

Implementasi Kontrol PID *Positioning* Robot Pada Titik Pengambilan Objek *Shagai* Menggunakan Sensor Kamera

M. Nizar Falafi Astoro, Totok Winarno, Achmad Komarudin

Abstrak — Dengan adanya berbagai robot yang telah diciptakan, ada beragam jenis kontes atau perlombaan robotika. Salah satunya adalah kontes robot bergengsi di ASIA yaitu *ABU ASIA-PASIFIC ROBOT CONTEST 2019*, sedangkan di Indonesia perlombaan ini disebut dengan Kontes Robot Abu Indonesia 2019. *Rule* perlombaan ini robot diharuskan untuk memilih dan mengambil objek *shagai* yang akan dilempar menuju zona tertentu sesuai aturan lomba. Pada robot ini diharuskan untuk melakukan pendeteksian suatu objek yaitu *shagai*. Proses ini memberikan kemampuan untuk mengetahui perbedaan objek tertentu. Salah satu metode yang digunakan adalah PID (*Proportional Integral Derivative*) sebagai kontrol gerak motor dan robot, menggunakan kamera untuk mengambil gambar lalu dengan algoritma tertentu robot dapat mengetahui titik koordinat posisi objek tersebut. Penggunaan metode PID menjadikan robot ini menjadi lebih stabil dalam mengikuti objek. Robot ini akan menstabilkan posisi robot terhadap objek sesuai *set point* yang diinginkan, dan robot akan bermanuver menuju ke titik pengambilan objek yang berjarak ± 1 m. Berdasarkan solusi diatas dibutuhkan nilai Kp: 4 Ki: 7 Kd: 15 agar mencapai *set point* yang ditentukan. Tujuan dari penelitian ini adalah untuk merealisasikan robot pengambil objek untuk mengikuti Kontes Robot Abu Robocon Indonesia 2019.

Kata kunci : Sensor Kamera, Robot, *Positioning*, *Shagai*, PID

Pengertian ini sangat erat hubungan antara robot dan otomatisasi sehingga dapat dipahami bahwa hampir setiap aktivitas kehidupan modern makin tergantung pada robot dan otomatisasi.

Salah satu kontes robot yang bergengsi di ASIA adalah ABU ASIA-PACIFIC ROBOT CONTEST 2019 dengan tema "Great Urtuu" dan di Indonesia disebut KRAI 2019 (Kontes Robot Abu Robocon Indonesia 2019). KRI atau Kontes Robot Indonesia sendiri adalah sebuah kontes robotika antar Perguruan Tinggi di Indonesia. Dimana pemenang dari KRI akan mewakili Indonesia dalam ABU (Asia-Pacific Broadcasting Union) Robocon yang diadakan setiap tahun dengan lokasi berpindah-pindah dalam negara anggota ABU. Pada perlombaan KRAI 2019 robot diharuskan untuk memilih dan mengambil objek shagai yang akan dilempar menuju zona tertentu sesuai rule.

Pada penelitian kali ini robot yang dibuat adalah jenis robot pengambil objek shagai. Dalam perancangan ini, robot menggunakan 1 sensor kamera dan 3 buah motor. Dimana robot akan bermanuver mendekati objek yang telah terdeteksi oleh sensor kamera. Kecepatan robot pada saat bermanuver untuk melakukan pengambilan objek shagai dapat di kontrol dengan mikrokontroler. Kemudian apabila objek digeser menjauhi robot maka robot akan melakukan scanning setelah posisi objek terdeteksi pada pandangan robot, maka robot akan bermanuver mendekati titik jangkauan pengambilan. Pada proses gerak robot diperlukan sebuah metode yaitu PID sebagai kontrol gerak motor robot.

I. PENDAHULUAN

Pada saat ini, robot memang telah hadir di dalam kehidupan manusia dalam bentuk yang bermacam-macam. Ada robot sederhana untuk mengerjakan kegiatan yang mudah atau berulang-ulang. Ada pula robot yang dirancang untuk "Berperilaku" sangat kompleks dan sampai batas tertentu dapat mengontrol dirinya sendiri. Di kalangan umum pengertian robot selalu dikaitkan dengan "Makhluk Hidup" berbentuk orang maupun binatang yang terbuat dari logam dan bertenaga listrik. Sementara dalam arti luas robot berarti suatu sistem yang terdiri dari mekanisme yang memiliki suatu control elektrik untuk melaksanakan suatu tugas tertentu.

II. TINJAUAN PUSTAKA

2.1 Sensor Kamera Pixy CMUCam 5

Sensor kamera yang dimaksud disini adalah suatu perangkat yang digunakan untuk mendeteksi suatu objek berupa gambar dan warna. Pixy menggunakan algoritma berbasis penyaringan warna untuk mendeteksi benda-benda. Metode penyaringan berdasarkan warna yang populer karena cepat, efisien, dan relatif kuat. Penyaringan warna yang biasa dikenal menggunakan RGB (merah, hijau, dan biru) untuk mewakili warna. Alat ini menggunakan teknik penentuan koordinat dalam pembacaan objek.



Gambar 1. Sensor Kamera CMUCam 5

2.2 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input output*. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

2.3 Motor DC Geared

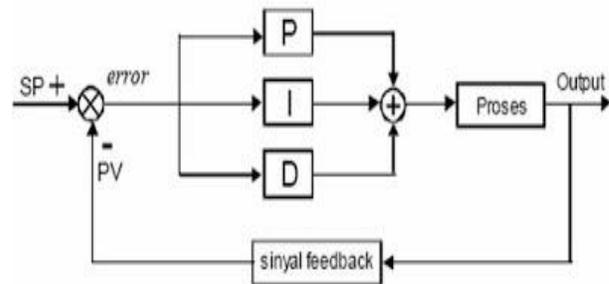
Motor listrik merupakan suatu *device* yang memiliki fungsi untuk mengubah energi listrik menjadi energi gerak. Pedoman motor listrik itu sendiri mengacu pada kaidah tangan kiri. Motor *Planetary Gear* (PG) 45 merupakan tipe motor DC *brushless* produksi *Oriental Motor Corp*. Motor ini dilengkapi *gearbox* dengan perbandingan yang bervariasi. Motor *Planetary Gear* (PG) 45 memiliki kecepatan maksimum 500 rpm. Dengan internak gearnya yang berukuran 10:1 maka kecepatan maksimum pada poros keluaran adalah 50 rpm, dengan torsi rata-rata 11,2 kg-cm. Catu daya yang dibutuhkan adalah 12V.

2.4 Driver Motor BTN7970

BTN7970 adalah IC yang dapat digunakan sebagai *driver* DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. BTN7970 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan motor ini bisa mencapai tegangan 5V-25V. Driver BTN7970 memiliki maksimal arus mencapai 70A. Dalam *chip* terdapat dua rangkaian *H-Bridge*. BTN7970 menggunakan rangkaian dasar transistor (BJT). Berikut adalah rangkaian driver motor BTN7970 yang sudah dirangkai dengan motor dan mikrokontroler.

2.6 Kontrol PID

Sistem kontrol PID terdiri dari tiga buah pengaturan yaitu kontrol P (*Proportional*), I (*Integral*) dan D (*Derivative*). Dalam implementasinya masing-masing dapat bekerja sendiri maupun gabungan, dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan.



Gambar 2. Sistem Kontrol PID

1. Kontrol *Proportional*

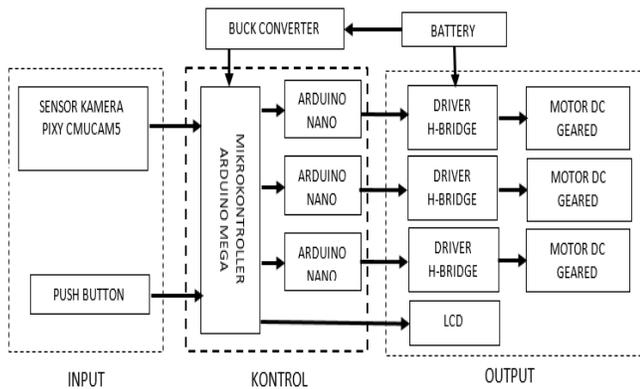
Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon *transien* khususnya *rise time* dan *settling time*. Kontrol proporsional berfungsi untuk memperkuat sinyal kesalahan penggerak (sinyal *error*), sehingga akan mempercepat keluaran sistem mencapai titik referensi.
2. Kontrol *Integral*

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\text{integrale}(t)dT]$ K_i dengan K_i adalah konstanta *integral* dan dari persamaan diatas, $G(s)$ dinyatakan sebagai $u = K_d \cdot [\Delta e / \Delta t]$ Jika $e(T)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki *error*. Jika $e(T)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah orde sistem.
3. Kontrol *Derivative*

Sinyal kontrol u yang dihasilkan oleh kontrol *derivative* dapat dinyatakan sebagai $G(s) = s \cdot K_d$ Dari persamaan di atas, Nampak bahwa sifat dari kontrol D ini dalam konteks “kecepatan” atau *rate* dari *error*. Sifat ini dapat digunakan untuk memperbaiki respon *transien* dengan memprediksi *error* yang akan terjadi. Kontrol *derivative* hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula menyebabkan kontroler *derivative* tidak dapat dipakai sendiri.

III. METODE PENELITIAN

3.1 Diagram Blok Sistem



Gambar 3. Diagram Blok Sistem

Input :

Untuk mendeteksi adanya objek *shagai* pada titik pengambilan, robot ini menggunakan sensor kamera Pixy CMUCam5. Sensor akan mendeteksi koordinat objek yang akan dibaca oleh mikrokontroler. *Push button* berfungsi sebagai *input*-an untuk mengatur fungsi- fungsi menu yang ada pada robot.

Kontrol :

Pada kontrol robot ini menggunakan Arduino Mega 2560 untuk memproses data hasil pembacaan sensor yang selanjutnya akan memberikan sinyal pada motor dc untuk robot, selanjutnya melakukan manuver menuju titik pengambilan objek yang telah terbaca oleh sensor kamera.

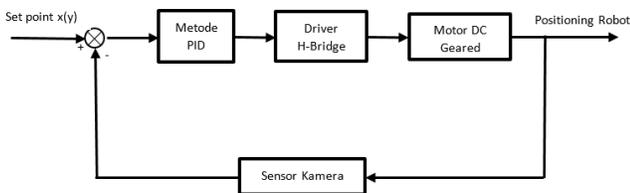
Output :

Driver H-Bridge digunakan untuk mengubah sinyal PWM dari mikrokontroler menjadi tegangan DC yang digunakan untuk men-*supply* motor. LCD digunakan untuk menampilkan hasil pembacaan sensor dan untuk menampilkan menu untuk mengatur robot.

Battery :

Sebagai *supply* robot sebesar 24V DC.

3.2 Diagram Blok Kontrol PID



Gambar 4. Diagram Blok Kontrol PID

Penjelasan dari diagram blok kontrol gambar 5 adalah sebagai berikut :

1. *Input* berasal dari hasil pendeteksian objek oleh sensor kamera.
2. Mikrokontroler akan mengolah data – data yang didapat dari sensor menjadi sebuah instruksi gerakan pada robot dengan kontrol PID.
3. *Driver H-Bridge* digunakan untuk mengatur kecepatan putaran motor DC dengan merubah

sinyal PWM yang dihasilkan oleh mikrokontroler agar dapat stabil.

4. Motor DC digunakan untuk menggerakkan roda robot.
5. Gerakan robot yang akan dikendalikan berupa kecepatan dan manuver dari tiga buah motor DC untuk mencapai titik jangkauan pengambilan objek.

3.4 Prinsip Kerja

Robot mendeteksi objek berupa objek *shagai* menggunakan sensor kamera Pixy CMUCam 5. Robot akan mencari titik koordinat dari objek tersebut melalui sensor kamera. Hasil tangkapan gambar dan koordinat dari sensor akan dikirim ke mikrokontroler. Setelah robot mengetahui titik koordinat objek, dengan menggunakan kontrol PID robot akan berusaha memposisikan objek berada di titik jangkauan robot dalam mengambil objek. Robot akan bermanuver untuk menuju titik jangkauan pengambilan objek. Apabila target berada di luar jangkauan pandang kamera maka robot akan melakukan *scanning*. Proses *scanning* adalah proses dimana robot akan mencari objek dengan warna yang sudah ditentukan, maka robot akan bergerak memposisikan objek berada pada titik jangkauan pengambilan objek.

3.7 Perancangan PID

Pada perancangan robot ini untuk *setpoint* kontrol PID adalah titik tengah dari koordinat sumbu x pada sensor kamera. Resolusi yang digunakan pada sensor kamera yaitu 320x210 dimana nilai resolusi ini adalah nilai resolusi maksimal pada koordinat x dan y pada sensor kamera Pixy CMUCam 5. Untuk mendapatkan nilai *setpoint* yaitu dengan cara membagi 2 nilai koordinat sumbu x, tetapi koordinat sumbu x sensor kamera yang semula 0-360 terlebih dahulu di konversikan menjadi 0-100 sehingga nilai *setpoint* yang didapatkan adalah:

$$Setpoint = \text{Koordinat } X / 2 \tag{1}$$

Sehingga : $Setpoint = 100 / 2 = 50$

Sedangkan untuk nilai *error* merupakan selisih dari *setpoint* dengan titik tengah dari koordinat sumbu x sensor kamera Pixy CMUCam 5.

Pada kontroler PID terdapat 3 parameter konstanta yang mempengaruhi kinerja pada kontroler ini yaitu konstanta *proportional* (k_p), konstanta *integral* (k_i) dan konstanta *derivative* (k_d).

Konstanta *proportional* berfungsi sebagai penguat dari sinyal *error* sehingga akan mempercepat respon sistem untuk mencapai kestabilan. Penerapan pada sistem adalah sebagai berikut.

$$P = error * k_p; \tag{2}$$

Pada kontrol *integral* bertujuan untuk mengurangi *offset* yang ditimbulkan oleh kontrol *proportional* yaitu dengan cara meng-*integral*-kan nilai *error* atau dengan kata lain menjumlahkan nilai *error* mulai dari *error* pertama hingga *error* ke-k (???). Dapat disimpulkan bahwa *error* yang sebelumnya dijumlahkan dengan *error-error* yang sebelumnya hingga *error* yang sekarang. Berikut penerapannya pada sistem.

$$I = (error + lastError) * kI * deltaTime; \quad (3)$$

Sedangkan kontrol *derivative* digunakan sebagai pengendali laju karena *output* kontroler sebanding dengan laju dari perubahan sinyal *error*, kontrol *derivative* juga sering disebut sebagai fungsi turunan sehingga dapat disimpulkan bahwa *error* yang sekarang dikurangi *error* yang sebelumnya. Berikut penerapannya pada sistem.

$$D = (error - lastError) * kD / deltaTime; \quad (4)$$

Sehingga penerapan rumus PID pada sistem adalah sebagai berikut :

$$PID = P+I+D; \quad (5)$$

IV. HASIL DAN ANALISA

4.1 Pengujian LCD 16x2

Pengujian ini bertujuan untuk memastikan apakah LCD bisa digunakan atau tidak. Pengujian pada *display* LCD 16x2 dilakukan untuk memastikan apakah lcd dapat menampilkan karakter sesuai dengan program yang ditanam pada mikrokontroler.



Gambar 5. Pengujian LCD

Dari hasil pengujian diatas didapatkan hasil bahwa LCD 16x2 yang dipakai dapat berfungsi sebagaimana mestinya.

4.2 Pengujian Tombol

Pengujian pada tombol dilakukan untuk mengetahui apakah tombol dapat berfungsi ketika ditekan sesuai dengan program yang tertanam didalam mikrokontroler Arduino. Terdapat 7 tombol yang terhubung pada mikrokontroler Arduino dimana setiap tombol memiliki fungsi masing masing-masing.

Tabel 1. Pin Tombol dan Fungsinya

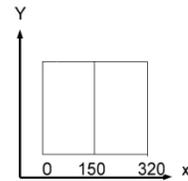
Pin	Fungsi
A9	Untuk menyimpan nilai konstanta PID
A8	Untuk mengurangi nilai konstanta PID
A7	Untuk menambah nilai konstanta PID
A6	Untuk menggeser ke kiri nilai konstanta PID
A5	Untuk menggeser ke kanan nilai konstanta PID
A4	Untuk mengatur ulang PID

Fungsi dari tombol-tombol tersebut dapat bekerja jika mendapatkan logika '1' dikarenakan tombol diberikan *pull-up*

external sehingga saat tidak ditekan pin tombol akan berlogika '0'.

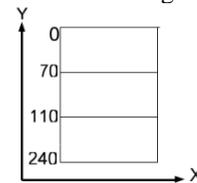
4.3 Pengujian Sensor Kamera

Kamera Pixy CMUCam5 menggunakan algoritma berbasis warna untuk mendeteksi benda - benda. Pixy akan menghitung warna dan saturasi setiap *pixel* RGB dari sensor gambar dan menggunakannya sebagai parameter penyarangan warna. Resolusi kamera Pixy CMUCam5 1280x800 dengan membagi koordinat ruang pandang kamera sesuai resolusinya menjadi 2 bidang X untuk menentikan pergerakan tiap motor dengan ketentuan resolusi 0-320 dan juga menjadi 3 bidang Y dengan ketentuan 0-240.



Gambar 6. Pembagian Koordinat X pada Ruang Kamera Pixy CMUCam5

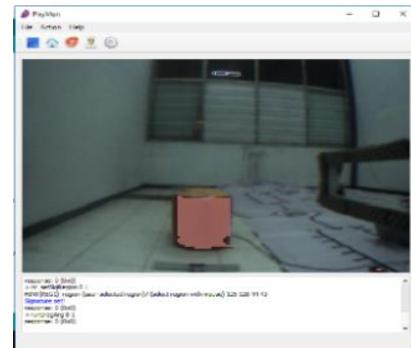
Pembagian resolusi pandangan kamera ini bertujuan untuk digunakan membatasi pergerakan dari robot dengan ketentuan jika pembacaan koordinat antara 150-320 maka robot akan bergerak ke kanan dan jika pembacaan koordinat kamera antara 0-150 maka robot bergerak ke kiri.



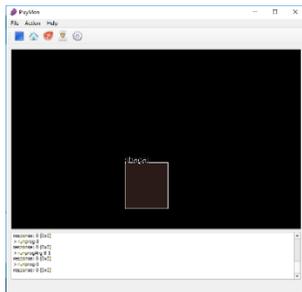
Gambar 7. Pembagian Koordinat Y pada Ruang Kamera Pixy CMUCam5

Pembagian resolusi pandangan kamera ini bertujuan untuk digunakan membatasi pergerakan dari robot dengan ketentuan jika pembacaan koordinat antara 0-70 maka robot akan maju, jika pembacaan koordinat 110-240 maka robot akan mundur, dan jika pembacaan koordinat kamera antara 70-110 maka robot akan berhenti.

Berikut adalah tampilan gambar dari sensor kamera CMUCam5 :



Gambar 8. Pengambilan Gambar pada PixyMon



Gambar 9. Pengambilan Detail Gambar Signature pada PixyMon

4.4 Pengujian Driver Motor

Driver motor BTN7970 digunakan sebagai pemicu dari arah pergerakan motor DC. Pada driver ini memiliki 1 buah pin enable, 2 buah pin input, dan 2 buah pin output yang mana memungkinkan bagi kita untuk mengontrol 1 buah motor DC sehingga pada alat ini digunakan 3 buah driver motor untuk mengontrol 3 buah motor DC. Pengujian dilakukan dengan cara memberikan input-an berupa logika ‘1’ dan ‘0’ pada masing-masing pin input-an secara bergantian kemudian melihat perubahan pada pergerakan motor, yang terlebih dahulu sudah disambungkan pada pin-pin output dari driver. Berikut adalah tabel hasil pengujian rangkaian driver motor BTN7970:

Tabel 2. Pengujian Driver Motor

Pin BTN7970			
Enable	Input 3	Input 4	Kondisi Motor
0	0	0	Diam
0	0	1	Diam
0	1	0	Diam
0	1	1	Diam
1	1	0	Putar Kanan
1	0	1	Putar Kiri
1	0	0	Diam
1	1	1	Diam

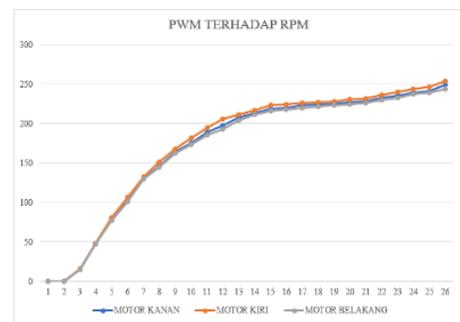
Untuk pengontrol driver diberi logika ‘1’ (high) atau ‘0’ (low) pada mikrokontroler Arduino dan diberikan nilai PWM pada pin enable. Dimana apabila logika pada pin enable ‘0’ (low) maka motor akan tetap diam meskipun logika pada input-an diganti-ganti. Pin enable ini bersifat aktif high ‘1’ sehingga aktif akan bila diberikan logika 255.

4.5 Pengujian Kecepatan Motor

Pengujian kecepatan motor ini untuk mengetahui kecepatan pada setiap motor dengan alat ukur tachometer. Setiap motor diberikan nilai pwm 0 sampai 250 kemudian diukur dan diamati nilai kecepatan (rpm) masing-masing motor pada digital tachometer. Berikut adalah tabel hasil pengujian kecepatan motor :

Tabel 3. Pengujian Kecepatan Motor

PWM	RPM		
	Motor depan kanan	Motor depan kiri	Motor belakang
0	0	0	0
10	0	0	0
20	16	16	15
30	47	48	47
40	79	81	77
50	104	107	101
60	131	133	130
70	146	151	145
80	164	168	162
90	175	182	173
100	189	195	185
110	197	206	193
120	208	211	204
130	213	217	211
140	219	223	216
150	220	224	218
160	223	226	220
170	224	227	221
180	225	228	223
190	227	231	224
200	228	232	226
210	233	236	230
220	235	240	233
230	239	244	237
240	241	246	239
250	249	254	244



Gambar 10. Grafik PWM Motor Terhadap RPM

Dari data percobaan 3 buah motor diatas dapat disimpulkan bahwa semua motor masih dapat digunakan. Hal ini dapat dilihat dari keluaran RPM dari ketiga motor yang memiliki selisih nilai keluaran RPM sangat kecil.

4.6 Pengujian PWM Sinyal sebagai Penggerak Motor

PWM yang digunakan pada *mobile* robot ini adalah PWM dengan tipe 8 bit atau nilai maksimal PWMnya adalah 255. Semakin besar nilai masukan PWM maka kecepatan putaran motor semakin cepat. Karena 0 = 0% *duty cycle* yang berarti motor tidak akan berputar dan 255 = 100% *duty cycle* yang berarti motor akan berputar secara maksimal. Apabila kita analisa melalui rumus *levelling*, maka perhitungannya sebagai berikut:

$$PWM = \frac{255}{DC\%} \times \% \tag{6}$$

Keterangan :

%DC = Persen *Duty Cycle*

Untuk *mapping* keluaran pada program berikut adalah hasil perhitungannya :

Tabel 4. Tabel Hasil Perhitungan

PWM	%DC
100	255
90	229,5
80	204
70	178,5
60	153
50	127,5
40	102
30	76,5
20	51
10	25,5
0	0

Pada *mobile* robot ini menggunakan nilai masukan PWM stabil 80. *Duty cycle* yang digunakan sebesar 204. Digunakannya nilai PWM 80 karena apabila semakin besar nilai masukan PWM maka laju motor akan semakin cepat hal ini akan berpengaruh pada pembacaan sensor kamera pada *mobile* robot karena *fps* sensor kamera yang rendah.

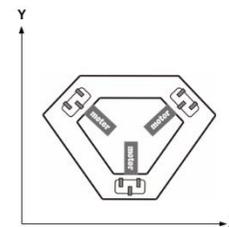
4.7 Pengujian Gerakan Robot

Di dalam proses pengendalian gerak robot menggunakan 3 roda *omni wheel* ini terdapat cara khusus dalam mengkombinasikan gerak antar motor agar dapat bermanuver sesuai yang diharapkan. Robot ini di desain dengan bentuk sasis *holonomic* agar robot dapat memfungsikan roda *omni wheel* secara maksimal. Berikut ini tabel proses gerak robot dan bentuk desain sasis *holonomic*.

Pada tabel 5 merupakan tabel kebenaran dari proses gerak motor yang mempengaruhi arah gerak dari robot. Untuk acuan arah geraknya berdasarkan titik koordinat X dan Y.

Tabel 5. Proses Gerak Robot

Koordinat	Gerak Motor			Arah Gerak
	Motor Depan Kanan	Motor Depan Kiri	Motor Belakang	
Koordinat X 0 -150	CW	CW	CCW	Kiri (-x)
Koordinat X 150 - 320	CCW	CCW	CW	Kanan (x)
Koordinat Y 0 -70	CW	CCW	-	Maju (y)
Koordinat Y 110 - 240	CCW	CW	-	Mundur (-y)
Koordinat Y 70 - 110	-	-	-	Berhenti (x,y)

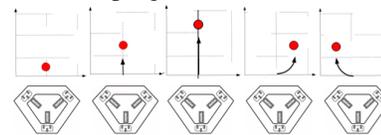


Gambar 11. Desain Sasis *Holonomic*

Pada gambar 11 arah panah menunjukkan bahwa sisi tersebut merupakan bagian depan robot. Dalam penggunaan roda *omni wheel* dengan 3 *wd*, penempatan motor maupun roda haruslah berbentuk. Hal tersebut dikarenakan keefektifannya dalam menggerakkan motor, ditambah lagi beban dalam menggerakkan robot dapat ditanggung oleh ke-3 buah motor, sehingga masing-masing motor tidak bekerja terlalu keras.

4.8 Pengujian Pembacaan Sensor Kamera Terhadap Gerakan Robot

Pergerakan tiap motor mengacu pada data yang dikirimkan dari sensor kamera Pixy CMUCam5 dengan melalui pembagian koordinat x dan y pada pembacaan objek yang sesuai dengan instruksi program.



Gambar 12. Simulasi Pergerakan Robot

Tabel 6. Posisi Gerakan Robot Mengikuti Objek

Letak Objek	Gerakan Robot
Koordinat X 0 – 150	Robot Kekiri
Koordinat X 150 - 320	Robot Kekanan
Koordinat Y 0 – 70	Robot Maju
Koordinat Y 70 - 100	Robot Berhenti
Koordinat 110 – 240	Robot Mundur

Pada uji pembacaan sensor kamera terhadap gerakan robot sudah berjalan sesuai yang diinginkan.

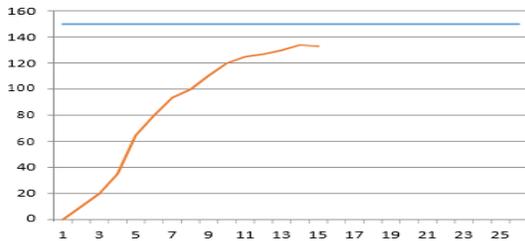
4.9 Pengujian PID

Pengujian kontrol PID dilakukan dengan tujuan agar pergerakan robot dapat stabil pada *setpoint* yang telah ditentukan sebelumnya yaitu robot dapat memposisikan objek berapa pada titik tengah pengambilan objek. Pada perancangan kontrol PID telah dilakukan *tuning* nilai konstanta *proportional*, *integral*, dan *derivative* dimana *tuning* PID menggunakan metode *trial and error*.

Selanjutnya untuk mendapatkan respon yang diinginkan maka digunakan metode *handtuning* (*trial and error*). Metode ini dilakukan dengan cara merubah nilai parameter K_p , K_i , atau K_d sampai menemukan respon sistem yang diinginkan.

Grafik Trial 1 :

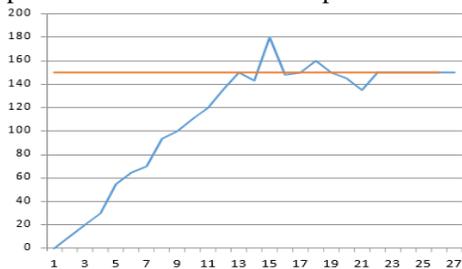
- $K_p = 3$; $K_i = 4$; $K_d = 8$
- Respon Sistem *Set Point* Terhadap Waktu



Gambar 13. Trial PID Pertama

Grafik Trial 2 :

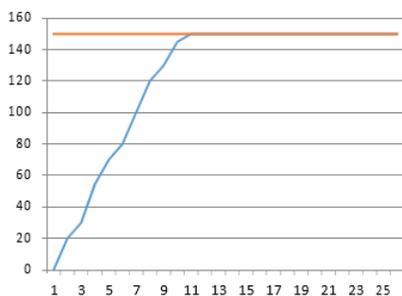
- $K_p = 4$; $K_i = 7$; $K_d = 26$
- Respon Sistem *Set Point* Terhadap Waktu



Gambar 14. Trial PID Kedua

Grafik Trial 2 :

- $K_p = 4$; $K_i = 7$; $K_d = 15$
- Respon Sistem *Set Point* Terhadap Waktu



Gambar 15. Trial PID Ketiga

Dari hasil percobaan PID yang telah dilakukan dengan mengkombinasikan parameter-parameter K_p , K_i , K_d dapat disimpulkan bahwasanya kombinasi dari ke tiga parameter tersebut sangatlah memengaruhi kinerja dari sistem. Proses *tuning* yang tepat akan menghasilkan sistem yang stabil. Seperti dalam mencari parameter P, dimana fungsinya untuk mempercepat respon. Tentunya perlu nilai P yang besar agar respon semakin cepat, namun dampak negatifnya juga mengikuti, yaitu osilasi pada sistem akan mempengaruhi kestabilan sistem tersebut. Oleh karena itu perlu ditambahkan pula parameter D dimana osilasi sistem akan diredam. Dengan nilai yang tepat pula akan menghasilkan sistem yang stabil pula. Pada sistem ini menambahkan parameter I bersifat operasional. Dari fungsinya untuk mempercepat hilangnya *offset*. Untuk pemberian nilai *integral* bergantung dari kestabilan robot ketika hanya diberikan nilai *proportional* dan *derivative*. Dari hasil *tuning* yang menuju stabil yaitu pada *trial 3* dengan nilai $K_p = 4$; $K_i = 7$; $K_d = 15$. Pada sistem ini robot dapat berjalan dengan baik memposisikan pada titik jangkauan pengambilan objek pada jangkauan pandang kamera.

V. PENUTUP

5.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem yang telah dibuat oleh robot sudah berjalan dengan baik dan sesuai rencana. Berikut ini adalah kesimpulan yang didapat dari percobaan alat :

1. Penggunaan sensor kamera Pixy CMUCam5 sebagai solusi penangkapan objek pada robot terbukti sangat akurat karena penangkapan objek dapat dilakukan secara *real time*.
2. Dengan menggunakan sensor kamera Pixy CMUCam5 robot dapat mengikuti benda secara *real time*, tetapi kecepatan motor belum stabil dalam memposisikan objek, hal ini dianggap wajar dikarenakan beban robot yang relatif berat dan jenis roda yang digunakan.
3. Berdasarkan hasil pengujian control PID pada robot, robot dapat memposisikan objek berada pada titik jangkauan pengambilan objek dengan tepat menggunakan metode *tuning trial and error* dengan nilai $K_p = 4$; $K_i = 7$; $K_d = 15$.

5.2 Saran

Perlu adanya perbaikan dan penyempurnaan agar robot dapat bekerja secara optimal. Ada beberapa hal yang disarankan oleh penulis untuk perbaikan dan penyempurnaan :

1. Selain dapat digunakan pada robot 3 motor, kontrol PID dapat juga digunakan untuk control robot 4 motor untuk menambah respon kecepatan pada motor.
2. Sebaiknya pada robot ini ditambahkan lagi sensor seperti sensor garis pada saat perlombaan agar manuver robot dapat beroperasi secara maksimal
3. Sebaiknya apabila digunakan untuk perlombaan, sensor kamera sebaiknya diganti karena sensor kamera Pixy CMUCam 5 memiliki respon *fps* yang lambat.

4. Selain dengan kontrol PID sebagai controller pergerakan robot dapat diterapkan juga menggunakan metode kontrol lain seperti metode logika *fuzzy*.

DAFTAR PUSTAKA

- [1] Aydan, Mohammad Habib. 2015. "Implementasi Kontrol PID pada Motor DC Sebagai Penggerak Robot Beroda 4WS Omni Wheels dengan Metode Wall Following". Skripsi Jurusan Teknik Elektro. Malang : Politeknik Negeri Malang.
- [2] Kurniawan, Restu. 2017. "Implementasi Kontrol PID Pada *Object Tracking* Menggunakan Sensor Kamera. Skripsi Jurusan Teknik Elektro. Malang : Politeknik Negeri Malang.
- [3] Lubis, Mohammad Agustinus Saputra Anang. 2016. "*Autonomous Hover dan Steering Line Following "Robot Hybrid"* pada Abu Robocon 2016 Dengan Metode PID". Skripsi Program Studi Teknik Elektro. Malang : Politeknik Negeri Malang.
- [4] Rizal, Muhammad, Maulana Akbar, 2017. Implementasi Kontrol PID Pada Kecepatan Motor Differential Drive Line Following ABU ROBOCON 2017. Malang : Politeknik Negeri Malang.
- [5] ABU Asia Pasific Robot Contest 2019. *Great Urtuu* (<https://robu.in/abu-https://robu.in/abu-robocon-2019-mongolia-rulebook-and-theme-video/>)
- [6] DatasheetKameraPixyCMUCam5