

Visual Perception Berbasis Camera Pixy Pada Robot Wall Follower

Devi Normalasari, Agusma Wajiyansyah, Arief Bramanto Wicaksono Putra

Politeknik Negeri Samarinda
Teknik Informatika Multimedia
Samarinda, Indonesia

devinormalasari@gmail.com, agusma.wajiansyah@gmail.com, ariefbram@gmail.com.

Abstrak—Pada penelitian ini akan di bentuk perception pada robot wall follower berbasiskan data visual lingkungan sekitar robot yang didapat dari kamera Pixy CMUCam 5. Kamera Pixy CMUCam 5 menggunakan algoritma berbasis warna untuk mendeteksi benda-benda. Proses dilakukan pada development board Arduino Mega. Proses dimulai dengan meng-akuisisi data visual dari kamera pixy dengan sarana komunikasi SPI. Data yang didapat akan dilakukan proses normalisasi untuk mempermudah mendapatkan informasinya. Lingkungan robot dibuat berbentuk labirin sederhana dengan dinding berwarna putih dan terdapat objek berupa kertas karton yang berbentuk persegi sama sisi berwarna merah dan biru yang disusun sedemikian rupa. Susunan kertas karton yang berbentuk persegi sama sisi pada dinding ini akan memberikan informasi berupa lorong, belokan atau persimpangan. Dengan pengetahuan yang diberikan pada robot diharapkan robot mampu mengenali informasi pada labirin tersebut.

Kata Kunci— Wall follower, Robot, visual perception

I. PENDAHULUAN

Mobile robot merupakan robot yang dapat bergerak atau berpindah tempat dengan menggunakan roda atau tiruan bentuk kaki. Pada bagian penggerak robot menggunakan empat buah motor sebagai penggerak. Pada penggerak dengan roda akan memberikan kecepatan dalam melintasi bidang yang rata. Robot Wall follower (RWF) merupakan salah satu jenis dari mobile robot yang bergerak secara mandiri mengikuti dinding. Salah satu hal terpenting pada robot dengan kemampuan bergerak mandiri adalah kemampuan mem-persepsi-kan lingkungan sekitarnya. Untuk menciptakan perceptual system, mobile robot harus memanfaatkan bidang analisis sinyal dan pengetahuan khusus seperti computer vision dengan benar untuk menggunakan banyak sensor. Komunikasi serial antar sensor dan pengendali berfungsi untuk berhubungan dengan robot atau untuk menerima tugas-tugas khusus dari pusat kendali[1-9].

Perception pada robot merupakan kemampuan menafsirkan lingkungan sekitar menggunakan panca indra berupa sensor. Perception dapat dibangun dengan menggunakan berbagai macam sensor antara lain heading sensor (gyroscope, inclinometer, compass), distance sensor (sonar, laser) dan lain – lain. Salah satu metode pengolahan citra yang menggunakan perception yaitu metode pengenalan pola yang bekerja

berdasarkan pencocokan antara citra sample dan citra streaming atau yang biasa dikenal template matching [10, 11].

Computer vision merupakan ilmu dan teknologi mesin yang dapat melihat, dimana mesin mampu mengekstrak informasi dari gambar yang di perlukan untuk menyelesaikan tugas-tugas tertentu. Computer vision didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer mengenali objek yang sedang di amati [11, 12].

Aplikasi computer vision berkisar dari tugas-tugas seperti sistem machine vision industri seperti, pemeriksa botol cepat dari lini produksi, untuk penelitian kecerdasan buatan dan komputer atau robot yang bisa memahami dunia di sekitar mereka. Computer vision dan machine vision fields memiliki sistem tumpang tindih yang signifikan. Computer vision meliputi teknologi inti dari analisis citra otomatis yang digunakan di berbagai bidang. Machine vision biasanya mengacu pada proses menggabungkan analisis citra otomatis dengan metode lain dan teknologi untuk menyediakan inspeksi otomatis dan bimbingan robot dalam aplikasi industri. Dalam banyak aplikasi computer vision, komputer adalah pre-programmed untuk memecahkan tugas tertentu, namun metode berdasarkan pembelajaran yang sekarang menjadi semakin umum. Contoh aplikasi dari computer vision mencakup system tertentu:

- Proses pengendalian, misalnya sebuah robot industri.
- Navigasi, misalnya oleh kendaraan otonom atau mobile robot.
- Mendeteksi peristiwa, misalnya untuk pengawasan visual atau menghitung orang.
- Mengorganisir informasi, misalnya untuk pengindeksan database foto dan gambar urutan.
- Pemodelan benda atau lingkungan, misalnya analisis citra medis atau model topografi.
- Interaksi, misalnya sebagai input ke perangkat untuk interaksi computer ke manusia.
- Inspeksi otomatis, misalnya dalam aplikasi manufaktur [12, 13].

Salah satu perkembangan dari penggunaan teknik *computer vision* adalah *tracking* objek. *Tracking* objek bertujuan untuk mendeteksi dan mengikuti posisi dari suatu objek bergerak yang diinginkan. *Tracking* objek banyak dibutuhkan oleh berbagai macam aplikasi *vision based* seperti *human computer interface*, kompresi/komunikasi video dan sistem keamanan. *Tracking* objek harus mampu mendeteksi objek yang bergerak, mem-filter *noise*, dan gerakan-gerakan lain yang tidak diperlukan[8].

Pada penelitian ini sensor yang digunakan sebagai indra penglihatan robot adalah Pixy CMUCam 5. Pixy CMUCam 5 menggunakan algoritma berbasis warna untuk mendeteksi benda-benda. Pixy akan menghitung warna dan saturasi setiap pixel RGB dari sensor gambar dan menggunakannya sebagai parameter penyangkapan warna. Segala proses masukan dari sensor gambar *Omnivision* akan di olah oleh prosesor dual core NXP LPC4330. Sudut pandang jangkauan kameranya 750 untuk bidang *horizontal* dan 470 untuk bidang *vertical*. komunikasi data yang digunakan dengan Arduino adalah komunikasi SPI (*Serial Peripheral Interface*), dengan tegangan kerja 5 V dan arus 140 mA [11, 14, 15].

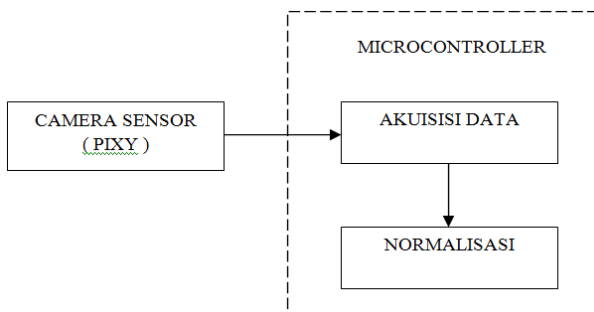
Pada penelitian ini *raw data* dari kamera pixy yang ditampilkan pada serial monitor tidak diketahui urutan posisi warna yang ditangkap oleh kamera pixy, oleh karena itu dibutuhkan proses normalisasi agar *raw data* bisa ditampilkan berurutan pada serial monitor.

Tujuan penelitian ini adalah untuk membangkitkan *perception* pada robot terhadap *raw data* yang didapat dari kamera pixy. *Raw data* tersebut akan melalui proses normalisasi agar *raw data* yang didapatkan dapat mempermudah untuk mendapatkan informasi yang di butuhkan untuk membangun *motion planning* pada robot. Semua proses ini akan diterapkan pada *development board* Arduino Mega. *Raw data* diterima oleh *microcontroller* melalui jalur komunikasi SPI. Dari *perception* yang dibangkitkan akan digunakan untuk pengambilan keputusan pada *motion planning* robot.

II. METODOLOGI

A. Blok Diagram

Gambaran umum dari pengimplementasian *computer vision* pada robot *wall follower* yang melingkupi seluruh proses ditunjukkan pada blok diagram fungsi pada Gambar 1.



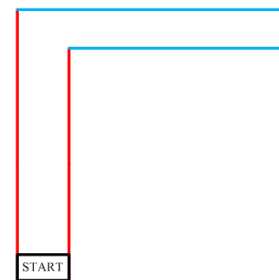
Gambar. 1. Blok Diagram Secara Fungsi.

Pada gambar 1 dijelaskan bahwa data pada kamera sensor (*pixy*) akan di akuisisi didalam *microcontroller* melalui serial komunikasi. Setelah data di akuisisi maka proses selanjutnya adalah normalisasi data agar data yang masuk dapat di urutkan berdasarkan sumbu X pada index *input*.

B. Arena Wall follower

Deskripsi arena yang digunakan untuk robot *wall follower*

- dinding berwarna dasar putih.
- terdapat karton merah dan biru ukuran 10 cm x 10 cm sebagai objek warna yang akan dikenali oleh kamera *pixy*.
- karton merah akan di tempatkan dengan arah dinding *vertical*.
- karton biru akan di tempatkan dengan arah dinding *horizontal*.

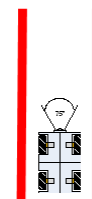


Gambar. 2. Desain arena track robot.

C. Perspective Pada Beberapa Kondisi Robot Di Arena

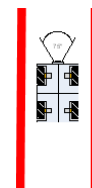
a) Robot Berada Di Lorong

Pada posisi pertama kamera *pixy* menangkap lebih banyak warna merah.



Gambar. 3. Robot pada posisi pertama kondisi berada di lorong.

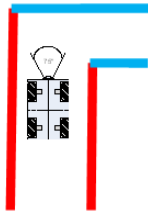
Pada posisi kedua kamera *pixy* menangkap banyak warna merah.



Gambar. 4. Robot pada posisi kedua kondisi berada di lorong.

b) Robot Berada Di Belokan

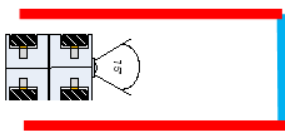
Pada posisi pertama kamera pixy menangkap 2 warna dengan urutan merah – biru – merah.



Gambar. 5. Robot pada posisi pertama kondisi berada di belokan.

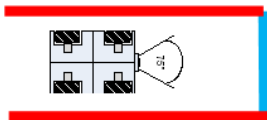
c) Robot Berada di Lorong Buntu

Pada posisi pertama kamera pixy menangkap 2 warna dengan urutan merah – merah – biru – merah – merah.



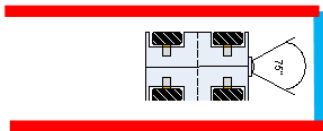
Gambar. 6. Robot pada posisi pertama kondisi berada di lorong buntu.

Pada posisi kedua kamera pixy menangkap 2 warna dengan urutan merah – biru – merah.



Gambar. 7. Robot pada posisi kedua kondisi berada di lorong buntu.

Pada posisi ketiga kamera pixy menangkap banyak warna biru.



Gambar. 8. Robot pada posisi ketiga kondisi berada di lorong buntu.

III. HASIL DAN PEMBAHASAN

A. Hasil Akuisisi Kamera Pixy

Pada gambar 9 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang terhubung langsung dengan komputer saat berada di lorong.



Gambar. 9. Gambar yang di tangkap kamera pixy saat berada di lorong.

Berikut adalah data numerik hasil akuisisi data yang di dapat melalui serial komunikasi antara kamera pixy dan *microcontroller*.

TABEL I. DATA *NUMERIC* HASIL AKUISISI YANG DIDAPAT MELALUI SERIAL KOMUNIKASI ANTARA KAMERA PIXY DAN *MICROCONTROLLER* SAAT BERADA DI LORONG.

Block	Sig	X	Y	Widht	Height
0	1	26	95	18	39
1	1	56	96	10	24
2	1	74	94	5	16
3	2	262	99	14	36
4	2	304	89	25	23
5	4	158	97	22	21
6	4	113	97	21	20
7	4	201	97	20	16
8	4	236	105	10	3
9	4	234	91	5	2

Berdasarkan data *numeric* hasil akuisisi data yang didapat, terdeteksi 10 *block*, dan *block* dimulai dari 0. *Block* adalah lingkup warna yang di deteksi oleh kamera sensor. *Sig* adalah singkatan dari *signature* yang digunakan untuk alamat warna yang akan di kenali oleh kamera sensor (pixy). Pada data yang *numeric* yang didapat, *signature* 1 adalah merah terang, *signature* 2 adalah merah gelap, *signature* 3 adalah biru gelap, dan *signature* 4 adalah biru terang. *X* adalah inisialisasi pada nilai sumbu x, y adalah inisialisasi pada nilai sumbu y, *width* adalah inisialisasi dari lebar *block*, dan *height* adalah inisialisasi dari tinggi *block*.

Pada gambar 10 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang terhubung langsung dengan komputer saat berada di belokan.



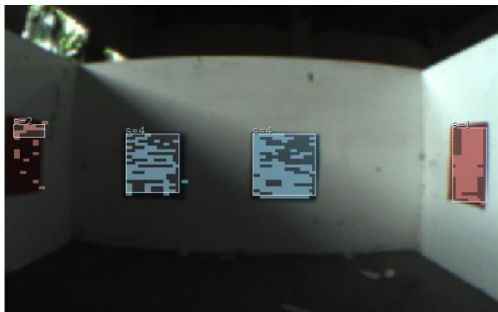
Gambar. 10. Gambar yang di tangkap kamera pixy saat berada di belokan.

Berikut adalah data numerik hasil akuisisi data yang di dapat melalui serial komunikasi antara kamera pixy dan *microcontroller*.

TABEL II. DATA NUMERIC HASIL AKUISISI YANG DIDAPAT MELALUI SERIAL KOMUNIKASI ANTARA KAMERA PIXY DAN MICROCONTROLLER SAAT BERADA DI BELOKAN.

Block	Sig	X	Y	Widht	Height
0	1	26	95	18	39
1	1	55	96	9	28
2	1	74	102	6	3
3	2	307	104	23	62
4	2	262	95	16	31
5	4	114	97	22	20
6	4	157	97	21	21
7	4	236	93	10	7
8	4	236	108	10	3

Pada gambar 11 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang terhubung langsung dengan komputer saat berada di lorong buntu.



Gambar. 11. Gambar yang di tangkap kamera pixy saat berada di lorong buntu.

Berikut adalah data numerik hasil akuisisi data yang di dapat melalui serial komunikasi antara kamera pixy dan *microcontroller*.

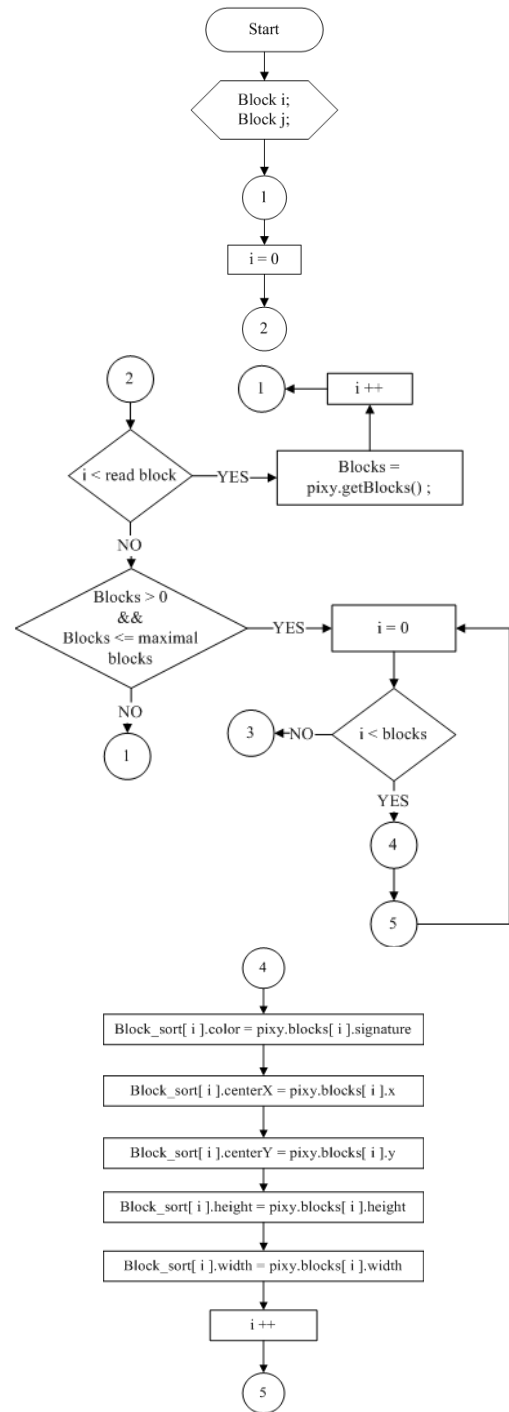
TABEL III. DATA NUMERIC HASIL AKUISISI YANG DIDAPAT MELALUI SERIAL KOMUNIKASI ANTARA KAMERA PIXY DAN MICROCONTROLLER SAAT BERADA DI LORONG BUNTU.

Block	Sig	X	Y	Widht	Height
0	1	294	107	21	48
1	1	260	105	13	34
2	2	21	95	22	49
3	2	57	97	10	29
4	4	180	105	23	30
5	4	114	96	22	15
6	4	104	113	5	2

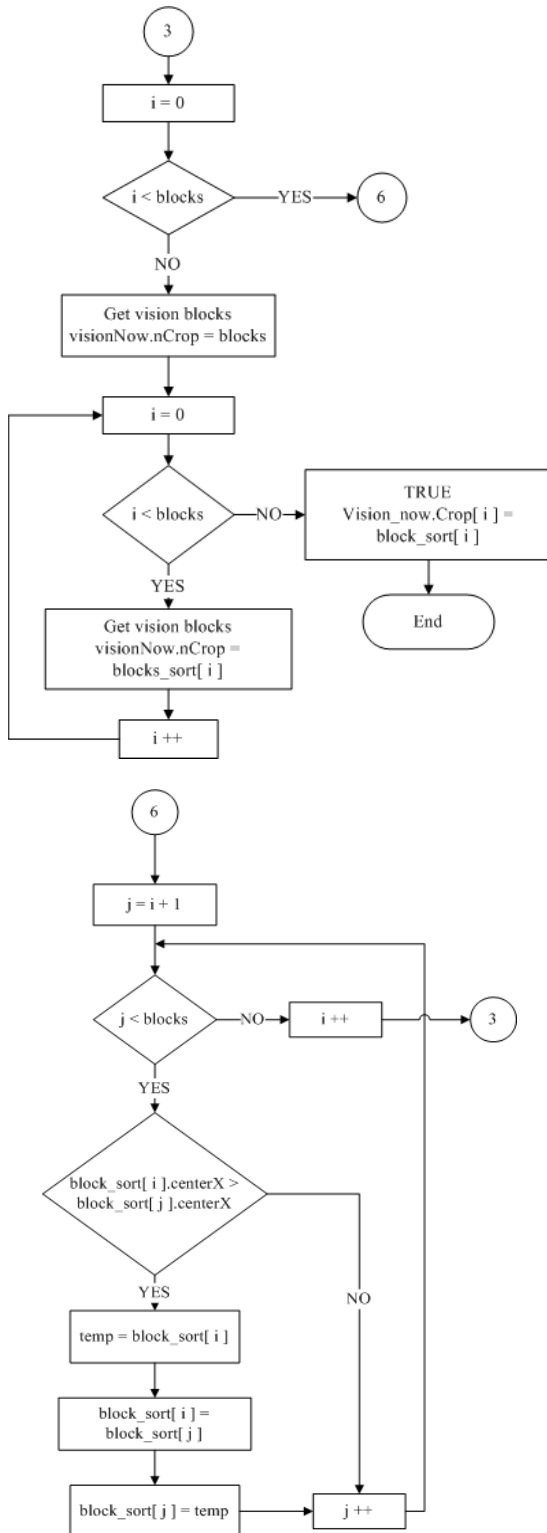
Hasil akuisisi data yang di hasilkan dari kamera pixy tidak berurutan, sehingga di butuhkan normalisasi untuk

menyesuaikan data dengan *perspective* yang diinginkan. Program normalisasi akan mencetak *output* hasil akuisisi data kamera pixy menjadi urut berdasarkan warna pada dinding dari kiri ke kanan.

Berikut adalah *flowchart* dari program normalisasi yang di bangun:



Gambar. 12. Flowchart inialisasi pada program normalisasi.



Gambar. 13. Flowchart program normalisasi hasil akuisisi data dari kamera pixy.

Pada gambar 12 di jelaskan bahwa langkah pertama adalah inialisasi penyimpanan data hasil akuisisi yaitu *block i* dan *block j*, lalu memasuki kondisi *i* sama dengan 0 dan *i* lebih kecil dari *block* yang di akuisisi. Dalam kondisi *i* lebih kecil dari *block* yang di akuisisi terdapat 2 *output* yang berupa kondisi, jika kondisi benar maka *blocks* sama dengan data hasil akuisisi kamera pixy lalu cetak *i* berurutan dan kembali pada kondisi *i* sama dengan 0. Jika kondisi salah maka masuk ke dalam kondisi *blocks* lebih besar dari 0 dan *block* lebih kecil dari *block* paling akhir, jika kondisi salah maka proses akan kembali pada proses *i* sama dengan 0. Jika kondisi benar maka *i* sama dengan 0 lalu memasuki kondisi *i* lebih kecil dari *blocks*, jika kondisi benar maka akan di buat inialisasi yaitu *block sort color* untuk penyimpan data numerik dari *signature*, *block sort center X* untuk penyimpan data numerik dari posisi pada sumbu X, *block sort center Y* untuk penyimpan data numerik dari posisi pada sumbu Y, *block sort height* untuk penyimpanan data numerik dari tinggi *block* yang di akuisisi, *block sort width* untuk penyimpan data numerik dari lebar *block* yang di akuisisi. Setelah itu proses berlanjut pada cetak *i* secara berurutan dan proses kembali pada kondisi *i* sama dengan 0. Tetapi jika *i* lebih kecil dari *block* dalam kondisi yang salah maka masuk ke kondisi *i* sama dengan 0 setelah itu masuk ke kondisi *i* lebih kecil dari *blocks*, jika kondisi *i* lebih kecil dari pada *blocks* adalah benar maka masuk dalam kondisi *j* sama dengan *i* ditambah 1 dan masuk ke dalam kondisi berikutnya yaitu *j* lebih kecil dari *blocks* jika kondisi salah maka cetak *i* secara berurutan dan proses kembali pada *i* sama dengan 0. Tetapi jika kondisi benar maka akan masuk ke dalam kondisi nilai variabel X dalam *block sort i* dan *block sort j* akan di dibandingkan. Jika kondisi salah maka *j* akan di cetak secara berurutan lalu kembali pada proses *j* lebih kecil dari *blocks*, tetapi jika kondisi benar maka nilai *block sort i* akan di simpan dalam temporari dan nilai dalam *block sort j* akan menggantikan nilai pada *block sort i* dan nilai didalam temporari akan menggantikan nilai di dalam *block sort j* lalu proses akan berlanjut pada cetak *j* secara berurutan dan kembali ke pada proses *j* lebih kecil dari *blocks*. Tetapi jika kondisi *i* lebih kecil dari *blocks* adalah salah maka langkah selanjutnya adalah kembali mengambil data dari pixy. setelah itu masuk ke dalam kondisi *i* sama dengan 0 lalu masuk ke dalam kondisi *i* lebih kecil dari *block*, jika kondisi benar maka data yang kembali diambil dari pixy adalah sama dengan *block sort i* lalu cetak *i* berurutan, setelah itu kembali pada proses *i* sama dengan 0. Tetapi jika kondisi salah maka data yang baru sama dengan *block sort* dianggap benar dan proses selesai.

B. Hasil Akuisisi Data Setelah Proses Normalisasi



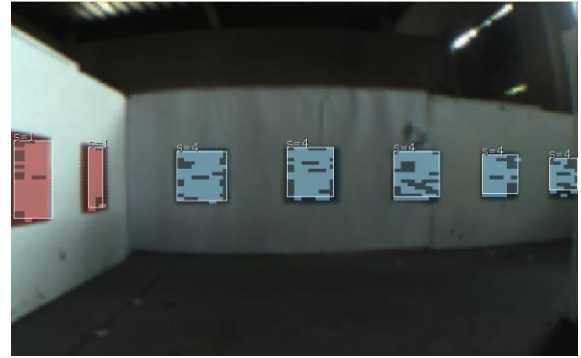
Gambar. 14. Gambar yang di tangkap kamera pixy saat berada di lorong.

Pada gambar 13 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang terhubung langsung dengan komputer saat berada di lorong. Berikut adalah data numerik hasil akuisisi data yang di dapat setelah proses normalisasi melalui serial komunikasi antara kamera pixy dan *microcontroller*.

TABEL IV. DATA NUMERIC HASIL AKUISISI YANG DI DAPAT SETELAH PROSES NORMALISASI SAAT BERADA DI LORONG.

Block	Cx	Cy	T	L	Warna
0	2	85	8	4	1
1	2	115	7	4	1
2	17	21	13	15	4
3	54	98	46	45	1
4	99	89	2	6	4
5	131	101	16	22	4
6	177	96	17	22	4
7	212	101	2	7	4
8	225	91	9	14	4
9	254	91	5	6	4
10	266	97	5	7	4
11	297	102	44	21	3

Berdasarkan data *numeric* hasil normalisasi data yang didapat, terdeteksi 12 *block*, dan *block* dimulai dari 0. *Block* adalah lingkup warna yang di deteksi oleh kamera sensor. Cx adalah singkatan dari center X yang digunakan untuk inialisasi nilai pada sumbu X. Cy adalah singkatan dari center Y yang digunakan untuk inialisasi nilai pada sumbu Y. T adalah singkatan dari tinggi yang digunakan untuk inialisasi nilai tinggi pada *block* warna yang terdeteksi oleh kamera pixy. L adalah singkatan dari lebar yang digunakan untuk inialisasi nilai pada lebar *block* warna yang terdeteksi kamera pixy. Warna adalah inialisasi yang digunakan untuk alamat warna yang dikenali oleh kamera sensor (pixy). Pada data yang *numeric* yang didapat, *signature* 1 adalah merah terang, *signature* 2 adalah merah gelap, *signature* 3 adalah biru gelap, dan *signature* 4 adalah biru terang.

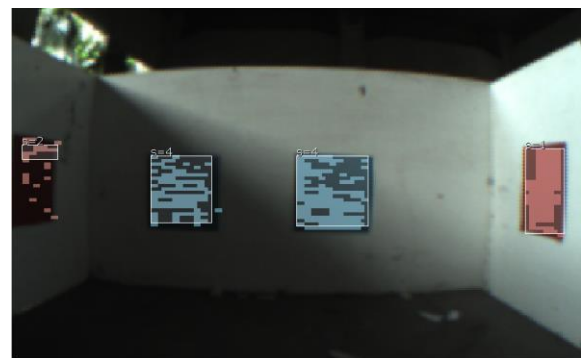


Gambar. 15. Gambar yang di tangkap kamera pixy saat berada di belokan

Pada gambar 14 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang terhubung langsung dengan komputer saat berada di belokan. Berikut adalah data numerik hasil akuisisi data yang di dapat setelah proses normalisasi melalui serial komunikasi antara kamera pixy dan *microcontroller*.

TABEL V. DATA NUMERIC HASIL AKUISISI YANG DI DAPAT SETELAH PROSES NORMALISASI SAAT BERADA DI BELOKAN.

Block	Cx	Cy	T	L	Warna
0	9	103	38	19	1
1	50	87	24	15	1
2	50	108	17	17	1
3	72	61	22	7	4
4	73	121	4	12	4
5	116	96	33	35	4
6	180	100	29	33	4
7	236	105	2	5	4
8	246	84	2	14	4
9	259	108	8	19	4
10	260	94	3	8	4
11	297	115	25	25	4
12	300	92	22	22	4



Gambar. 16. Gambar yang di tangkap kamera pixy saat berada di lorong buntu.

Pada gambar 15 memperlihatkan gambar yang di tangkap oleh kamera pixy menggunakan aplikasi Pixy Monitor yang

terhubung langsung dengan komputer saat berada di lorong buntu. Berikut adalah data numerik hasil akuisisi data yang di dapat setelah proses normalisasi melalui serial komunikasi antara kamera pixy dan *microcontroller*.

TABEL VI. DATA NUMERIC HASIL AKUISISI YANG DI DAPAT SETELAH PROSES NORMALISASI SAAT BERADA DI LORONG BUNTU.

Block	Cx	Cy	T	L	Warna
0	20	112	8	22	3
1	22	77	19	26	3
2	27	98	2	7	3
3	63	7	7	13	4
4	107	86	8	29	4
5	113	99	9	16	4
6	174	84	5	5	4
7	174	100	2	7	4
8	186	84	2	13	4
9	308	93	23	18	1
10	311	119	10	15	1

IV. KESIMPULAN

Berdasarkan analisa dan implementasi yang dilakukan pada penelitian ini, dapat disimpulkan bahwa program normalisasi berhasil digunakan untuk menyusun kembali data hasil akuisisi dari kamera pixy yang acak saat di akuisisi oleh *microcontroller* menjadi data yang tersusun rapi berdasarkan sumbu X terkecil. Sehingga akan memudahkan dalam mengetahui posisi *block* yaitu dari *block* warna kiri sampai *block* warna kanan.

DAFTAR PUSTAKA

- [1] Winarno and D. Arifianto, *Bikin Robot Itu Gampang*. Jakarta: PT. Kawan Pustaka, 2011.
- [2] Nofriyani, E. S. Ningrum, and M. I. Nugraha, "Rancang Bangun Robot Leader Dan Robot Follower Dengan Sistem Navigasi Sensor Infra Merah " *Industrial Electronics Seminar 2011 (IES 2011)* 2011.
- [3] T. Kobayashi, T. Yoshida, J. Maenishi, J. Imae, and G. Zhai. (2008). *Automation And Robotics*.
- [4] Z. Kowalczyk and D. Wesierski. (2008). *Automation And Robotics*.
- [5] D. Tresnawan and Meidi, "Implementasi Metode Maze Dan PID Pada Robot Vacuum Cleaner Otomatis," *Jurnal Ilmiah Mikrotek*, vol. 1, pp. 164-174, 2015.
- [6] Y. Hasyim and A. R. Putri, "Implementasi Sistem Navigasi Robot Wall Following Dengan Metode Fuzzy Logic Untuk Robot Pemadam Api Divisi Berkaki Onix II Pada KRPAAI Tahun 2017," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPDI)*, vol. 02, pp. 26-31, Mei 2017.
- [7] U. Sutisna, W. D. Siregar, and S. Nurhadiyono, "Penerapan Kendali Hybrid Logika Fuzzy PID Untuk Performansi Navigasi Robot Beroda *Wall follower*," *Techno*, vol. 17, pp. 079-087, 2016.
- [8] A. Azhar, K. D. K. W, and H. Subagiyo, "Perancangan Fuzzy Logic Sugeno Untuk *Wall Tracking*," *Jurnal Elementer*, vol. 1, 2015.
- [9] Nuryanto and A. Widiyanto, "Rancang Bangun *Mobile Robot* 2WD Dengan 2 Sensor Untuk Arah Belokan," *Seminar Nasional Teknologi Informasi Dan Multimedia*, 2016.
- [10] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, "Introduction To Autonomous *Mobile Robots*," The MIT Press, p. 11, 2004.
- [11] K. D. Setyanto, I. Fibriani, and Sumardi, "Pengendalian *Mobile Robot Vision* Menggunakan Webcam Pada Objek Arah Panah Berbasis Raspi," *Jurnal Arus Elektro Indonesia (JAEDI)*, vol. 2, pp. 27-32, 2016.
- [12] A. P. W. Wibowo, "Implementasi Teknisi *Computer vision* Metode Colored Markers Trajectory Secara Real Time," *Jurnal Teknik Informatika*, vol. 8, pp. 38-42, 2016.
- [13] D. H. Ballard and C. M. Brown, *Computer vision*. New Jersey: Prentice Hall Inc., 1982.
- [14] Suwasono and I. D. W., "Optimasi Akurasi Deteksi Goal Gawang Dengan Metode Square Grid," *Jurnal Pendidikan: Teori, Penelitian, Dan Pengembangan*, vol. 2, pp. 341-349, 2017.
- [15] A. Deolika and B. A. R., "Rancang Bangun Robot Sepakbola Beroda Menggunakan Image Sensor.," *Jurnal Teknik Informatika Dan Sistem Informasi*, vol. 5, pp. 1173-1310, 2016.