Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

Simulasi Sistem Navigasi Berbasis-Perilaku dan Kontroler PID pada Manuver Mobile Robot

Edy Supriyadi, Aktsar Abdikar dan Nizar Rosyidi A.S

Jurusan Teknik Elektro, Fakultas Teknologi Industri – Institut Sains dan Teknologi Nasional Jakarta

Abstrak — Merancang dan mengimplementasikan algoritma kendali *behavior based* (berbasis perilaku) pada mobile robot beroda tipe *differential steering* dan kontroler PID (Proporsional, Integral, Diferensial) pada manuver robot ini. Tugas robot ini adalah menyusuri dan mencari target yang diletakan secara acak (*random*) pada suatu citra (*environment*) yang terdiri dari dua: citra sederhana dan citra labirin. Kendali berbasis perilaku menggunakan modul-modul perilaku yang bekerja secara bersamaan membentuk perilaku robot yang ingin dicapai.Masing-masing perilaku sifatnya independen, memiliki hubungan langsung dengan sensor dan aktuator. Selain menerapkan kendali berbasis perilaku laporan penelitian ini juga merancang dan mengimplementasikan kontroler PID pada manuver robot *maze* terhadap jarak dinding citra. Kontroler PID bertujuan untuk memuluskan pergerakan robot saat menelusur ruangan/lorong citra dan labirin.Dengan bantuan kontroler PID robot *maze* mampu bermanuver dengan aman, halus, responsif dan cepat. Hasil parameter kontroler PID yang dicapai dari penelitian laporan penelitian ini diperoleh dari hasil *tuning* eksperimen metode *Trial and Error* dengan hasil Kp = 17, Ki = 5 dan Kd = 0 pada citra sederhana dan Kp=10, Ki= 2, dan Kd=0 pada citra labirin.

Abstract – Design and implement a control algorithms consisting of PID and behaviour-based in a differential mobile robot. The robot's goal is to explore the map (environmant) and find the randomly placed target. The environment is provided in two types: simple environment and labirynth environment. The behaviour-based control algorithm utilizes behaviour moduls which are together forming the desired entirety control of the robot. The respective behaviour is working independently and alternately) and has a direct connection to the sensors and actuators. This paper also designing and implementing PID controller to track the manuever of the robot in order to explore the map/environment without touching or hitting the wall. The PID control allows the robot to explore the map in a safe, smooth, responsive, and quick manner. The resulting PID parameters are derived in a "Trial and Error" tuning method that is: Kp: 17, Ki = 5, and Kd = 0 in a simple environment; and Kp = 10, Ki = 2, and Kd = 0 in a labirynth environment.

I. PENDAHULUAN

Dengan sistem yang semakin kompleks pada dunia nyata (*real world*), dibutuhkan perkembangan penerapan dalam sistem kontrol modern terutama di bidang robotika.Penerapan ini diharapkan mampu mengatasi masalah-masalah *real world* yang terjadi pada industri-industri dan lingkungan lainnya yang membutuhkan robotika.

Skripsi ini merancang dan mengimplementasikan kendali PID (Proporsional, Integral, Diferensial) pada manuver mobile robot dengan menambahkan suatu perilaku untuk beradaptasi pada lingkungan *real world*. Tugas mobile robot ini adalah menyusuri dan mencari target yang diletakkan secara acak pada suatu peta atau *environment* dengan rintangan-rintangan yang bervariasi.Kemampuan khusus mobile robot ini adalah sifat adaptasi mobile robot ini terhadap variasi rintangan dengan menerapkan perilaku-perilaku berdasarkan variasi rintangan tersebut. Kontroler PID bertujuan untuk menjaga robot mengarah ke tujuan, sehingga juga akan memuluskan pergerakan robot saat menulusuri ruangan/lorong peta atau *environment*. Dengan bantuan kontroler PID mobile robot mampu bermanuver dengan aman, halus, responsif, dan cepat.

Simulasi pemrograman berbahasa *python* dipilih untuk menyelesaikan skripsi ini. Perangkat lunak *spyder* dipilih sebagai *compiler* untuk menjalankan berbagai kode pengontrolan robot, perangkat lunak ini akan menjalankan (*run*) program atau aplikasi bernama *Qtsimiam* yang merupakan versi ekstensi dari program matlab sim.i.am.

II. TEORI PENUNJANG

A. Bahasa Pemrograman Python

Python merupakan bahasa *scripting* tingkat tinggi dan berorientasi objek.*Python* dirancang untuk mudah dibaca. Bahasa pemrograman ini merupakan bahasa yang sangat sering menggunakan kata-kata kunci berbahasa inggris, tidak seperti bahasa lain yang banyak menggunakan tanda-tanda baca, bahasa ini juga memiliki lebih sedikit konstruksi-konstruksi sintaktikal dibandingkan dengan bahasa pemrograman lainnya. *Python* dapat beroperasi di hampir semua platform , seperti UNIX, Mac, Windows , OS/2, ataupun yang lain.

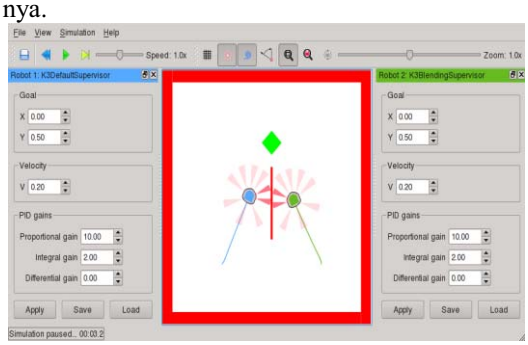
Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

1) Spyder

Spyder merupakan platform *integrated development environment* (IDE) untuk tujuan pemrograman ilmiah dalam bahasa *Python*. Spyder mengintegrasikan Numpy, Scipy, Matpotlib, dan IPython, dan juga perangkat lunak *open source* lainnya.Software ini dikeluarkan di bawah lisensi *Massachusetts Institute of Technology* (MIT).

2) Qtsimiam

Qtsimiam merupakan Sim.I.Am matlab versi pemrograman *python*. Terinspirasi dari Jean-Pierre de la Croix yang merupakan dosen pemrograman robot di Georgia Institute of Technology. Aplikasi ini harus dijalankan oleh versi *python 2.7* ke atas dan membutuhkan dua *library* untuk kebutuhan pengkodeannya yaitu *Numpy* untuk keperluan pengkodean matematis dan *PyQT* untuk Antarmuka Pengguna Grafis (GUI) nya.



Gambar 2.1. Antarmuka Pengguna Grafis (GUI) Qtsimiam

B. Mobile Robot Diferensial

Robot diferensial merupakan mobile robot yang pergerakannya didasarkan pada dua roda yang digerakkan secara terpisah yang ditempatkan pada sisi-sisi mobile robot.Robot diferensial ini dapat mengubah arah dengan memvariasikan laju relatif dari rotasi roda-rodanya dan karenanya tidak membutuhkan gerakan pengemudian tambahan.

1.) Perumusan Kinematika Mobile Robot

Secara ringkas hubungan antara kecepatan linear (v) dan kecepatan angular (ω) dengan kecepatan motor penggerak (V_r dan V_l) dapat dituliskan seperti di bawah ini :

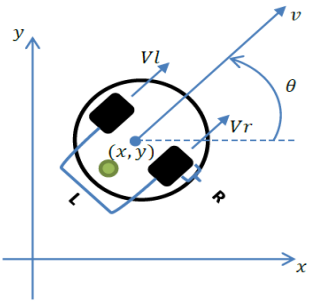
$$x' = v\cos(\theta) = \frac{R}{2}(v_r + v_l)\cos(\theta) \quad (2.1)$$

$$y' = v\sin(\theta) = \frac{R}{2}(v_r + v_l)\sin(\theta) \quad (2.2)$$

di mana,

$$\theta' = \omega = \frac{R}{L}(v_r - v_l)$$

Dengan menggunakan persamaan (2.1) dan (2.2) mobile robot diferensial dapat dikendalikan dengan hanya menentukan parameter umum seperti $v$anω dari robot tersebut tanpa harus menentukan secara spesifik kecepatan dari masing-masing roda penggerak.



Gambar 2.2 Penggambaran mobile robot diferensial pada bidang kartesian

2.) *Khepera III*

Robot Khepera III diproduksi oleh "K-Team *Corporation*" dengan bantuan pengembangan dari *Distributed Intelligent Systems and Algorithms Laboratory*(DISAL). Robot ini memiliki diameter 12 cm, hal tersebut membuat robot ini sesuai untuk eksperimen multi-robot dalam ruangan. Robot ini menggunakan sistem penggerak diferensial dengan dua buah motor yang dapat dikendalikan secara independen. Khepera III menggunakan "Korebot *Platform*", diamana *platform*tersebut menjalankan *standard embedded* Linux *operating system* pada prosesor IntelXSCALE PXA-255 yang memiliki kecepatan 400 MHz. Sebuah *bus* ekspansi *stackable* memungkinkan penambahan untuk modul robot yang diinginkan.

3.)Odometri

Odometri adalah sebuah sistem atau model persamaan matematika yang berfungsi untuk mengetahui keberadaan robot berdasarkan pergerakan yang telah dilakukan oleh robot tersebut. Misalkan robot berada pada posisi $x = 0$, $y = 0$, dan $\theta = 0$, lalu robot berjalan lurus searah dengan sumbu x selama 3 detik dengan kecepatan 3 m/s, maka dapat kita perkirakan bahwa posisi robot selanjutnya (setelah berjalan selama 3 detik) ada pada koordinat 3, 0, 0 (x, y, θ), seperti itulah esensi dari metode Odometri.

Secara ringkas, persamaan-persamaan penting yang didapat dari metode odometri untuk digunakan pada pengkodean *python* adalah sebagai berikut :

$$\varphi r_{right} = d_{right} \quad (2.3)$$

$$\varphi r_{left} = d_{left} \quad (2.4)$$

$$d_{center} = \frac{d_{left} + d_{right}}{2} \quad (2.5)$$

$$\varphi = \frac{d_{right} - d_{left}}{d_{baseline}} \quad (2.6)$$

$$\theta' = \theta + \varphi$$

$$x' = x + d_{center}\cos(\theta) \quad (2.7)$$

$$y' = y + d_{center}\sin(\theta) \quad (2.8)$$

Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

C.Navigasi Mobile Robot

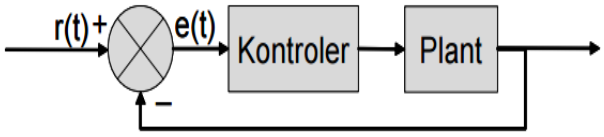
1.) Penghindaran Rintangan

Penghindaran rintangan robot berfokus pada mengubah trajektori robot sebagaimana diinformasikan oleh sensor-sensornya selama pergerakan robot ke tujuan.Pergerakan robot dihasilkan oleh fungsi pembacaan-pembacaan sensor di saat awal dan saat setelahnya dan posisi tujuan dan lokasi relatif terhadap posisi tujuan.Algoritma penghindaran rintangan bergantung pada perubahan derajat keberadaan dari *global peta* dan bergantung pada pengetahuan presisi robot dari lokasinya relatif terhadap peta.

2.) KONTROLER PID

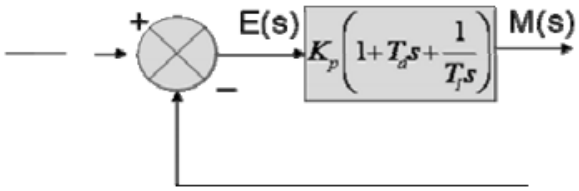
Teori Dasar Sistem Kontrol

Diagram blok sistem kontrol sederhana diperlihatkan pada **Gambar 2.3** Kontrol automatik membandingkan harga yang sebenarnya dari keluaran *plant* dengan harga yang diinginkan (r(t)), menentukan deviasi (e(t)), serta menghasilkan sinyal kontrol untuk mengurangi deviasi sampai nol atau sangat kecil. Cara kontroler automatik bekerja untuk menghasilkan sinyal kontrol disebut aksi pengontrolan.



Gambar 2.3 Diagram blok kontroler loop tertutup

Kontroler Proporsional-Integral-Derivatif (PID)



Gambar 2.4 Diagram blok kontroler PID

Kontroler PID merupakan gabungan dari kontroler proporsional, integral, dan derivatif. Persamaan kontroler dengan aksi gabungan diberikan oleh persamaan:

$$m(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt} \qquad (2.9)$$

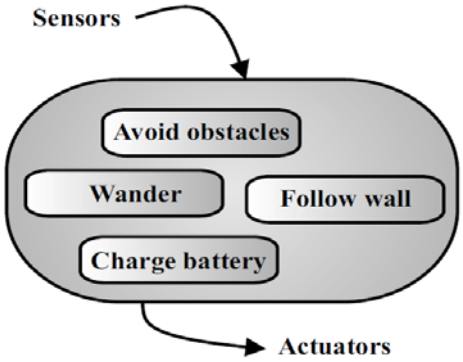
Kontroler PID memiliki keunggulan dibandingkan kontroler konvensional jenis lain karena karakteristik dari kontroler PID merupakan gabungan dari karakteristik dari kontroler-kontroler yang dimiliki oleh kontroler P, kontroler PD, dan kontroler PID.

D.Robotika Berbasis Perilaku (*Behaviour-Based Robotics*)

Robotika berbasis perilaku merupakan alternatif dari kecerdasan buatan (*Artificial Intelligence*), di mana perilaku intelijen dibuat dengan cara dari bawah ke atas, di mulai dari perilaku-perilaku yang sederhana, di mana di antaranya berjalan sama-sama dalam otak robot, memberikan saran-saran mengenai aksi-aksi apa yang harus diambil oleh robot. Untuk lebih jelasnya dapat dilihat pada Gambar 2.5.

Perilaku dan Aksi

Perilaku di sini secara sederhana didefinisikan sebagai serangkaian aksi yang dilakukan untuk mencapai suatu tujuan.Sebagai contoh, suatu perilaku penghindaran rintangan dapat terdiri dari aksi-aksi berhenti, belok, dan berjalan kembali dalam arah yang berbeda.



Gambar 2.5 Robotika Berbasis-perilaku

Perilaku pintar dan penalaran

Dalam robotika berbasis perilaku, perilaku pintar dapat didefinisikan sebagai kemampuan untuk bertahan dan untuk berjuang mencapai tujuan-tujuan lainnya dalam suatu lingkungan yang tak terstruktur.

Sifat-sifat robot berbasis-perilaku

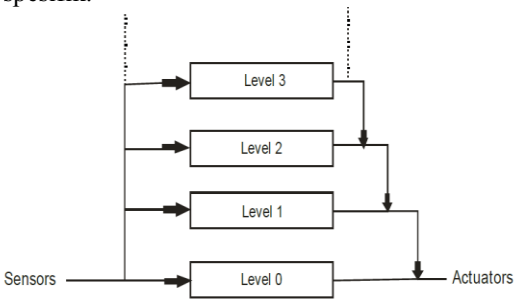
Robot-robot berbasis-perilaku yang berbeda umumnya memiliki sifat-sifat dasar tertentu yang serupa, meskipun tidak semua sifat tersebut terdapat pada robot-robot tersebut.Untuk awal, Robot-robot berbasis-perilaku dilengkapi dengan perilaku yang paling sederhana seperti penghindaran rintangan untuk bekerja pada lingkungan tak terstruktur. Perilaku-perilaku yang lebih rumit ditambahkan pada tahap berikutnya.Kedua, beberapa perilaku umumnya berada dalam operasi yang bersamaan, dan aksi terakhir dilakukan oleh robot mewakili salah satu pilihan antara saran-saran yg diberikan oleh berbagai perilaku, atau aksi rata-rata yang dibentuk dari aksi-aksi yang disarankan oleh beberapa perilaku.Ketiga, robot berbasis perilaku adalah sangat penting pada robot-robot otonom, yaitu robot-robot yang dapat bergerak secara bebas tanpa pengawasan langsung dari manusia. Yang terakhir, konsep dari letak merupakan prinsip dasar dari robot berbasis perilaku: Robot-robot berbasis-perilaku tidak membangun model-model *world* yang kompleks maupun abstrak. Melainkan, robot berbasis-perilaku

Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

umumnya disituasikan (beroperasi pada dunia nyata), dan banyak perilaku-perilaku dalam robot berbasis-perilaku yang reaktif, yaitu memiliki hubungan langsung antara sensor-sensor dan aktuator-aktuator.

Arsitektur Subsumption

Arsitektur *subsumption* adalah struktur BBR yang diusulkan oleh **Rodney Brooks [3]**. Dalam membangun robotnya, Rodney Brooks menguraikan permasalahan sistem kendali robot sesuai dengan manifestasi luar yang diinginkan oleh sistem kendali robot, tidak berdasarkan pada operasi internal dari sistem kendali robot sebagaimana yang telah dilakukan oleh beberapa peneliti sebelumnya. Oleh karena itu, Brooks mendefiniskan sejumlah level kompetensi pada *mobile robot* mandiri. Level kompetensi adalah spesifikasi informal dari sekelompok perilaku yang diinginkan robot bekerja pada semua lingkungan yang akan dihadapi. Level kompeten yang lebih tinggi menunjukkan kelompok perilaku yang lebih khusus/spesifik.



Gambar 2.6 Arsitektur Subsumsi

III. PERANCANGAN SISTEM

Kendali berbasis perilaku menggunakan modul-modul perilaku yang bekerja secara bersamaan membentuk perilaku robot agar tujuan (*goal*) dapat dicapai.Masing-masing perilaku sifatnya independen, memiliki hubungan langsung dengan sensor, kontroler, dan aktuator. Masing-masing perilaku juga dapat mengirim pesan pada perilaku lain atau menghambat output perilaku lainnya.

A. Kinematika Mobile Robot

Di sini akan di rumuskan parameter-parameter penting mobile robot agar dapat berjalan ke tujuan, yaitu:

- Posisi mobile robot (*x, y, θ*)
- Kecepatan roda kanan dan kiri (v_l dan v_r)

Persamaan matematisnya adalah sebagai berikut:

$$\dot{x} = \frac{R}{2}(v_r + v_l)cos\phi \quad (3.1)$$

$$\dot{y} = \frac{R}{2}(v_r + v_l)sin\phi \quad (3.2)$$

$$\dot{\phi} = \frac{R}{L}(v_r - v_l) \quad (3.3)$$

di mana:
v : kecepatan translasi (m/s)
ω : kecepatan sudut (rad/s)
R : jari-jari roda (m)
(x, y) : posisi robot pada sistem koordinat
L : Jarak antara kedua roda (m)
ϕ : sudut arah mobile robot (rad)
Sehingga untuk kecepatan roda kanan (v_r) dan roda kiri (v_l) diperoleh,

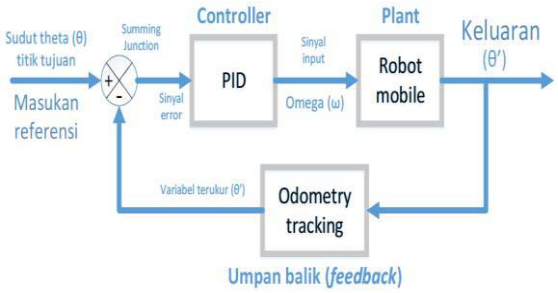
$$v_r = \frac{2v + \omega L}{2R} \quad (3.7)$$

$$v_l = \frac{2v - \omega L}{2R} \quad (3.8)$$

B. Citra Lingkungan (Environment)

Citra atau *environment* merupakan tempat atau peta untuk ditelusuri oleh mobile robot. Citra ini berisi jalur-jalur, rintangan (*obstacles*), tempat tujuan (*goal*), dll. Citra ini harus terintegrasi (secara pemrograman) dengan mobile robot jika ingin digunakan dalam simulasi, agar mobile robot dapat menelusuri dan mendeteksi setiap rintangan.

C. Kontrol PID Pada Mobile Robot



Gambar 3.1 Diagram blok implementasi PID pada mobile robot

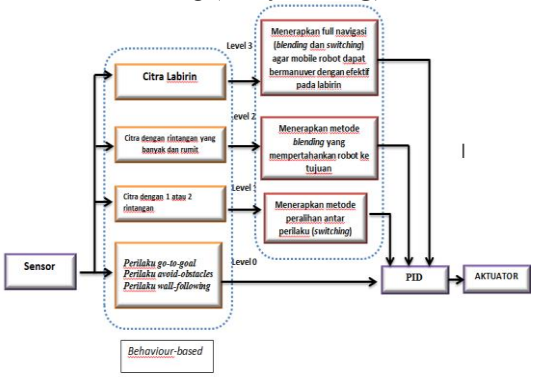
Dari Gambar 3.1. dapat dijelaskan bahwa masukan referensi atau target yang harus dicapai oleh robot adalah berupa nilai sudut dari titik tujuan di mana titik tujuan tersebut dapat berupa titik koordinat robot pemimpin ataupun jalur lintasan dari robot pemipin. Pada setiap siklusnya, sistem akan melakukan pembacaan nilai parameter yang dikendalikan (parameter *θ'*) dengan menggunakan metode odometri. Parameter tersebut selanjutnya akan dibandingkan nilainya dengan nilai parameter masukan referensi. Jika nilai antara variable terukur (*feedback*) tidak sama dengan nilai masukan referensi, maka blok pengendali (PID *controller*) akan melakukan perhitungan dan mengirimkan hasil perhitungan tersebut sebagai sinyal masukan untuk *plant* agar kondisi *plant* dapat berubah sehingga mengubah nilai parameter yang dikendalikan hingga mencapai kondisi yang diinginkan (nilai parameter output (*θ'*) sama dengan nilai parameter masukan referensi (*θ*).

D. Perancangan Perilaku

Aksi-aksi yang diperlukan untuk mengatur navigasi mobile robot di dalam suatu citra adalah:

Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

- Mengenali target (*goal*).
- Bernavigasi menuju target.
- Bernavigasi secara aman (tanpa menabrak atau menyentuh rintangan).
- Menelusuri dinding (*wall following*).



Gambar 3.1. Rancangan berbasis perilaku (behaviour-based) pada navigasi mobile robot

1.) Kontroler Berjalan Menuju Tujuan (go-to-goal Behaviour)

Secara dasar, untuk mengemudikan mobile robot menuju tujuan dideskripsikan dengan parameter-parameter berikut ini:

(x,y) : posisi mobile robot awal (pada sistem koordinat)
ϕ/ω : posisi sudut mobile robot terhadap tujuan (rad)
ϕ_d : posisi sudut yang diinginkan terhadap tujuan (rad)
e (error) : selisih antara posisi yang diinginkan dan posisi sebenarnya (rad)

$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\phi = \omega$$
$$\phi_d = arc\tan\left(\frac{y_g - y)}{x_g - x}\right)$$
$$e = \phi_d - \phi$$
$$\omega = PID\ (e)$$

2.) Kontroler Penghindaran Rintangan (avoid-obstacles behaviour)

Untuk membentuk kontroler penghindaran rintangan, perlu dilakukan suatu pembentukan vektor dalam kerangka referensi robot yang mengarahkan robot untuk menghindar.

3.) Kontroler Penelusuran Tembok (Wall Following Behaviour)

Sensor-sensor IR akan digunakan untuk mendeteksi rintangan dan membentuk vektor yang tegak lurus terhadap tembok (*wall*). Pada Gambar 3.8, vektor $\boldsymbol{u_{fw,t}}$ diilustrasikan dengan warna magenta.

E. SUPERVISOR

Supervisor merupakan tempat untuk menaruh kontroler-kontroler perilaku agar dapat diterapkan pada citra (*environment*) yang sesuai. Supervisor-supervisor yang akan digunakan dalam simulasi, yaitu:

- *Blending*
- *Switching*
- Campuran *blending* dan *switching* (*mix behaviour*)
- Odometri

3) 3.5.1. Odometri

Odometri diimplementasikan pada robot sehingga dapat diketahui posisi mobile robot (x, y, θ) setelah jarak yang ditempuh oleh tiap roda mobile robot. Pemrograman menyimpan posisi yang dihitung odometri (x, y) dan θ setelah mobile robot berjalan dari posisi awal (x_0, y_0, θ_0). Lalu nanti supervisor akan memperbaharui perkiraan posisi yang dihitung.

Dari persamaan pada pembahasan di bab II tentang Odometri, persamaan-persamaan berikut diambil untuk menjadi parameter-parameter yang dibutuhkan pada pengkodean, yaitu:

$$d_{center} = \frac{d_{left} + d_{right}}{2}$$
$$\varphi = \frac{d_{right} - d_{left}}{d_{baseline}}$$
$$\theta' = \theta + \varphi$$
$$x' = x + d_{center}\cos(\theta)$$
$$y' = y + d_{center}\sin(\theta)$$

Nilai d_{left} dan d_{right} didapatkan dengan perhitungan sebagai berikut:

$$\Delta tick_{right} = tick'_{right} - tick_{right}$$
$$\Delta tick_{left} = tick'_{left} - tick_{left}$$
$$D = 2\pi R\frac{\Delta tick}{N}$$

Di mana,
tick : 2π derajat

1.) Supervisor Blending

Supervisor ini mengimplementasikan tipe pertama dari mekanisme arbitrasi antara dua kontroler yaitu *blending* (mencampurkan).Supervisor ini efektif digunakan pada citra sederhana dengan berbagai rintangan yang tidak menyerupai tembok (labirin).Supervisor ini menyatukan dua kontroler kontroler *go-to-goal* dan *avoid-obstacles* menjadi satu kontroler agar bekerja dalam satu waktu secara bersamaan. Berikut merupakan perancangannya:

U_{ao} : Vektor penghindaran rintangan (*avoid obstacles*)
U_{gtg} : Vektor menuju tujuan (*go-to-goal*)

2.) Peralihan antara perilaku-perilaku (switching)

Tipe kedua dari mekanisme arbitrasi adalah peralihan (*switching*). Daripada mengeksekusi kedua kontroler (*go-to-goal* dan *avoid-obstacles*) secara bersamaan, supervisor ini hanya akan mengeksekusi satu kontroler pada satu waktu, namun setiap kontroler akan diterapkan bergantian bergantung pada kondisinya.

Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

3.) Mencampurkan Perilaku blending dan switching (mixing behaviour)

Keuntungan kontroler blending adalah dapat mencampurkan secara mulus dua kontroler (*go-to-goal* dan *blending*).Bagaimanapun, ketika tidak ada rintangan, adalah lebih baik dengan hanya menggunakan kontroler *go-to-goal* dan ketika robot sedang dalam bahaya mendekati rintangan, adalah lebih baik dengan hanya menggunakan kontroler *avoid-obstacles*. Logika peralihan bekerja lebih baik dalam situasi-situasi tersebut, namun kegugupan (*jitters*) antara kontroler *go-to-goal* dan *avoid-obstacle* terjadi ketika dekat dengan tujuan (*goal*). Solusinya adalah untuk merapatkan kontroler blending di antara kontroler *go-to-goal* dan *avoid-obstacles*.

IV. HASIL SIMULASI DAN ANALISIS

Analisis ini dilakukan dengan meletakkan sebuah target (berwarna biru) pada citra sederhana. Selama proses berjalannya robot tersebut maka sensor akan memicu berbagai perilaku (*behavior*). Selama robot mencari target maka arsitektur kendali *behavior based* memegang peranan penting. Saat menjumpai rintangan, perilaku penghindaran rintangan (*obstacle-avoidance*) diaktifkan sehingga koordinasi perilaku (*coordinasi competitive*) mengutamakan *behavior* ini dalam aksi pergerakan aktuator.Supervisor yang digunakan adalah *blending*, supervisor ini memiliki dua kontroler – *hold* dan campuran antara perilaku *go-to-goal* dan *avoid-obstacles*.

Tabel 4.1 Tabel respon perilaku terhadap situasi citra labirin

Perilaku	Situasi Citra	Respon
go-to-goal	Tidak ada rintangan	Berjalan menuju rintangan
Wall-following	Terdapat rintangan sebelah kanan dan kiri (labirin)	Berjalan tegak lurus terhadap tembok dengan jarak aman (d_{fw})
hold	Tempat tujuan	Berhenti (memberikan input nol pada kecepatan motor mobile robot)

Tabel 4.2 Pengujian kontroler proporsional pada citra labirin

No. Percobaan	Kp	Ki	Kd	Pengujian kontroler proporsional	Waktu Tempuh (s)
1	1	0	0	Respon robot terlihat sangat lambat dan robot menabrak rintangan	> 500
2	5	0	0	Respon robot sedikit meningkat, namun masih menabrak rintangan	>500
3	**10**	**0**	**0**	**Respon robot cukup baik, namun masih lama mencapai tujuan**	**>500**

Pada pengujian kontroler proporsional, ketika diberi Kp=1 mobile robot menabrak tembok ketika baru saja berjalan lurus. Kp = 10 merupakan nilai yang sesuai untuk mobile robot agar dapat bernavigasi di labirin dengan cukup aman. Meskipun begitu, mobile robot masih membutuhkan waktu yang relatif lama yaitu di atas 500 detik untuk mencapai tujuan.

Tabel 4.3. Pengujian kontroler integral pada citra labirin

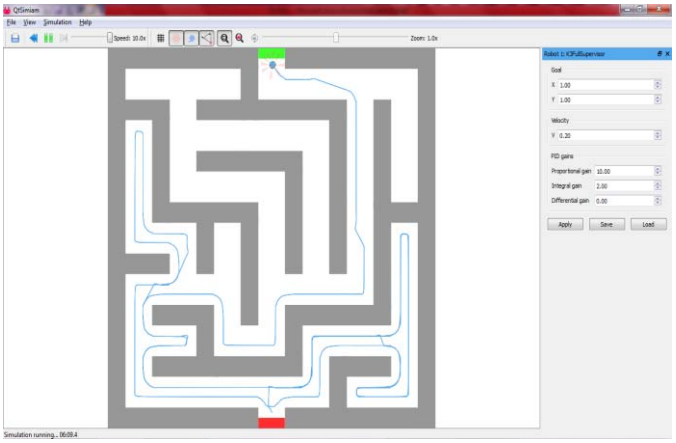
No. Percobaan	Kp	Ki	Kd	Pengujian kontroler integral	Waktu Tempuh (s)
1	10	1	0	Respon robot lambat, osilasi rendah	>500
2	10	3	0	Respon robot terlihat meningkat, osilasi berkurang dan stabil	>500
3	**10**	**5**	**0**	**Respon robot terlihat cepat dan stabil tanpa osilasi**	**<500**

Pada percobaan pertama dan kedua dengan nilai Kp=1 dan Kp=3, robot memberikan respon yang meningkat pada navigasinya di labirin. Namun waktu yang ditempuh oleh mobile robot masih di atas 500 detik.Dengan menyesuaikan kontroler integral menjadi 10, mobile robot dapat mencapai tujuan di bawah waktu rata-rata 500 detik.

Tabel 4.4. Pengujian kontroler diferensial pada citra labirin

No. Percobaan	Kp	Ki	Kd	Pengujian kontroler diferensial	Waktu Tempuh (s)
1	12	2	5	Osilasi tinggi	>300
2	12	2	15	Osilasi tinggi	>300
3	**12**	**2**	**0**	**Tidak berosilasi**	**360**

Pada percobaan di citra labirin, kontroler diferensial lagi-lagi memberikan respon yang sangat drastis berapapun nilainya. Mobile robot mengalami osilasi yang tinggi dan tidak berjalan (bergetar di tempat), seperti pada percobaan di citra sederhana. Sehingga kontroler diferensial diberi input nol. Jadi, pada percobaan labirin agar mobile robot dapat bernavigasi dengan aman, halus, dan stabil. Nilai kontroler yang digunakan yaitu: Kp = 12, Ki = 2, dan Kd = 0.



Gambar 4.1 Hasil simulasi pada citra labirin (Kp =10, Ki=2, dan Kd=0)

Sinusoida Vol. XX No. 1, Januari 2018
ISSN 1411 - 4593

V. SIMPULAN

Setelah dilakukan percobaan untuk simulasi mobile robot berbasis perilaku denganmenggunakan sistem kendali PID sebagaimana yang telah ditunjukkan pada bab pembahasan sebelumnya, diketahui bahwa :

1. Rancangan perilaku dimulai dari *blending* ketika ditemui dua kondisi yaitu *avoid-obstacles* dan *wall-following* (labirin) dan beralih ke *switching* di saat satu kondisi saja yang ditemui *go-to-goal* atau *avoid-obstacles*. Sehingga performansi navigasi dicapai dengan maksimum.
2. Penerapan kontroler PID pada mobile robottelah mampu membuat pergerakan mobile robotmenjadi stabil dan mampu membuat mobile robotbermanuver dengan aman, halus, responsif dan cepat, parameter kontroler PID tersebut diperoleh dari hasil *tuning* eksperimen dengan Kp=17, Ki=5 dan Kd=0 pada citra sederhana dan Kp=10, Ki= 2, dan Kd=0 pada citra labirin.

DAFTAR PUSTAKA

[1] Siegwart, R., Nourbakhsh, I. (2004). *Introduction to Autonomous Mobile Robots*.MIT Press.
[2] Spiegel, Murray. (1959). *Vector Analysis and an Introduction to Tensor Analysis*.Schaum's Outline.
[3] Ogata, K. (1980). *Teknik Kontrol Automatik*. Erlangga.
[4] Wahde, M. (2008). *Biologically Inspired Optimization Methods.*MIT Press
[5] Brooks, R. (1986). ―*A robust layered control system for a mobile robot*‖, IEEE Journal of Robotics and Automation Vol. 2, No. 1, hal.14–23.
[6] Dudek, G., Jenkin, M. *Computational Principles of Mobile Robotics*. Cambridge University Press.
[7] Egerstedt, Magnus. (2013). *Control of Mobile Robots.*Georgia Tech.
[8] Fahmizal., Effendi, R., Iskandar, E. (2013). *Implementasi Sistem Navigasi Behaviour-based dan Kontroler PID pada Manuver Robot Maze*.
[9] Kuhlman, D. 2013. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. MIT Press
[10] K Team. *Khepera III User Manual ver 3.5*.
[11] Malu, Sandeep Kumar dan Majumdar, Jharna. 2014. *Kinematics, Localization and Control of Differential Drive Mobile Robot*. Global Journal of Researches in Engineering : Robotics & Nano-Tech, Vol 14 Issue 1 Ver 1.0.