struktur pada suatu area pengamatan, dan untuk memperkuat *noise*. Fungsi dari GP ini salah satunya adalah untuk mereduksi *redundancy* dan memodelkan suatu gambar untuk efisiensi coding, *image enhancement/restoration*, dan untuk *image analysis/synthesis*.

2.5 Korelasi Silang

Korelasi silang atau *cross correlation* adalah suatu metode untuk mendeteksi tingkat korelasi antara dua buah sinyal dalam domain spasial. Cara menghitungnya yaitu hanya dengan mengalikan lalu menjumlahkan kedua sinyal tersebut.

Persamaan untuk menghitung korelasi silang:

$$corr(x, y) = \sum_{n=0}^{n-1} x[n] * y[n], \tag{2.4}$$

dimana x dan y adalah sinyal yang dideteksi tingkat korelasinya. Setelah diperoleh nilai dari persamaan tersebut, kedua sinyal dapat ditentukan apakah berkorelasi atau tidak dengan melihat nilai korelasi silang.

Nilai yang diperoleh dari korelasi silang harus dinormalisasi. Berikut adalah tiga masalah yang ada pada korelasi silang:

1. Sulit untuk memahami nilai yang diperoleh dari korelasi silang.
2. Kedua *metrics* harus memiliki *energy* yang sama. Jika grafik B adalah salinan yang sama dari grafik A namun memiliki ukuran yang dua kali lebih kecil, maka korelasi tidak akan terdeteksi.
3. Berdasarkan rumusnya, nilai 0 tidak akan diperhitungkan. Sebagai contoh $0 * 0 = 0$ dan $0 * 200 = 0$, maka penilaian tidak akan diperhitungkan.

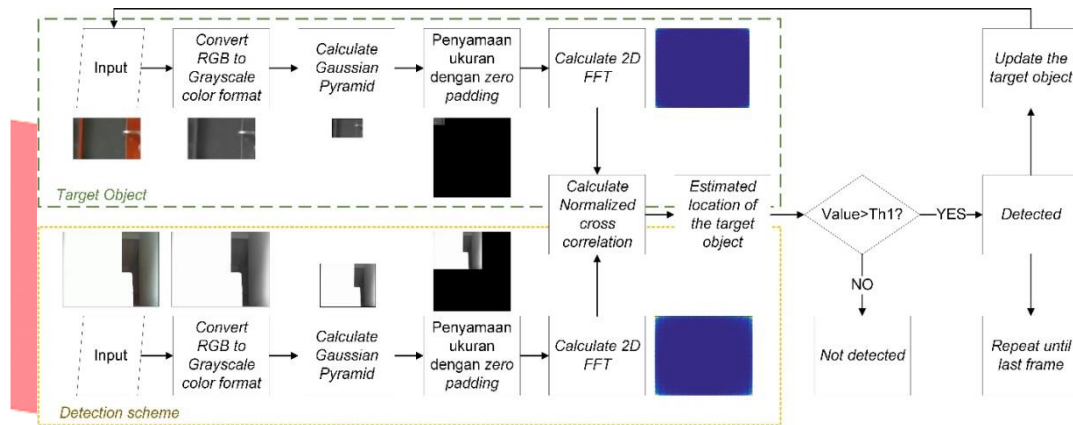
Persamaannya yaitu sebagai berikut:

$$norm_corr = \frac{\sum_{n=0}^{n-1} x[n]*y[n]}{\sqrt{\sum_{n=0}^{n-1} x[n]^2 + \sum_{n=0}^{n-1} y[n]^2}} \tag{2.5}$$

Nilai hasil dari normalisasi korelasi silang akan berupa angka dari -1 hingga 1. Jika nilai tersebut mendekati 1 maka kedua sinyal tersebut berkorelasi, sebaliknya jika angka tersebut mendekati -1 maka kedua sinyal tersebut tidak berkorelasi. Dengan menggunakan persamaan (2.5), berikut contoh proses normalisasi korelasi silang dengan meneruskan ilustrasi pada proses korelasi silang:

2.6 Desain Model Sistem

Pemodelan sistem mengenai deteksi *landmark* dibuat dalam tugas akhir ini. Sistem tersebut memiliki masukan berupa *frame-frame* dari video yang telah direkam menggunakan kamera yang diletakan pada suatu robot. Model sistem pada tugas akhir ini memiliki 2 skema utama, yaitu: skema untuk *target object* dan skema untuk bagian deteksi. Dua skema tersebut memiliki langkah yang sama dan secara umum tugas akhir ini memiliki kerangka kerja sebagai berikut:



Gambar 1. Skema Sistem

a) Convert RGB to Grayscale color format

Pada tahap ini gambar masukan diubah format warnanya, dari format warna *red, green, blue* (RGB) menjadi format warna pada skala keabuan. Tujuan dari perubahan format warna ini adalah agar dalam proses *downsampling* menggunakan GP berlangsung secara sederhana (1 layer). Untuk bagian *target object*, gambar masukan yang awalnya berukuran $91 \times 151 \times 3$ ketika diubah menjadi format warna pada skala keabuan menjadi 91×151 . Sementara itu untuk bagian pada skema deteksi (*detection scheme*), gambar masukan yang berukuran $480 \times 640 \times 3$ berubah menjadi 480×640 . Nilai masing-masing gambar tersebut berubah menjadi 480×640 . Nilai masing-masing gambar tersebut berubah menjadi sebuah nilai yang memiliki rentang 0 sampai 1. Perhitungan konversi format warna dari RGB ke skala keabuan ini dapat dilakukan dengan menggunakan persamaan (2.1) seperti yang telah dijelaskan pada bab sebelumnya. Ilustrasi hasil proses ini dapat dilihat pada Gambar 3.1.

b) Calculate Gaussian Pyramid

Setelah melalui proses perubahan format warna dari RGB ke skala keabuan, proses selanjutnya adalah *downsampling* menggunakan GP. Alasan dilakukannya proses *downsampling* ini adalah untuk mempercepat dan mengefisiensikan waktu komputasi pada proses-proses selanjutnya. Level yang digunakan pada *downsampling* ini adalah dua. Sehingga untuk bagian *target object*, gambar masukan yang berukuran 91×151 ketika diubah menjadi format warna pada skala keabuan menjadi 23×28 . Sementara itu untuk bagian pada skema deteksi (*detection scheme*), gambar masukan yang berukuran 480×640 berubah menjadi 120×160 . Proses perhitungan *downsampling* ini dapat dilakukan dengan menggunakan persamaan (2.3) dengan nilai filter yang sama seperti yang dijelaskan pada bab sebelumnya. Ilustrasi hasil proses ini dapat dilihat pada Gambar 3.1.

c) Penyamaan Ukuran Dengan Zero Padding

Dikarenakan ukuran hasil pada *target object* dan *image* pada bagian skema deteksi tidak sama setelah proses *downsampling* menggunakan GP, maka agar proses perhitungan *normalized cross correlation* dapat dilakukan, ukuran *target object* dan *image* tersebut haruslah sama. Dikarenakan hal tersebut, maka proses *zero padding* dilakukan agar ukuran *target object* dan *image* pada bagian skema deteksi adalah 256×256 . Ilustrasi hasil proses ini dapat dilihat pada Gambar 3.1.

d) Calculate 2D FFT

Setelah proses *zero padding* dilakukan, maka langkah selanjutnya adalah perubahan domain dari domain spatial ke domain frekuensi dengan cara menghitung *2D FFT* dari sinyal tersebut. Langkah ini dibutuhkan untuk mempercepat dan mengefisiensikan waktu komputasi karena dalam domain frekuensi, konvolusi antara dua sinyal tersebut hanya berupa perkalian biasa. Ilustrasi hasil proses ini dapat dilihat pada Gambar 3.1.

e) Calculate normalized cross correlation

Untuk mengetahui apakah *target object* dan *image* tersebut berkorelasi, maka dilakukanlah proses untuk menghitung korelasi silang. Korelasi silang tersebut bertujuan untuk mencari respon antara kedua sinyal tersebut. Hasil pada proses korelasi silang yaitu berupa angka, jika angka yang dihasilkan tinggi, maka dapat disimpulkan bahwa kedua sinyal tersebut berkorelasi (memiliki tingkat kemiripan yang tinggi). Akan tetapi nilai yang dihasilkan

pada proses korelasi silang sulit dipahami. Sehingga untuk lebih mudah dimengerti, hasil korelasi silang harus dinormalisasi. Hasil pada normalisasi tersebut berupa angka dari -1 hingga 1. Dalam kasus ini, nilai *threshold* yang ditentukan adalah 0.99, maka nilai dari proses normalisasi korelasi silang yang lebih tinggi dari 0.99 disimpulkan bahwa *target object* dan *image* tersebut berkorelasi. Proses perhitungan normalisasi ini dapat dilakukan dengan menggunakan persamaan (2.xx) seperti yang dijelaskan pada bab sebelumnya.

f) Estimated location of the target object

Nilai-nilai yang diperoleh dari hasil normalisasi korelasi silang kemudian ditampung jika nilai tersebut melebihi dari batas *threshold* yang ditentukan. Nilai-nilai tersebut memiliki informasi dimana posisi *target object* berada pada *image*, hal ini dikarenakan nilai-nilai tersebut memiliki arti bahwa *target object* dan *image* tersebut memiliki tingkat kemiripan yang tinggi. Hasil dari proses ini yaitu sebuah koordinat *target object* pada *image*-nya.

g) Decision the detected object

Proses ini menentukan apakah *target object* harus di *update* atau tidak. Sistem terus berjalan hingga minimal ada 1 *frame image* pada video yang didalamnya terdeteksi ada suatu *landmark* atau tidak. Jika sistem telah mendeteksi adanya suatu *landmark*, maka sistem melanjutkan hingga *frame image* tersebut merupakan *frame image* terakhir terdeteksinya suatu *landmark*. Setelah tidak adanya *landmark* yang terdeteksi, sistem melanjutkan proses *update* untuk *landmark* berikutnya.

h) Update the target object

Setelah ditentukan apakah *landmark* tersebut telah mencapai *frame* terakhir, langkah selanjutnya adalah meng-*update target object* untuk pendeteksian *landmark* berikutnya. *Target object* yang di *update* diulang prosesnya dari merubah citra RGB ke skala keabuan hingga menghitung *2D FFT* dari sinyal tersebut yang kemudian masuk ke proses *looping* untuk menentukan terdeteksi atau tidaknya *target object* pada *image*.

3. Pembahasan

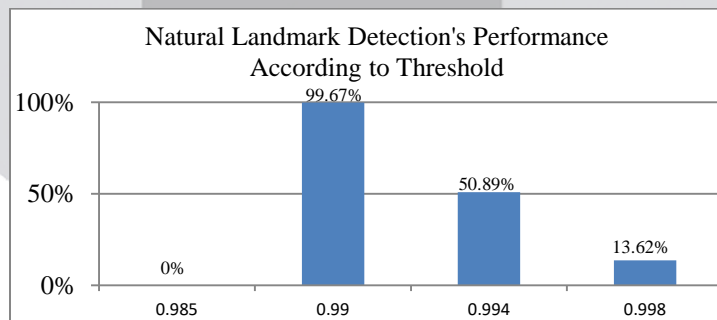
Untuk mengetahui performansi sistem yang dirancang, maka diperlukan pengujian terhadap sistem tersebut. Robot berjalan pada lorong Gedung BTP pada siang hari dengan arah dan kecepatan yang berbeda-beda pada tiap skenarionya. Ciri *landmark* didapat dengan cara memotong citra pada *frame* video yang telah ditentukan. Video direkam dengan menggunakan *webcam* dengan spesifikasi yang dijelaskan pada sub bab 4.1. *Landmark* dideteksi secara berurutan sebagai pengenalan posisi robot beroda tersebut.

3.1 Pengujian Skenario 1

Pada pengujian skenario 1, sistem menggunakan nilai *threshold* pada saat *pattern matching* yang berbeda-beda. Nilai *threshold* pada *pattern matching* ini berpengaruh pada tingkat akurasi sistem.

Nilai *threshold* yang digunakan adalah: 0.985, 0.99, 0.994, 0.998. Hal yang diteliti dalam skenario ini adalah seberapa banyak jumlah *frame* yang terdeteksi oleh sistem. Tingkat akurasi didapat dari perbandingan antara jumlah *frame* yang seharusnya terdeteksi dengan pengamatan manual dengan jumlah *frame* yang terdeteksi oleh sistem. Video yang digunakan dalam pengujian ini adalah video yang bergerak maju dengan kecepatan stabil.

Berikut merupakan hasil pengujian sistem:



Grafik 4.1 Grafik Pengaruh *Threshold* Terhadap Akurasi Pendeteksian *Landmark*

Bedasarkan hasil pengujian, akurasi tertinggi mencapai 99.672853225% didapatkan ketika sistem menggunakan *threshold* = 0.99. Sedangkan akurasi terendah didapatkan ketika sistem menggunakan *threshold* = 0.998. Pengujian pada *threshold* = 0.985 *error*, hal ini terjadi karena nilai *threshold* yang terlalu rendah.

Tabel 2 Waktu Komputasi Skenario 1

THRESHOLD	Waktu Komputasi (detik)				
	LANDMARK 1	LANDMARK 2	LANDMARK 3	LANDMARK 4	Rata -Rata
0.99	0.00650436503014	0.00639467236398	0.00605125364740	0.00629187746469	0.00631054212655
0.994	0.00625561875655	0.00621426376018	0.00621465882128	0.00615323472081	0.00620944401471
0.998	0.00608364865793	0.00631468829284	0.00620896204016	0.00615292657315	0.00619005639102
Rata - Rata Total					0.00623668084409

Waktu komputasi pada sistem yang dirancang pun cukup cepat untuk mendeteksi *landmark*. Rata-rata waktu komputasi untuk 1 frame adalah 6ms.

3.2 Pengujian Skenario 2

Pada pengujian skenario 2, sistem diuji dengan beberapa video yang kecepatan dan arahnya berbeda-beda. Video yang diuji terbagi menjadi 4 macam yaitu video yang bergerak maju dengan kecepatan stabil, video yang bergerak mundur dengan kecepatan stabil, video dengan kecepatan tak beraturan yang bergerak maju, dan video dengan kecepatan tak beraturan yang bergerak mundur.

Nilai *threshold* yang digunakan pada pengujian ini adalah 0.99 karena pada pengujian skenario 1, *threshold* tersebut memiliki tingkat akurasi yang paling tinggi. Hal yang diperhatikan yaitu jumlah *landmark* yang dikenal. Sistem dikatakan dapat mengenal *landmark* jika tingkat akurasi pada jumlah frame yang seharusnya terdeteksi bernilai $\geq 50\%$. *Landmark* tersebut berjumlah 4 yaitu lokasi 1, lokasi 2, lokasi 3, dan lokasi 4.

Berikut merupakan hasil pengujian sistem:

1. Video yang bergerak maju dengan kecepatan stabil :

Tabel 3 Akurasi Landmark dengan Arah Maju dan Kecepatan Stabil

Arah	Kecepatan	Landmark ke-	Pengujian					Akurasi
			1	2	3	4	5	
Maju	Stabil	1	Ya	Ya	Tidak	Tidak	Tidak	40%
		2	Ya	Ya	Ya	Ya	Ya	100%
		3	Ya	Tidak	Ya	Tidak	Ya	60%
		4	Ya	Ya	Ya	Ya	Ya	100%

2. Video yang bergerak mundur dengan kecepatan stabil :

Tabel 4 Akurasi Landmark dengan Arah Mundur dan Kecepatan Stabil

Arah	Kecepatan	Landmark ke-	Video ke- (Terdeteksi ?)					Akurasi
			1	2	3	4	5	
Mundur	Stabil	1	Tidak	Ya	Ya	Ya	Ya	80%
		2	Ya	Ya	Ya	Ya	Ya	100%
		3	Ya	Tidak	Tidak	Ya	Ya	60%
		4	Tidak	Tidak	Ya	Tidak	Ya	40%

3. Video yang bergerak maju dengan kecepatan tak beraturan :

Tabel 5 Akurasi *Landmark* dengan Arah Mundur dan Kecepatan Stabil

Arah	Kecepatan	Landmark ke-	Video ke- (Terdeteksi ?)					Akurasi
			1	2	3	4	5	
Maju	Tak Beraturan	1	Ya	Tidak	Ya	Ya	Ya	80%
		2	Ya	Ya	Ya	Ya	Ya	100%
		3	Tidak	Ya	Tidak	Ya	Tidak	40%
		4	Ya	Ya	Ya	Ya	Ya	100%

4. Video yang bergerak mundur dengan kecepatan tak beraturan :

Tabel 6 Akurasi *Landmark* dengan Arah Mundur dan Kecepatan Stabil

Arah	Kecepatan	Landmark ke-	Video ke-					Akurasi
			1	2	3	4	5	
Mundur	Tak Beraturan	1	Ya	Ya	Ya	Tidak	Ya	80%
		2	Ya	Ya	Ya	Ya	Ya	100%
		3	Tidak	Ya	Ya	Tidak	Tidak	40%
		4	Ya	Ya	Ya	Ya	Ya	100%

Dari 4 pengujian diatas didapatkan table akurasi per-*landmark*, yaitu:

Tabel 7 Tingkat Akurasi per-*landmark*

Landmark	Pengujian				Akurasi
	1	2	3	4	
1	40%	80%	80%	80%	70%
2	100%	100%	100%	100%	100%
3	60%	60%	40%	40%	50%
4	100%	40%	100%	100%	85%

Akurasi terbaik sebesar 100% didapatkan ketika sistem melakukan deteksi pada *landmark* kedua. Sementara akurasi terendah sebesar 50% didapatkan ketika sistem melakukan deteksi pada *landmark* ketiga. Hal ini disebabkan karena ciri *landmark* kedua sangat baik dengan intensitas cahaya yang stabil sehingga tidak banyak menimbulkan perbedaan bayangan pada *landmark* tersebut dan kecepatan sistem pada saat mendeteksi *landmark* kedua cenderung stabil. Sementara pada *landmark* lainnya intensitas cahaya berubah-ubah dan kecepatan sistem pada saat mendeteksi *landmark* cenderung sangat cepat sehingga banyak menimbulkan perbedaan bayangan.

4. Kesimpulan

Berdasarkan tujuan dari penelitian dan hasil pengujian sistem, maka dapat ditarik kesimpulan:

1. Pengenalan posisi dilakukan berdasarkan pendeteksian *landmark* dengan melakukan perhitungan normalisasi korelasi silang antara *target object* dengan *detection scheme*-nya. Jika hasil nilai nomralisasi korelasi silang berada diatas nilai *threshold* maka *landmark* yang dianalisis dapat terdeteksi.
2. Performansi sistem meliputi akurasi pengenalan dan waktu. Akurasi pengenalan *landmark* terbaik sebesar 99.67% terjadi pada saat nilai *threshold* 0.99. Semakin tinggi nilai *threshold* tidak menentukan tingkat akurasi yang lebih tinggi. Hal ini disebabkan karena jika nilai *threshold*-nya terlalu tinggi, maka sistem mencari objek yang sangat mirip, sedangkan disetiap video ada beberapa faktor yang berpengaruh yaitu kecerahan dan jarak.
3. Parameter arah dan kecepatan berpengaruh pada tingkat akurasi sistem. Semakin cepat robot bergerak, maka tingkat *blurring* akan tinggi yang menyebabkan sistem tersebut menjadi kurang baik karena kamera yang

digunakan kurang banyak mengambil *frame* setiap detiknya. Akurasi terbaik didapatkan pada *landmark* kedua dengan nilai akurasi sebesar 100%.

Daftar Pustaka:

- [1] Bonin-Font, Francisco., Ortiz, Alberto., Oliver, Gabriel. 2008. "Visual Navigation for Mobile Robots : a Survey". *Journal of Intelligent and Robotic Systems Volume 53*, Issue 3, pp 263-296
- [2] Gonzales, Rafael C., Woods, Richard E. 2002. *Digital Image Processing*. 2nd Edition. Upper Saddle River, New Jersey: Tom Robbins.
- [3] Guzel, Sedar, Mehmet., Bicker, Robert. 2012. "A Behaviour-Based Architecture for Mapless Navigation Using Vision." *International Journal of Advanced Robotic System (INTECH)*.
- [4] Euayana. 2014. "Sejarah Robot dan Pengertian tentang Robotika". [Online]. <http://www.euayana.com/2012/08/sejarah-robot-dan-pengertian-tentang.html>. [Diakses pada tanggal 20 Oktober, 2017]
- [5] Liy, Noor. 2012. "Pengolahan Citra – Perbaikan Kualitas Citra". [Online]. http://noorliys.blogspot.com/2012/09/pengolahan-citra-perbaikan-kualitas_17.html. [Diakses pada 14 Desember 2017]
- [6] Nugraha, Lerry Bayunuari. 2014. "Perancangan Sistem Pendeteksi Arah Sebagai Alat Bantu Navigasi Pada Robot Beroda Berbasis Pengolahan Citra Digital". Skripsi Sarjana Telkom University Bandung : tidak diterbitkan.
- [7] Rusdinar, Angga., Kim, Sungshin. 2013. "Vision-Based Localization Using Artificial". *International Journal of Fuzzy Logic and Intelligent System*. Korean Institute of Intelligent System Vol 13, No 2, June 2013, pp 133-139.
- [8] Bay, Herbert, Tinne, Tuytelaars., Gool, Lucvan. 2008. "SURF: Speeded Up Robust Features". *Computer Vision and Image Understanding* 110, no. 3, pp 346-359
- [9] Carvalho, Bruno M., Gau, C. Joe., Herman, Gabor T., Kong, T. Yung. 1999. "Algorithms for Fuzzy Segmentation". *Pattern Analysis & Application* (Springer-Verlag).
- [10] Ukhtary, Muhammad Shoufie. 2014. "Deret dan Transformasi Fourier". [Online]. <http://majalah1000guru.net/2014/04/deret-dan-transformasi-fourier/>. [diakses pada 7 Desember 2017]
- [11] Anomaly. 2016. "Explain (Cross / Auto) Correlation, Normalize & Time shift". [Online]. <https://anomaly.io/understand-auto-cross-correlation-normalized-shift/>. [diakses pada 3 Januari 2018]
- [12] Anbarjafari, Gholamreza. "Digital Image Processing". [Online]. <https://sisu.ut.ee/imageprocessing/book/1>. [diakses pada 9 Januari 2018]
- [13] "Gaussian and Laplacian Pyramid". [Online]. <https://www.cs.utah.edu/~arul/report/node12.html>. [diakses pada 10 Januari 2018]
- [14] "Andromina robot v.1.2, a rover with 4 directional wheels". [Online]. <https://www.robotshop.com/letsmakerobots/andromina-robot-v12-a-rover-with-4-directional-wheels>. [diakses pada 10 Januari 2018]