Prosiding Seminar Nasional Ilmu Pengetahuan dan Teknologi
Jenderal Achmad Yani (SNIJA) 2017
Cimahi, 20 Desember 2017
ISBN: 978-602-429-130-3

Perancangan dan Realisasi Sistem Pendeteksi Objek menggunakan Perangkat Lunak Python 2.7

Asep Najmurrokhman, Aditya Nugraha, Kusnandar, Udin Komarudin, Bambang Wibowo
Jurusan Teknik Elektro, Universitas Jenderal Achmad Yani
Jalan Terusan Jenderal Sudirman PO Box 148 Cimahi 40533
corresponding author : asep.najmurrokhman@lecture.unjani.ac.id

Abstrak*—Di dunia manufaktur, produktivitas merupakan hal yang utama dalam menjalankan bisnisnya. Salah satu aspek yang dapat meningkatkan produktivitas adalah penggunaan robot dalam proses produksi. Salahsatu kendala aplikasi robot dalam proses produksi adalah penentuan informasi tentang lokasi objek dan spesifikasinya yang akan dipindahkan diberikan oleh operator. Untuk mengatasi masalah tersebut, sebuah pendeteksi objek yang menyatu dengan robot dapat meningkatkan kemampuan robot dalam mengenali objek dengan spesifikasinya yang mencakup koordinat tujuan, ukuran dan warna benda. Makalah ini mendeskripsikan sistem pendeteksi objek berbasis pengenalan pola yang dibuat dengan perangkat lunak Python 2.7 yang terintegrasi dengan sebuah kamera 3.0 MP dan mikrokontroler Arduino. Kamera 3.0 MP digunakan sebagai perangkat keras untuk mengambil gambar area kerja yang kemudian keluaran gambar tersebut diproses oleh perangkat lunak Python 2.7 yang sudah diprogram untuk mendeteksi benda kerja. Informasi yang diperoleh berupa ukuran, koordinat dan warna benda kerja yang kemudian dikirimkan ke mikrokontroler Arduino untuk diteruskan ke pengendali robot. Hasil pengujian menunjukkan bahwa sistem kendali yang telah dibuat dapat mendeteksi koordinat benda pada area kerja dengan error rataan mutlak dalam arah x sebesar 1,13 mm dan error rataan mutlak dalam arah y sebesar 0,78 mm dengan tingkat akurasi arah sumbu x sebesar 96,65 % dan akurasi arah sumbu y sebesar 98,07 %. Sementara itu, hasil pengujian dimensi benda memberikan nilai error rataan mutlak sebesar 0,94 mm dengan tingkat akurasi 97,01%, sedangkan hasil pengujian pendeteksian warna benda menghasilkan akurasi sebesar 100% untuk mendeteksi warna hijau, merah dan biru dengan latar benda berwarna hitam.

Kata kunci—kamera; mikrokontroler Arduino; python; pengenalan pola; robot.

I. PENDAHULUAN

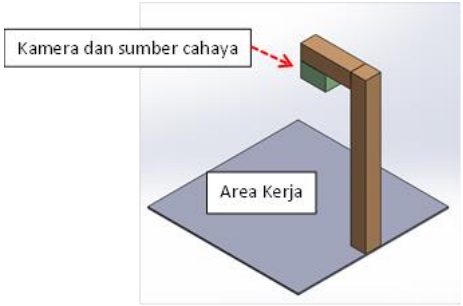
Di dunia manufaktur, produktivitas merupakan hal yang utama dalam menjalankan bisnisnya. Salah satu aspek yang dapat meningkatkan produktivitas adalah penggunaan robot dalam proses produksi. Salahsatu kendala aplikasi robot dalam proses produksi adalah penentuan informasi tentang lokasi objek dan spesifikasinya yang akan dipindahkan diberikan oleh operator. Untuk mengatasi masalah tersebut, sebuah pendeteksi objek yang menyatu dengan robot dapat meningkatkan kemampuan robot dalam mengenali objek dengan spesifikasinya yang mencakup koordinat tujuan, ukuran dan warna benda. Deteksi objek (*object detection*) adalah proses penentuan objek dalam sebuah citra digital. Citra tersebut diperoleh melalui kamera dengan spesifikasi tertentu kemudian diolah dengan metode spesifik sehingga objek tersebut dapat diidentifikasi atau diklasifikasikan. Pengenalan objek ini menjadi bagian yang sangat penting dan kritis dalam pemrosesan citra karena akan menentukan akurasi dalam proses pengambilan keputusan yang sangat bergantung kepada identifikasi gambar atau objek [1]. Beberapa peneliti melaporkan metode deteksi objek dengn menggunakan teknik dan perangkat lunak tertentu maupun implementasi perangkat kerasnya. Hu *et al.* dalam [2] menguraikan sebuah teknik deteksi objek yang banyak dalam satu kerangka kerja sehingga proses deteksi berlangsung dengan cepat. Sementara itu, Radovic *et al.* dalam [3] menerapkan metode *convolutional neural network* (CNN) dalam deteksi objek oleh pesawat udara nirawak sehingga pesawat tersebut dapat membantu secara efektif dalam penanganan bencana, pengiriman logistik, atau transportasi secara umum. Penelitian lainnya berupa metode deteksi objek untuk diterapkan dalam pengendalian setir secara otomatis [4], algoritma *scale invariant feature transforms* untuk membantu akurasi deteksi objek oleh robot yang beroperasi dalam durasi yang lama [5], deteksi objek robot manipulator dengan menggunakan jaringan syaraf tiruan [6], deteksi objek dengan posisi tertentu dalam operasi penyimpanan barang di gudang oleh robot [7], dan sebagainya.

Makalah ini mendeskripsikan sistem pendeteksi objek berbasis pengenalan pola yang dibuat dengan perangkat lunak Python 2.7 yang terintegrasi dengan sebuah kamera 3.0 MP dan mikrokontroler Arduino. Kamera 3.0 MP digunakan sebagai perangkat keras untuk mengambil gambar area kerja yang kemudian keluaran gambar tersebut diproses oleh perangkat lunak Python 2.7 yang sudah diprogram untuk mendeteksi benda kerja. Informasi yang diperoleh berupa ukuran, koordinat, dan warna benda kerja yang kemudian dikirimkan ke mikrokontroler Arduino untuk diteruskan ke pengendali robot. Python adalah salahsatu bahasa pemrograman berorientasi objek yang dapat digunakan dalam beragam aplikasi yang menuntut kecepatan dan ketepatan prosesnya. Perangkat lunak tersebut dapat diunduh dan dikembangkan secara bebas karena mengusung konsep *open source* [8]. Beberapa contoh penggunaan Python diantaranya dalam deteksi kendaraan di jalan tol [9], pengolahan citra uang kertas untuk mengetahui keasliannya [10], pelacak lokasi kendaraan [11], dan lain-lain.

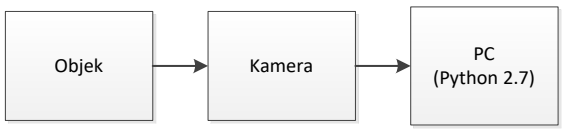
ISBN: 978-602-429-130-3

II. METODE

Sistem pendeteksi objek yang dirancang dan direalisasikan diberikan dalam Gambar 1, sedangkan alur prosesnya secara sederhana diperlihatkan pada Gambar 2.

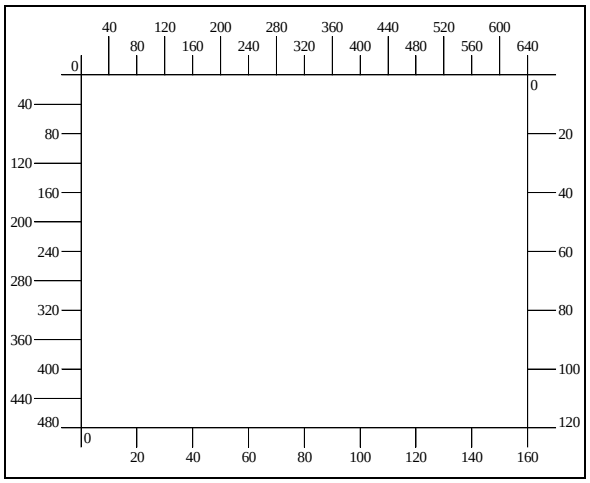


Gambar 1. Perangkat pendeteksi objek



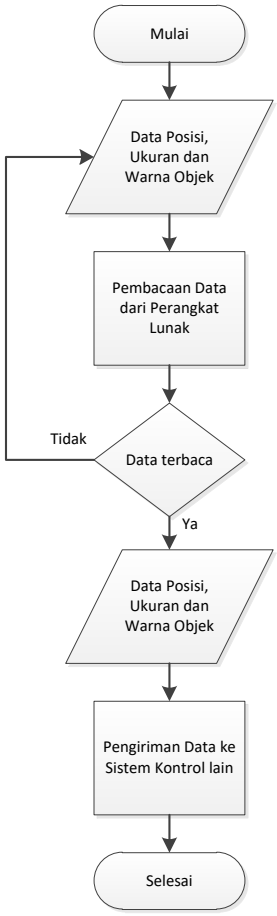
Gambar2. Alur proses pengolahan citra

Sistem dibuat untuk mendeteksi objek yang berada pada jangkauan area kerja secara otomatis. Sistem ini menggunakan metoda pengenalan pola yang mengharuskan intensitas cahaya pada area kerja tidak berubah oleh karenanya diharuskan penambahan sumber cahaya eksternal. Konstruksi penyangga kamera, penyangga sumber cahaya dan area kerja diberikan pada Gambar 1. Penyangga kamera dan sumber cahaya dibuat dari akrilik dengan tebal 3 mm. Alasan pemilihan bahan ini adalah karena mudah dalam proses pembuatan dan perakitannya serta bersifat kaku sehingga tidak mudah mengalami kerusakan. Landasan area kerja dibuat dari bahan akrilik yang memiliki gambar latar belakang yang berisi angka-angka koordinat yang memudahkan proses pengukuran dan pengujian sistem seperti diperlihatkan Gambar 3.



Gambar 3. Rancangan latar belakang area kerja

Objek yang berada pada area kerja atau ROI (*Region of Interest*) akan diakuisisi oleh kamera dengan syarat intensitas cahaya yang dipantulkan sekitar 700-1000 lux. Agar cahaya yang didapatkan di sekitar area kerja mencukupi, maka sistem dilengkapi dengan sumber cahaya tambahan berupa lampu pijar yang memiliki spektrum warna lengkap sehingga tidak akan mengubah warna objek yang diakuisisi oleh kamera. Data citra (*image*) dari kamera ini direpresentasikan dengan muatan listrik pada tiap piksel di bidang sensor kamera kemudian informasi tersebut menghasilkan keluaran berupa matriks *image* yang akan diolah lebih lanjut di PC yang sudah terinstalasi perangkat lunak Python. Proses pengolahan citra yang pertama dilakukan pengubahan data citra yang telah didapatkan menjadi bentuk *grayscale* yaitu bentuk abu-abu. Proses berikutnya adalah pengubahan citra menjadi citra biner untuk memisahkan objek benda dengan latar belakangnya. Proses pendeteksian tepi selanjutnya dilakukan untuk menentukan apakah objek yang terekam oleh kamera merupakan objek yang sudah didefinisikan sebelumnya yang dilanjutkan dengan proses penentuan titik tengah objek dan estimasi posisi objek pada sumbu x dan sumbu y pada area kerja. Terakhir semua data yang telah didapatkan oleh perangkat lunak dikirimkan ke mikrokontroler Arduino agar dapat dieksekusi ke langkah berikutnya. Diagram alir proses yang direalisasikan melalui pemrograman dalam mikrokontroler Arduino diberikan pada Gambar 4.

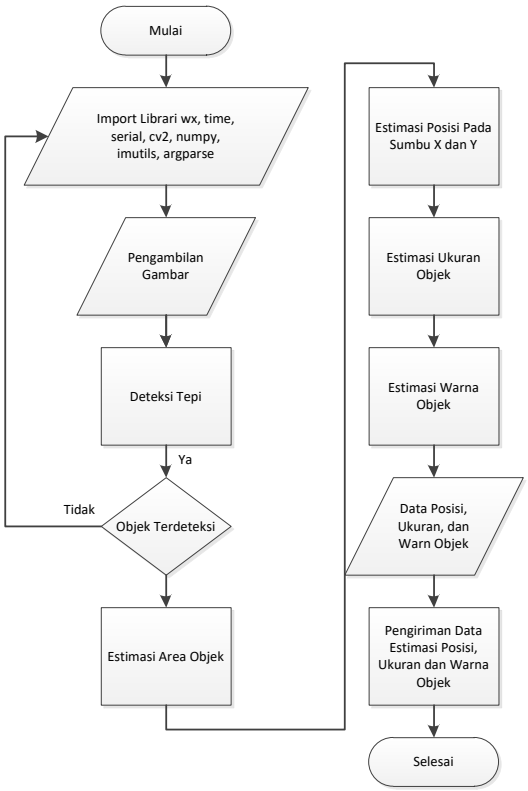


Gambar 4. Diagram alir proses pada mikrokontroler

Sementara itu, perangkat lunak yang digunakan untuk memproses objek yang diinstalasi dalam komputer adalah Python versi 2.7. Perangkat lunak ini digunakan sebagai pengendali dan antarmuka HMI. Sistem yang akan dibuat

ISBN: 978-602-429-130-3

pada perangkat lunak ini diharapkan memudahkan pengguna dalam proses pengoperasian dan memiliki kehandalan dalam melakukan proses pengendalian. Gambar 5 menunjukkan diagram alir dalam perangkat lunak Python yang direalisasikan dalam sistem pendeteksi objek tersebut.



Gambar 5. Diagram alir proses deteksi objek dalam yang direalisasikan oleh perangkat lunak Python

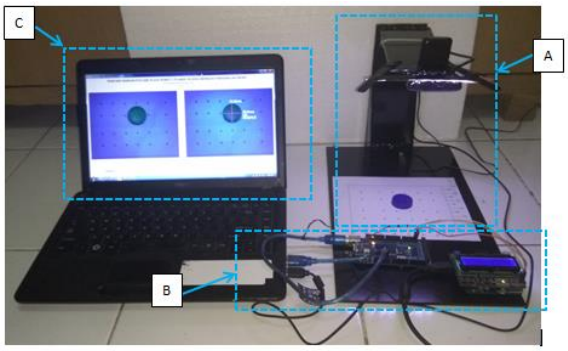
Pengambilan gambar dilakukan menggunakan kamera berbentuk *webcam* dengan resolusi 3.0 megapiksel. Ukuran gambar yang diambil berdimensi 640 x 480 piksel. Proses berikutnya adalah deteksi tepi. Deteksi tepi dilakukan untuk menentukan komponen-komponen yang bermakna untuk diproses pada gambar. Dengan menggunakan proses deteksi tepi maka objek akan diklasifikasikan dari gambar latar belakangnya. Proses deteksi tepi yang dilakukan pada proses ini adalah menggunakan deteksi tepi metode *Canny*. Sebelum melakukan proses deteksi tepi metode Canny, gambar yang awalnya berwarna diubah menjadi abu-abu. Lalu dilakukan proses deteksi tepi dengan menggunakan metode *Canny* dengan nilai ambang batas bawah sebesar 50 dan nilai ambang batas atas sebesar 100.

Sementara itu, proses estimasi area objek dilakukan untuk mencari area (kontur) yang telah dilakukan proses deteksi tepi. Hasil keluaran dari proses ini adalah koordinat ujung kanan atas, ujung kanan bawah, ujung kiri atas, dan ujung kiri objek. Setelah didapatkan keempat titik ujung koordinat objek, maka dicari koordinat tengah posisi objek pada gambar dengan mencari nilai titik tengah antara keempat tituk ujung koordinat objek. Setelah didapatkan titik tengah objeknya, nilai koordinat tersebut merupakan nilai koordinat piksel pada gambar. Untuk mendapatkan nilai koordinat objek asli, nilai koordinat piksel pada gambar dikalikan dengan nilai variabel 4,3 yang merupakan nilai hasil proses kalibrasi antara koordinat tengah objek pada gambar dengan koordinat tengah objek dengan ukuran di dunia nyata. Untuk mendapatkan nilai ukuran objek, digunakan metode perhitungan jarak Euclidean antara titik tengah yang sudah dilakukan perhitungan sebelumnya. Sama halnya dengan perhitungan posisi objek, nilai jarak yang sudah didapat masih berupa jarak piksel pada gambar. Untuk mengubahnya menjadi jarak pada ukuran aslinya maka dibagi dengan variabel 4,3. Lalu setelah mendapatkan ukuran asli benda, kedua nilai ukuran diambil rata-ratanya agar mendapatkan nilai yang lebih akurat. Langkah berikutnya adalah estimasi warna objek. Langkah tersebut dilakukan untuk menentukan warna objek yang terdeteksi pada area kerja. Hasil keluaran proses ini adalah warna objek yang bisa mendeteksi tiga macam warna yaitu, biru, merah dan hijau.

Pengiriman data estimasi posisi, ukuran dan warna objek dilakukan dari program Python yang terinstalasi dalam komputer kemudian menggunakan metode komunikasi serial dikirim ke mikrokontroller Arduino. Pada program Python data dikirimkan berupa *string* dengan format tertentu.

III. HASIL DAN DISKUSI

Realisasi sistem pendeteksi objek diberikan pada Gambar 6. Pada Gambar 6, hasil realisasi sistem mekanik yang ditunjukkan dengan huruf A yang terdiri dari konstruksi yang terbuat dari akrilik, kamera, lampu dan area kerja yang tepat berada di bawah kamera dengan jarak yang sudah diatur sedemikian rupa. Selain itu juga terdapat realisasi sistem elektrik yang ditunjukkan pada huruf B yang terdiri dari Arduino Mega 2560 sebagai perangkat keras yang digunakan untuk penerima data dari perangkat lunak yang kemudian mengirimkan data tersebut ke kontrol lainnya, dan juga terdiri dari Arduino Uno ditambah LCD (*Liquid Crystal Display*) sebagai pengganti kontrol robot untuk penerima data akhir. Terakhir terdapat realisasi sistem perangkat lunak yang merupakan bagian utama untuk pemrosesan pengolahan data yang ditunjukkan oleh huruf C.

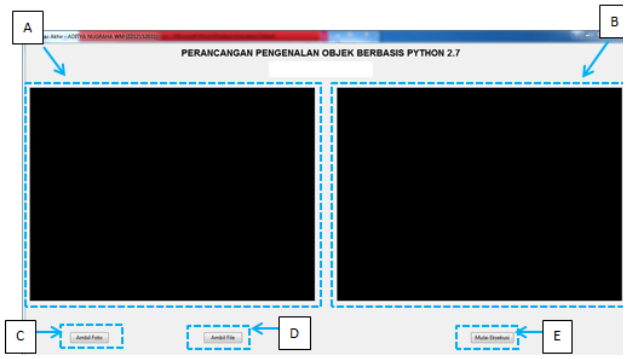


Gambar 6. Realisasi sistem pendeteksi objek

Tampilan sistem di layar komputer saat program yang telah dibuat bekerja dengan baik diberikan pada Gambar 7. Darah yang ditunjukkan oleh huruf A merupakan tempat untuk menampilkan foto atau *file* gambar yang akan dilakukan proses pengenalan objeknya, sedangkan

ISBN: 978-602-429-130-3

area yang ditunjukkan oleh huruf B adalah tempat untuk menampilkan hasil foto atau *file* gambar yang telah dilakukan proses pengenalan objeknya. Sementara itu, kotak yang ditunjukkan oleh huruf C merupakan tombol pilihan untuk mengambil foto langsung dari kamera pada perangkat keras, sedangkan kotak yang ditandai dengan huruf D merupakan tombol pilihan untuk mengambil *file* gambar dari komputer. Kotak yang ditandai dengan huruf E merupakan tombol untuk mengeksekusi foto atau file gambar yang sudah diambil sebelumnya.



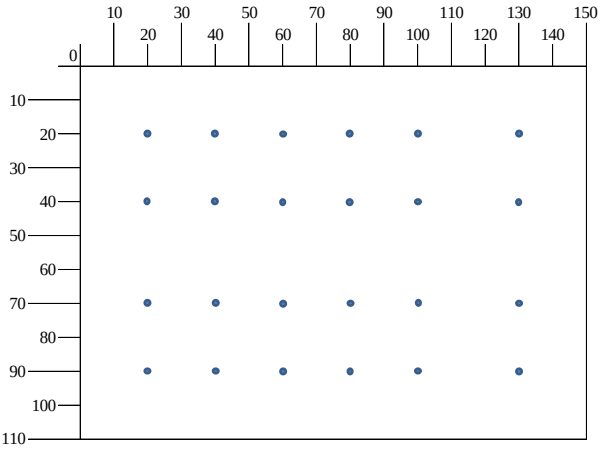
Gambar 7. Tampilan pada layar komputer

Salahsatu hasil eksekusi sistem pendeteksi objek diberikan dalam Gambar 8. Uji coba ini adalah untuk memeriksa apakah komunikasi yang telah dibuat berjalan dengan lancar. Data koordinat benda yang diterjemahkan pada koordinat kartesian (sumbu x, sumbu y) , data ukuran benda dan data warna benda yang telah didapatkan dari program Python dikirimkan melalui komunikasi serial ke mikrokontroler Arduino Mega 2560. Berdasarkan Gambar 8, data koordinat benda berada pada titik 39,2;58,5 kemudian data ukuran benda adalah rata-rata pengukuran diameternya yaitu (31,2+31,6)/2 yaitu 31,4 dan data warnanya yaitu warna biru. Seluruh data tersebut dikirimkan hingga berada pada Arduino Uno yang kemudian ditampilkan pada penampil LCD yang diperlihatkan pada Gambar 9.

Gambar 8. Tampilan hasil eksekusi pada layar komputer

Gambar 9. Tampilan dalam LCD

Eksperimen berikutnya yang dilakukan adalah pengujian akurasi data keluaran dari perangkat lunak Python ketika mendeteksi posisi benda yang berada pada area kerja. Proses ini dilakukan dengan menyimpan benda dengan titik tengah benda berada pada titik-titik yang telah disiapkan agar dapat dibandingkan dengan ukuran ideal seperti diberikan pada Gambar 10.



Gambar 10. Area kerja untuk proses uji coba

Hasil pengujian akurasi data keluaran perangkat lunak dicantumkan dalam Tabel 1.

Tabel 1. Hasil uji coba pengukuran koordinat benda

No	Ideal		Pengukuran		Error			
	Koordinat Sumbu x (mm)	Koordinat Sumbu y (mm)	Koordinat Sumbu x (mm)	Koordinat Sumbu y (mm)	Koordinat Sumbu x		Koordinat Sumbu y	
					(mm)	(%)	(mm)	(%)
1	20	20	18.3	19	1.7	8.50	1	5.00
2	20	40	18	39.7	2	10.00	0.3	0.75
3	20	60	18	59.3	2	10.00	0.7	1.17
4	20	80	18.4	80.3	1.6	8.00	0.3	0.37
5	20	100	18.8	99.3	1.2	6.00	0.7	0.70
6	20	130	17.7	129.3	2.3	11.50	0.7	0.54
7	40	20	39.4	18.5	0.6	1.50	1.5	7.50
8	40	40	40.1	38.8	0.1	0.25	1.2	3.00
9	40	60	39.2	58.5	0.8	2.00	1.5	2.50
10	40	80	39.2	79.7	0.8	2.00	0.3	0.37
11	40	100	39	99.6	1	2.50	0.4	0.40
12	40	130	38.8	129.1	1.2	3.00	0.9	0.69
13	70	20	69	18.3	1	1.43	1.7	8.50
14	70	40	69.4	39.1	0.6	0.86	0.9	2.25
15	70	60	69.8	59.4	0.2	0.29	0.6	1.00
16	70	80	68	80.3	2	2.86	0.3	0.37
17	70	100	68.7	99.3	1.3	1.86	0.7	0.70
18	70	130	68.8	129.1	1.2	1.71	0.9	0.69
19	90	20	89.8	18.7	0.2	0.22	1.3	6.50
20	90	40	89	40.1	1	1.11	0.1	0.25
21	90	60	89.1	59.5	0.9	1.00	0.5	0.83
22	90	80	89.7	79.3	0.3	0.33	0.7	0.88
23	90	100	88.8	99.3	1.2	1.33	0.7	0.70
24	90	130	88	129.1	2	2.22	0.9	0.69
Rata-rata					1.13	3.35	0.78	1.93

Berdasarkan hasil pengujian yang ditampilkan pada Tabel 1, tingkat akurasi dari program pengukuran koordinat benda yang telah dibuat cukup tinggi karena memiliki rata-rata *error* di bawah 5 %. *Error* tertinggi pada koordinat sumbu x yaitu sebesar 11,5 % pada titik koordinat (20,130) yang dapat dilihat pada Gambar 11 dan error tertinggi pada koordinat sumbu y yaitu sebesar 8,5 % pada titik koordinat (70,20) yang dapat dilihat pada Gambar 12. Untuk menghindari *error* ini maka tingkat kinerja kamera harus ditingkatkan dengan cara menaikkan tingkat piksel gambar, meninggikan posisi kamera atau mengecilkan area kerja.

ISBN: 978-602-429-130-3

Gambar 11. Posisi benda kerja dengan error koordinat sumbu x terbesar

Gambar 12. Posisi benda kerja dengan error koordinat sumbu y terbesar

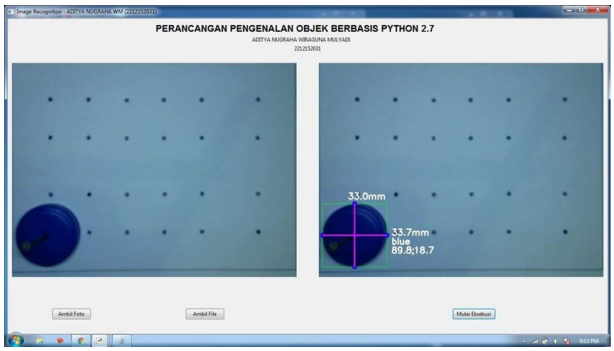
Pengujian berikutnya dilakukan untuk memeriksa tingkat akurasi data keluaran dari perangkat lunak Python ketika mendeteksi ukuran benda yang berada pada area kerja. Proses ini dilakukan bersamaan dengan uji coba pendeteksian posisi benda. Hasil pengujian dibandingkan dengan data ukuran benda asli yang diukur dengan jangka sorong digital yang diperlihatkan pada Gambar 13.

Gambar 13. Pengukuran benda dengan menggunakan jangka sorong digital

Hasil pengujiannya diberikan dalam Tabel 2. Berdasarkan data dari tabel tersebut, tingkat akurasi dari program pengukuran ukuran benda yang telah dibuat cukup tinggi karena memiliki rata-rata *error* dibawah 5%. *Error* tertinggi sebesar 6,68% pada koordinat (90,20) yang ditunjukkan pada Gambar 14. *Error* terbesar terjadi pada ujung kiri bawah area kerja karena tingkat akurasi kamera yang terbatas sehingga terdapat kesalahan pengambilan data (kesalahan paralaksis) karena benda kerja yang berada pada posisi yang jauh dari kamera.

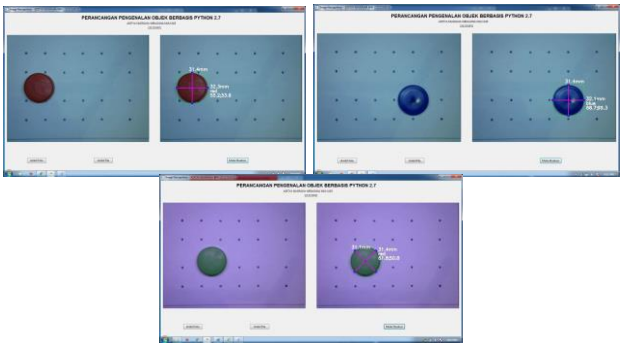
Tabel 2. Hasil uji coba pengukuran ukuran benda

No	Koordinat Sumbu x (mm)	Koordinat Sumbu y (mm)	Ukuran Ideal (mm)	Pengukuran (mm)	Error	
					(mm)	(%)
1	20	20	31.31	33.1	1.79	5.72
2	20	40		32.5	1.19	3.80
3	20	60		32	0.69	2.20
4	20	80		32	0.69	2.20
5	20	100		32.1	0.79	2.52
6	20	130		33	1.69	5.40
7	40	20		33	1.69	5.40
8	40	40		32.1	0.79	2.52
9	40	60		31.4	0.09	0.29
10	40	80		31.4	0.09	0.29
11	40	100		31.7	0.39	1.25
12	40	130		32.8	1.49	4.76
13	70	20		33	1.69	5.40
14	70	40		32	0.69	2.20
15	70	60		31.3	0.01	0.03
16	70	80		31.4	0.09	0.29
17	70	100		31.7	0.39	1.25
18	70	130		32.6	1.29	4.12
19	90	20		33.4	2.09	6.68
20	90	40		32.5	1.19	3.80
21	90	60		31.9	0.59	1.88
22	90	80		31.7	0.39	1.25
23	90	100		32.2	0.89	2.84
24	90	130		33.1	1.79	5.72
Rata-rata					0.94	2.99



Gambar 14. Posisi benda kerja dengan error pengukuran ukuran benda terbesar

Uji coba berikutnya dilakukan untuk menguji tingkat akurasi data keluaran dari perangkat lunak Python ketika mendeteksi warna benda yang berada pada area kerja. Proses uji coba ini dilakukan dengan cara menyimpan benda pada area kerja dengan posisi acak yang kemudian dilakukan pengambilan gambar dan dilakukan proses pendeteksian warna seperti diperlihatkan pada Gambar 15.



Gambar 15. Deteksi warna merah (kiri), biru (tengah), dan hijau (kanan)

Hasil pengujian menunjukkan bahwa sistem pendeteksi objek dapat mengenali warna benda. Dari pengamatan ditemukan bahwa latar belakang tempat benda menentukan akurasi pengenalan warna objek. Semakin

ISBN: 978-602-429-130-3

kontras warna benda dengan warna latar belakangnya, semakin tinggi akurasi hasil identifikasi warna benda tersebut.

IV. KESIMPULAN

Perancangan dan realisasi sistem pendeteksi objek menggunakan perangkat lunak Python telah diuraikan. Berdasarkan hasil eksperimen, sistem pendeteksi objek ini mampu menangkap gambar pada area kerja dan mendeteksi objek berbentuk lingkaran dengan diameter 30 mm yang berada pada area kerja. Selain itu, sistem juga dapat mengukur dimensi, posisi koordinat, dan warna benda kerja. Akurasi hasil pengujian pengukuran posisi benda kerja menunjukkan nilai *error* rataan mutlak dalam arah x yaitu sebesar 96,65% dan dalam arah y sebesar 98,07%. Sementara itu, akurasi hasil pengujian pengukuran dimensi benda kerja menunjukkan nilai *error* rataan mutlak sebesar 97,01%. Akurasi hasil pengujian warna benda kerja menunjukkan bahwa warna dari latar belakang objek menentukan tingkat akurasinya. Semakin gelap warna latar belakang, semakin baik tingkat akurasi deteksi objeknya.

Hasil penelitian yang telah diperoleh dapat dimanfaatkan dalam pendeteksi objek untuk membantu sistem otomasi pemindahan barang sesuai dengan spesifikasinya di industri manufaktur. Untuk mengintegrasikan dengan sistem tersebut, penelitian lanjutan dapat diarahkan kepada perluasan dimensi area kerja yang dicakup oleh pendeteksi objek ini. Selain itu, teknik kecerdasan buatan dalam deteksi objek juga menarik untuk diterapkan karena dapat meningkatkan akurasi seperti yang dilaporkan dalam [6]. Integrasi sistem dengan basis jaringan komunikasi sehingga merealisasikan sebuah *networkd control systems* (NCS) juga menarik untuk diterapkan karena dapat menghemat biaya dan meningkatkan efisiensi seperti yang diuraikan dalam [12].

DAFTAR PUSTAKA

[1] P. Hsiao, F. Chang, and Y. Lin, "Learning Discriminatively Reconstructed Source Data for Object Recognition with Few Examples," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3518–3532, 2016.

[2] Q. Hu, S. Paisitkriangkrai, C. Shen, A. Van Den Hengel, and F. Porikli, "Fast Detecting Multiple Objects in Traffic Scenes with a Single Detection Framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1002–1014, 2016.

[3] M. Radovic, O. Adarkwa, and Q. Wang, "Object Recognition in Aerial Images Using Convolutional Neural Networks," *J. Imaginf*, vol. 3, no. 21, pp. 1–9, 2017.

[4] A. Uçar, Y. Demir, and C. Güzeliş, "Object recognition and detection with deep learning for autonomous driving applications," *Simulation*, vol. 93, no. 9, pp. 759–769, 2017.

[5] A. P. George and F. Joseph, "Object Recognition Algorithms for Computer Vision System : A Survey," *Int. J. Pure Appl. Math.*, vol. 117, no. 21, pp. 69–74, 2017.

[6] K. Kim and J. Kim, "Dynamic Object Recognition Using Precise Location Detection and ANN for Robot Manipulator," in *International Conference on Control, Artificial Intelligence, Robotics & Optimization*, 2017, pp. 237–241.

[7] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A Dataset for Improved RGBD-based Object Detection and Pose Estimation for Warehouse," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 1179–1185, 2016.

[8] A. Patil, D. Huard, and C. J. Fonnesbeck, "PyMC : Bayesian Stochastic Modelling in Python," *J. Stat. Softw.*, vol. 35, no. 4, pp. 1–81, 2010.

[9] M. V. G. Aziz, H. Hindersah, and A. S. Prihatmanto, "Implementation of Vehicle Detection Algorithm for Self-Driving Car on Toll Road Cipularang using Python Language," in *4th International Conference on Electric Vehicular Technology (ICEVT)*, 2017, pp. 149–153.

[10] A. Jalil, "Pengolahan Citra Mendeteksi Keaslian Uang Kertas menggunakan Raspberry Pi," *J. IT*, vol. 14, pp. 28–40, 2014.

[11] A. Roihan, M. S. B. Prasetyo, and A. Rifa'i, "Monitoring Location Tracker Untuk Kendaraan," *CERITA*, vol. 3, no. 2, pp. 148–161, 2017.

[12] A. Najmurrokhman, B. Riyanto, A. Syaichu-Rohman, and Hendrawan, "Dissipative controller design for networked control systems via the Markovian jump system approach," *J. Eng. Technol. Sci.*, vol. 45 B, no. 1, pp. 25–45, 2013.