

RANCANG BANGUN 3 WHEELS OMNI-DIRECTIONAL MOBILE ROBOT MENGGUNAKAN SENSOR POSITION SENSITIVE DEVICE (PSD) SERTA SENSOR VISION DENGAN METODE KENDALI FUZZY LOGIC CONTROLLER (FLC) UNTUK MENGHINDARI HALANGAN

Wahyu Setyo Pambudi

Jurusan Teknik Elektro, Universitas Internasional Batam, Batam 29442

Jl. Gajah Mada, Baloi Sei Ladi, Batam

Handphone: 0812-3362210, Phone: +62-778-600 2999, 7437 111, Fax: +62-778-600 4219, 7437 112

Email : pambudi@uib.ac.id ; yoe2fa@yahoo.com

ABSTRAK

Penelitian tentang mobile robot telah banyak dilakukan dan dikembangkan, salah satunya tentang obstacle avoidance robot. Penelitian tentang obstacle avoidance telah menerapkan kendali adaptif, diantaranya menggunakan fuzzy logic controller (FLC). Obyek penelitian tentang mobile robot ada berbagai macam salah satunya adalah dengan menggunakan prinsip gerak 3 Wheels Omni-directional. Jenis penggerak ini memiliki keunikan, dimana dapat bergerak ke segala arah tanpa harus melakukan manuver terlebih dahulu. Agar dapat bergerak menghindari halangan maka diperlukan sensor untuk dapat mengenali lingkungan tempat robot ini berjalan, sehingga dapat terhindar dari halangan. Penggunaan sensor optical position sensitive device (PSD) sebagai sensing jarak dan sensor vision terbukti dapat dikolaborasi menggunakan behavior base sebagai input proses navigasi robot. Pengolahan sensor vision memanfaatkan model rgb atau normalisasi nilai-nilai red, green dan blue. Metode ini dapat mengurangi pengaruh intensitas cahaya pada saat mendeteksi adanya obyek atau obstacle. Metode kendali robot pada penelitian ini menggunakan PID sebagai internal kontrol untuk masing-masing motor DC dan FLC sebagai eksternal kontrol. Parameter yang sesuai untuk kendali PID pada penelitian ini adalah $K_p=0.2$, $K_i=0.001$ dan $K_d=0.2$ dengan error rata-rata 3,2%, dimana overshoot respon motor sangat kecil. Keluaran hasil defuzzifikasi FLC digunakan sebagai input nilai set point PID sebagai sebuah perpaduan sistem navigasi mobile robot.

Kata Kunci : 3 Wheels Omni-directional, PSD, Vision, FLC

1. PENDAHULUAN

Penelitian dan pengembangan tentang robot pertama kali hanya dikhususkan menangani masalah di industri dan terbatas untuk masalah peningkatan produksi sehingga muncul istilah industrial robot. Definisi robot industri pada saat itu hanya terbatas pada robot tangan (*arm robot*) dengan lengan-lengan kaku yang terhubung secara seri serta memiliki pergerakan memutar (rotasi) memanjang/memendek (translasi/prismatic).

Dewasa ini teknologi robot industri mulai tertinggal perkembangannya dengan teknologi *mobile robot*. Teknologi *mobile robot* digunakan secara meluas sejak tahun 80-an. Aplikasinya diarahkan untuk membantu tugas manusia di berbagai bidang dan bukan hanya pada industri [1]. Contoh *mobile robot* yang digunakan selain di industri adalah seperti robot Sojourner digunakan untuk misi ke Mars, robot Pioneer digunakan untuk misi ke Chernobyl, Tour guide robot yang merupakan jenis robot *Obstacle Avoidance* (Penghindar Halangan) dan masih banyak lagi [2].

Penelitian tentang robot *Obstacle Avoidance* ini banyak difokuskan pada sistem kendali adaptifnya. Sistem ini digunakan untuk mengendalikan robot agar dapat menghindari benturan dengan halangan yang terdapat pada lingkungan tempat robot berjalan. Agar robot mempunyai kemampuan untuk memberikan reaksi terhadap kondisi lingkungan yang belum diketahui maka sistem kendali yang digunakan harus mampu menyelesaikan masalah-masalah tersebut. Salah satu metode itu adalah dengan *Fuzzy Logic Controller*

(FLC), dimana dengan metode ini permasalahan ketidakpastian yang diakibatkan oleh lingkungan sekitar dapat diatasi. Kelebihan FLC ini yang kemudian dimanfaatkan sebagai sistem kendali pada *mobile robot* sedangkan untuk mengenali lingkungan sekitarnya, robot ini dilengkapi dengan sensor jarak [3].

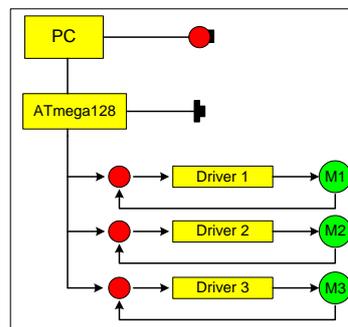
Position Sensitive Device merupakan sensor jarak yang aplikasinya digunakan *mobile robot* untuk dapat mengenali lingkungannya, dimana jenis sensor PSD diantaranya adalah sonar, laser dan infrared. Sebagian besar *mobile robot* yang menggunakan *microcontroller* sebagai unit pemrosesnya mengenali lingkungannya menggunakan sensor PSD ini [4]. Penelitian ini menggunakan penggerak differensial motor (2 roda penggerak dan 1 roda bebas). Penggunaan differensial motor dengan 1 roda bebas memiliki keterbatasan dalam menghindari halangan. Metode steering ini memerlukan lintasan yang lebar untuk melakukan manuver dan menghindari halangan, berbeda apabila yang digunakan *omni-directional wheels* yang hanya memerlukan lintasan kecil [5].

Selain menggunakan PSD untuk mengenali lingkungan *mobile robot* dapat juga menggunakan sensor vision salah satunya berbentuk kamera digital. Tingkat ketelitiannya sensor ini berdasarkan resolusi gambar yang dihasilkan. Pengolahan pada sensor vision mempunyai kompleksitas tinggi karena memerlukan kecepatan *processor* dan kapasitas memory ekstra dari *embedded system*. Kecepatan *processor* yang tinggi serta *memory* yang besar untuk saat ini hanya terdapat pada *personal computer* (PC), sehingga sensor vision akan lebih cocok apabila menggunakan PC sebagai unit pemrosesnya [6].

Berdasarkan hal tersebut maka dalam penelitian ini akan dikembangkan sebuah *mobile robot* yang dapat bernavigasi secara mandiri. Agar dapat mengenali lingkungannya digunakan beberapa sensor PSD yang saling mendukung serta sensor vision sebagai pendeteksi obyek, unit pemroses yang digunakan merupakan kombinasi antara *microcontroller* dengan PC. Metode pendukung keputusan yang digunakan untuk mengatasi kondisi lingkungan dinamis pada *mobile robot* adalah FLC. Mekanisme *steering* robot menggunakan 3 *wheels omni-directional*, dengan harapan bahwa *mobile robot* ini mampu menghindari halangan secara *smooth* (halus) dan fleksibel serta terbebas dari benturan.

2. PERANCANGAN SISTEM.

Perancangan sistem dapat digambarkan dalam bentuk blok diagram seperti yang diperlihatkan pada gambar 1. Sistem ini terdiri atas 4 bagian utama yaitu sensor vision, PC embedded/Laptop, mikrokontroler ATmega128 dan mikrokontroler ATmega8 yang dilengkapi dengan Diver Motor. Sensor vision yang ditempatkan pada PC embedded sebagai pengolahan citra. Sensor ini akan memberikan informasi kondisi lingkungan yang ada didepan robot dan mendeteksi obyek. Hasil bacaan sensor akan dikirim ke mikrokontroler ATmega128 untuk diproses. Motor DC yang di-coupling dengan roda akan bergerak sesuai informasi dari sensor PSD dan sensor vision. Steering dari robot akan dikontrol dengan mengubah-ubah arah putar ketiga motor DC tersebut. Kecepatan putaran diatur dengan memberikan sinyal PWM (pulse width modulation). Pengontrolan kecepatan dan arah tergantung kepada besaran yang dibaca sensor-sensor.



Gambar 1: Diagram blok 3 *wheels omni-directional mobile robot*

2.1 OMNI-DIRECTIONAL WHEELS

Roda Omni-directional adalah suatu roda unik karena memiliki kemampuan bergerak bebas dua arah. Roda ini berputar seperti roda pada umumnya serta mampu bergeser kesamping menggunakan roda di sepanjang

lingkar luar roda. Roda Omni-directional memungkinkan robot untuk mengkonversi dari robot non-holonomic untuk robot holonomic. Sebuah robot non-holonomic yang menggunakan roda normal hanya memiliki 2 DOF (Degree of Freedom) yang terkendali, yaitu bergerak maju / mundur dan rotasi. Robot non-holonomic tidak memiliki kemampuan untuk bergerak kesamping kiri / kanan sehingga membuat robot lebih lambat dan kurang efisien dalam mencapai tujuan yang diberikan.

Roda omni-directional holonomic mampu mengatasi masalah ini karena roda memiliki 3 DOF. Berbeda dengan robot non-holonomic normal, robot omni-directional holonomic mampu bergerak ke segala arah tanpa mengubah arah roda. Roda omni-directional holonomic dapat bergerak maju mundur, geser ke samping, dan berputar pada posisi tetap. Kemampuan ini memungkinkan robot yang menggunakan omni-directional mampu bermanuver untuk lebih lincah dan lebih efisien [5][7].



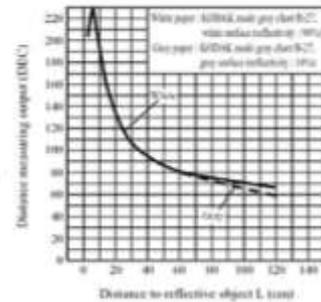
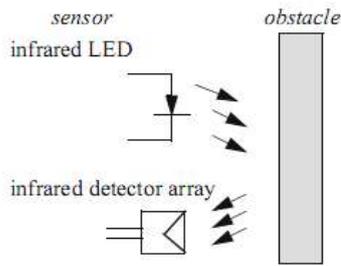
Gambar 2: Bentuk fisik roda omni-directional [8] (a), Diagram kinematik dasar 3 roda [9] (b)

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta) & \cos(\theta) & R \\ -\sin(\theta + \alpha_2) & \cos(\theta + \alpha_2) & R \\ -\sin(\theta + \alpha_3) & \cos(\theta + \alpha_3) & R \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & 0 \\ 0 & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\theta} \end{bmatrix} \quad (1)$$

2.2 Sensor

Selain menggunakan sonar sensor atau *ultrasonic* untuk mengukur jarak obyek pada mobile robot, sensor lain yang dapat digunakan adalah sensor infrared maupun laser sensor. Bentuk implementasi sensor infrared untuk mengukur jarak obyek, salah satunya digunakan pada robot untuk exteroceptor system (sistem untuk mengetahui respon dari luar) dimana sensor ini menggunakan sembilan GP2D12 infrared (IR) sebagai sensor jarak dan empat GP2D15 sebagai contact sensor. Menggunakan berbagai sensor IR untuk mendukung sistem exteroceptor pada mobile robot dan bukan sensor sonar, karena untuk menghindari beberapa kelemahan. Kelemahan tersebut adalah gangguan akibat adanya banyak pantulan yang ditimbulkan oleh pemancar sensor, kecepatan yang terbatas pada saat pengukuran dan jika permukaan target tidak rata maka frekwensi ultrasonic yang dipantulkan banyak berkurang [4].

Sensor IR GP2D12 dan GP2D15 produk dari SHARP ini menggunakan prinsip pengiriman pulsa infrared dengan frekwensi 40Khz dengan panjang gelombang 880nm, pulsa infrared dipantulkan oleh obstacle dan kemudian diterima detection array. Jarak yang diukur merupakan representasi dari sudut yang dibentuk oleh infrared LED, obstacle/obyek dan infrared detector array[6][10].



(a)

(b)

Gambar 3: Infrared range sensor [6] Gambar 4: Sharp sensor PSD[6] (a), diagram respon sensor[6] (b)

Berdasarkan gambar diagram respon dari sensor, dapat diketahui bahwa sensor ini memiliki kelemahan tidak dapat melakukan pengukuran apabila jarak obstacle dibawah 6 cm. Hal ini disebabkan karena dibawah dan diatas 6 cm nilai DEC cenderung turun, sehingga sulit untuk menentukan bahwa jarak tersebut lebih besar atau lebih kecil dari pada 6 cm. Solusi untuk mengatasi permasalahan ini adalah dengan menempatkan sensor lebih menjorok masuk ke badan robot, sehingga obstacle tidak pernah mendekati nilai 6 cm, solusi selanjutnya dengan menambahkan sensor infrared atau tactile switch sebagai proximity sensor sehingga jarak minimum obstacle dapat diketahui.

2.3 Sensor vision dan pengolahan citra.

Bentuk dari sensor vision adalah kamera digital dan webcam. Sensor vision merupakan sensor yang paling kompleks digunakan dalam robotika. Sensor ini tidak banyak digunakan pada embedded sistem sampai saat ini, karena keterbatasan kecepatan prosesor dan kapasitas memori yang diperlukan untuk pengolahannya.



Gambar 5: Webcam sebagai Sensor vision [11]

Tahapan pengolahan sensor vision ini dinamakan pengolahan citra digital. Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra [12]. Citra yang ditangkap oleh kamera yang dikuantisasi dalam bentuk diskrit disebut sebagai citra digital (*digital image*), dimana citra digital ini memiliki 2 jenis yaitu , citra diam (*still image*) dan citra bergerak (*moving image*) [13]. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (pixel = picture element) atau elemen terkecil dari sebuah citra.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \dots \dots \dots (2)$$

Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

- $0 \leq x \leq M-1$
- $0 \leq y \leq N-1$
- $0 \leq f(x,y) \leq G-1$

dimana : M = jumlah piksel baris (row) pada array citra
 N = jumlah piksel kolom (column) pada array citra
 G = nilai skala keabuan (graylevel)

Salah satu model pengolahan citra adalah melalui proses pengolahan warna pada citra digital. Pada pengolahan warna citra, ada bermacam-macam model warna. Model RGB (*Red Green Blue*) merupakan model yang banyak digunakan, salah satunya adalah monitor. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain model ini terdapat juga model *normalized* RGB atau *rgb space* dimana model ini terdapat 3 komponen yaitu, r, g dan b. Nilai *rgb* ini merupakan prosentase nilai RGB pada citra ($r+g+b=1$)[15]. Bentuk dari aplikasi model *normalized* RGB ini adalah untuk mendeteksi warna kulit maupun *traffic sign* [16][17][18].

2.4 Sistem kendali mobile robot

2.4.1 Proportional Integral Derivative (PID) Control.

Kontroler proporsional-integral-derivatif (PID controller) adalah model *control close loop* yang banyak digunakan secara luas pada industrial sistem control [1]. Model kontrol ini akan menghitung *error* sebagai nilai perbedaan antara hasil proses terukur dengan nilai *setpoint* yang diinginkan. Kontroler ini akan mencoba untuk meminimalkan kesalahan dengan menyesuaikan input proses kontrol.

Sebuah kontroler PID dapat dianggap sebagai bentuk pengendali dengan kompensator fasa lead-lag dari nilai *setpoint* titik awal sampai tak terhingga. Demikian pula dengan kontroler sejenis, yaitu PI dan PD dapat juga dianggap sebagai bentuk pengendali dengan kompensator dari fasa lag dan fasa lead secara berurutan [19]. Sebuah kontroler PID standar juga dikenal sebagai pengendali *three-term* atau tiga kondisi, yang fungsi transfer umumnya ditulis dalam bentuk paralel seperti persamaan 3.

$$G(s) = K_p + \frac{K_i}{s} + K_d s \dots\dots\dots (3)$$

dimana nilai K_p untuk *proportional gain*, K_i untuk *integral gain* dan K_d untuk *derivative gain*.

Pengendali tiga kondisi tersebut mempunyai fungsi masing-masing, diantaranya adalah :

- a) *Proportional term* - memberikan aksi kontrol secara proporsional terhadap respon sinyal error melalui *all-pass gain factor*.
- b) *Integral term* - mengurangi *error steady-state* melalui kompensasi frekwensi rendah dengan sebuah *integrator*.
- c) *Derivative term* - meningkatkan respon transien melalui kompensasi frekwensi tinggi dengan sebuah *differentiator*.

Efek dari perubahan nilai untuk masing-masing kondisi tersebut, terdapat pada tabel 2 berikut ini [19] :

Tabel 2: Efek kontrol terhadap perubahan tiap-tiap kondisi

Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_p	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_i	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Adanya perkembangan computer dan embedded system, maka metode kendali PID ini dapat dijalankan secara digital atau dengan menggunakan program. Agar dapat mengimplementasikan sistem kendali PID secara digital, maka persamaan PID harus diubah ke dalam persamaan diskrit [20].

$$V_o = K_p + K_i \int edt + K_d \frac{de}{dt} \dots\dots\dots (4)$$

diturunkan menjadi :

$$\frac{dV_o}{dt} = K_p \frac{de}{dt} + K_i \frac{d}{dt} \left(\int edt \right) + K_d \frac{d^2e}{dt^2} \dots\dots\dots (5)$$

$$\frac{dV_o}{dt} = K_p \frac{de}{dt} + K_i e + K_d \frac{d}{dt} \left(\frac{de}{dt} \right) \dots\dots\dots (6)$$

kemudian persamaan 7 dirubah kedalam bentuk diskrit,

$$\frac{\Delta V_o}{T_s} = K_p \frac{\Delta e}{T_s} + K_i e + K_d \frac{\Delta}{T_s} \left(\frac{\Delta e}{T_s} \right) \quad (7)$$

dimana $d = \Delta$ dan $dt = T_s$, kemudian dikalikan dengan T_s , maka :

$$\Delta V_o = K_p \Delta e + K_i e T_s + K_d \Delta \left(\frac{\Delta e}{T_s} \right) \quad (8)$$

Harga $\Delta V_o = V_{on} - V_{on-1}$ dan $\Delta e_n = e_n - e_{n-1}$, sehingga menjadi

$$V_{on} - V_{on-1} = K_p (e_n - e_{n-1}) + K_i e_n T_s + \frac{K_d}{T} (\Delta e_n - \Delta e_{n-1}) \quad (9)$$

Pada kondisi akhir, perubahan Δ pada *error* sebelumnya dapat didistribusikan menjadi $\Delta e_n = e_n - e_{n-1}$ dan $\Delta e_{n-1} = e_{n-1} - e_{n-2}$, kemudian disubstitusikan ke dalam persamaan 9, maka persamaan tersebut menjadi :

$$V_{on} = V_{on-1} + K_p (e_n - e_{n-1}) + K_i e_n T_s + \frac{K_d}{T} (e_n - 2e_{n-1} + e_{n-2}) \quad (10)$$

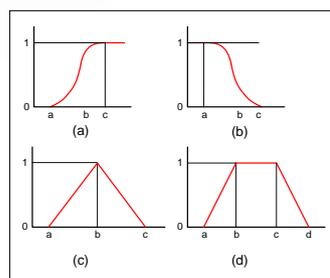
Dimana persamaan 10 tersebut merupakan hasil akhir persamaan system kendali PID untuk diskrit.

2.4.1. Fuzzy Logic Controller (FLC).

Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfy A. Zadeh dari Universitas California di Berkeley (1965). Logika Fuzzy ini berbeda dengan logika Boolean dimana keadaan hanya menjadi dua kondisi yaitu 0 dan 1 sedangkan logika fuzzy memberikan kemungkinan suatu keadaan dalam interval antara 0 dan 1 [21]. Sistem kendali *fuzzy* adalah metodologi disain sistem kendali yang berbasiskan pada *fuzzy logic*. Definisi *fuzzy logic* berarti operasi logika yang menggunakan aturan *if-then rules*, dimana kendali fuzzy mengkombinasikan aturan *if-then* tersebut untuk menghitung input sistem kontrol.

Dalam sistem *fuzzy logic controller*, *crisp input* akan dirubah ke dalam *degree of membership* (derajat keanggotaan) fuzzy pada unit fuzzifikasi. Derajat keanggotaan ini disimpan dan dikombinasikan sesuai dengan aturan *rulebase* (basis aturan) pada *rule evaluation* untuk menghasilkan pemotongan konsekuen fuzzy set (*aggregator*). Defuzzifikasi adalah proses untuk merubah *inference fuzzy control action* ke dalam *crisp output* untuk *control action* [21].

Langkah pertama untuk mendesain sistem *fuzzy logic controller* adalah mengambil nilai *crisp input*, dan menentukan fungsi derajat keanggotaan (*degree of membership function*) untuk masing-masing fuzzy set yang sesuai. Fungsi derajat keanggotaan fuzzy memainkan peranan yang sangat penting untuk merepresentasikan masalah dan menghasilkan keputusan yang akurat.



Gambar 8: Fungsi Keanggotaan Fuzzy, (a) tipe Z, (b) tipe S, (c) tipe π segitiga (*Triangle*), (d) tipe π trapesium (*Trapeziod*).

a. Tipe S.

Seperti namanya, fungsi ini berbentuk kurva sigmoidal seperti huruf S. Setiap nilai x (anggota *crisp set*) dipetakan ke dalam interval $[0,1]$. Grafik dan notasi matematika untuk fungsi ini adalah seperti berikut :

$$Triangle(x, a, b, c) = \begin{cases} 0, & x < a \ \& \ x > c \\ \frac{x-a}{b-a}, & a \leq x < b \\ \frac{c-x}{c-b}, & b \leq x < c \end{cases} \dots\dots\dots (11)$$

b. Tipe Z

Tipe Z ini merupakan kebalikan dari fungsi S.

$$Z(x, a, b, c) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-a}{c-a}\right)^2, & b < x \leq a \dots\dots\dots (12) \\ 2\left(\frac{x-a}{c-a}\right)^2, & c < x \leq b \\ 0, & c < x \end{cases}$$

c. Tipe π Segitiga.

$$Triangle(x, a, b, c) = \begin{cases} 0, & x < a \ \& \ x > c \\ \frac{x-a}{b-a}, & a \leq x < b \\ \frac{c-x}{c-b}, & b \leq x < c \end{cases} \dots\dots\dots (13)$$

e. Tipe π Trapesium.

$$Trapeziod(x, a, b, c) = \begin{cases} 0, & x \leq a \ \& \ x > d \\ \frac{b-x}{b-a}, & b \leq x \leq a \dots\dots\dots (14) \\ 1, & b < x \leq c \\ \frac{x-d}{c-d}, & c < x \leq d \end{cases}$$

Crips input merupakan nilai numerik yang terbatas pada semesta pembicaraan (*universe of discourse*). Rentang dari semesta pembicaraan dapat ditentukan berdasarkan persepsi dari seorang ahli permasalahan yang akan diselesaikan. Salah satu contoh, misalnya jika ingin diketahui keterlibatan staff dalam sebuah proyek maka diperlukan seorang ahli untuk menentukan nilai antara 0 sampai 100 % yang mewakili gaji untuk masing-masing staff proyek, dimana hal ini harus benar-benar memadai. Guna menyelesaikan masalah tersebut menggunakan *fuzzy* sistem, akan berdasarkan pada berbagai *crisp input* yang berbeda. Sementara beberapa input dapat diukur secara langsung, misalnya : tinggi, berat, kecepatan, jarak, suhu, tekanan dll), beberapa dari input tersebut dapat hanya didasarkan pada perkiraan ahli.

Tahapan *rule evaluation* adalah tahapan untuk menggunakan aturan (*rule*) untuk menentukan aksi apa yang harus dilakukan. Sebuah basis aturan adalah sekumpulan rule *if-then fuzzy* yang secara khusus menghubungkan antara variabel linguistik dan output dari sistem kontrol. Secara umum rule if-then mempunyai bentuk :

IF A0 is a0 AND A0 is a1...and An is an THEN C0 is c1 AND Cm is cm

dimana A0 *antecedent*, yaitu masukan yang telah defuzzifikasi, a0 adalah *antecedent* linguistik label, Cm adalah *consequents* yaitu aksi pengendalian (keluaran) dan cm adalah *consequent* linguistik label. Linguistik label adalah nama lain dari fungsi keanggotaan. Hubungan antara *antecedent* dan *consequent* disebut aturan (rule), dan antara satu rule dengan yang lain tidak terdapat hubungan sebab akibat.

Apabila FLC ini diaplikasikan pada robot dengan menggunakan *defferential steering*, dimana 2 buah motor digunakan sebagai penggerak robot, yang memiliki array sensor untuk mengukur jarak rintangan di sekitarnya. Informasi yang diperoleh dari array sensor ini akan diolah dengan menggunakan aturan *fuzzy*. Keluaran dari aturan akan digabungkan dengan operasi FLC untuk mengontrol kecepatan roda penggerak robot tersebut, dimana roda penggerak ini dilambangkan sebagai *left_v* dan *right_v*, yang akan digunakan untuk menghasilkan aturan *fuzzy* [22].

Aturan Fuzzy : If (left-obs is near and right-obs is medium and front-obs is far and head-ang is N) Then (left-v is slow and right-v is medium).

dimana *left-obs* = jarak halangan sebelah kiri, *right-obs* = jarak halangan sebelah kanan, *front-obs* = jarak halangan dari depan, *head-ang* = sudut robot, *left-v* = kecepatan roda penggerak kiri dan *right-v* = kecepatan roda penggerak kanan. Sedangkan untuk aplikasi pada 3 *wheels omni-directional* ini maka aturannya adalah sebagai berikut :

Aturan Fuzzy 3 wheels omni-directional mobile robot :
If (Jarak1 is VS1 and Jarak2 is VS2 and Jarak2 is VS2) Then (Motor1 is NS1) (Motor2 is NS2) (Motor3 is NS3).

Dengan menggunakan *fuzzy reasoning* dan metode *centroid defuzzification*, aturan *fuzzy* yang terkait dengan *obstacle avoidance* dan diukur bobotnya untuk menentukan aksi kontrol yang sesuai (kecepatan Motor1, Motor2 dan Motor3) pada roda penggerak robot.

Proses defuzzifikasi adalah proses pemetaan dari bagian pengambilan keputusan aksi kendali fuzzy ke bagian aksi kontrol non fuzzy (crisp). Dalam defuzzifikasi, nilai crisp dihitung dari konsequent monolitik. Output defuzzifikasi secara umum dapat dihitung dengan persamaan 14 untuk fungsi kontinyu dan persamaan 15 untuk fungsi diskrit:

$$z = \frac{\int_{j=1}^j z_j \cdot \mu(z_j)}{\int_{j=1}^j \mu(z_j)} \dots\dots\dots (15)$$

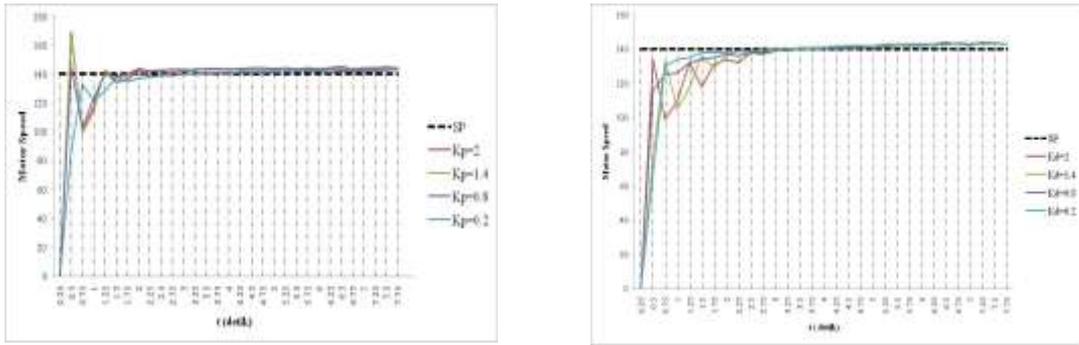
$$z = \frac{\sum_{j=1}^j z_j \cdot \mu(z_j)}{\sum_{j=1}^j \mu(z_j)} \dots\dots\dots (16)$$

Salah satu keuntungan dari defuzzifikasi centroid adalah metode ini menggambarkan kesimpulan menggunakan lebih banyak informasi yang tersedia dari fungsi keanggotaan monolitik. Ini dapat disimpulkan bahwa lebih banyak rule dari basis aturan diambil sebagai pertimbangan ketika memutuskan output akhir. Walaupun metode centroid mencakupi perhitungan sebuah integrasi dimana perhitungannya lebih rumit.

3. Hasil dan pembahasan.

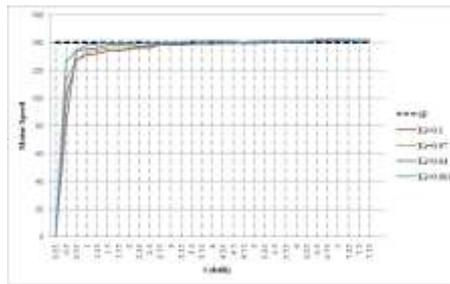
3.1 Pengujian kendali PID.

Hasil respon kendali PID ini salah satunya ditentukan oleh parameter Kp, Ki dan Kd. Metode yang digunakan untuk pengujian ini adalah dengan mencoba beberapa sampel nilai, dimana hasilnya akan disimpan pada komputer. Proses kontrol ini dilakukan oleh mikrokontroler AVR, dimana input didapat dari sensor *rotary encoder*, maka hasilnya adalah :



(a)

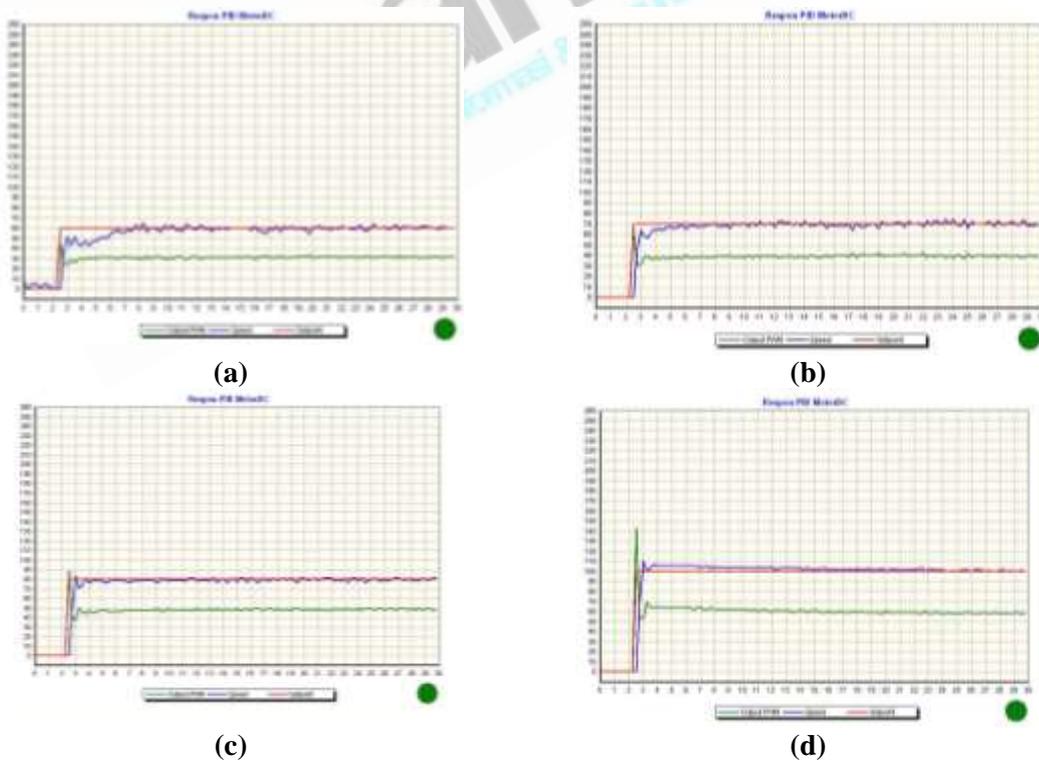
(b)



(c)

Gambar 9: Output kendali PID pada saat SP=140, $K_i=0.1$, $K_d=2$ (a), SP=140, $K_p=0.2$, $K_i=0.1$ (b), SP=140, $K_p=0.2$, $K_d=0.2$, $K_i=0.001$ (c).

Berdasarkan nilai parameter-parameter PID ini maka perlu dilakukan analisa menggunakan kecepatan yang berbeda-beda, hasilnya adalah :



(c)

(d)

Gambar 10: Hasil respon kendali PID untuk kecepatan motor DC

(a) Setpoint = 60, (b) Setpoint = 70, (c) Setpoint = 80, (d) Setpoint = 100

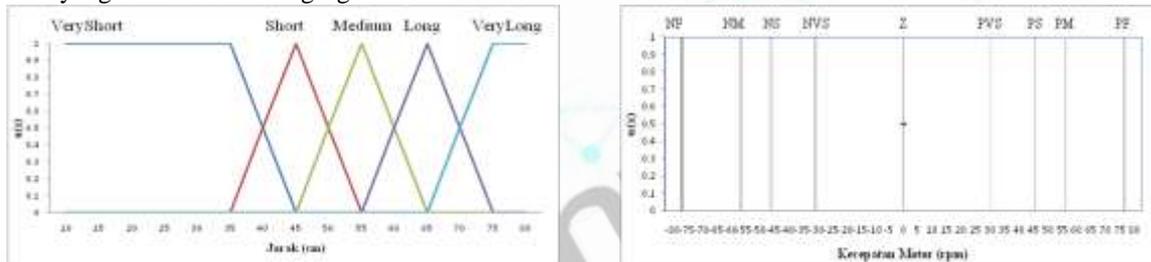
Tabel 4.1. Rata-rata error kecepatan motor DC terhadap *setpoint*

No	Kecepatan Motor (rpm)	Rata-rata error (%)
1	60	4%
2	70	2%
3	80	2%
4	90	4%
5	100	4%
Total rata-rata error		3.2%

3.2 Pengujian kendali FLC.

Dalam kendali FLC ini yang perlu didesain adalah *fuzzyfication*, *rule evaluation* dan *defuzzyfication*. Input yang digunakan untuk FLC berasal dari 3 buah sensor jarak sedangkan outputnya akan digunakan sebagai nilai *setpoint* pada PID.

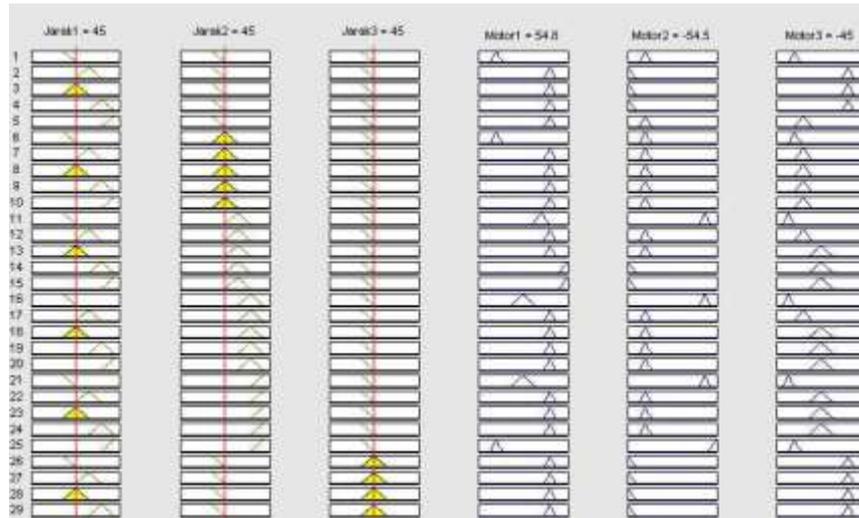
Pada desain *fuzzification* untuk penelitian ini, yang diperlukan adalah mengambil nilai *crisp input* berupa data jarak dari sensor GP2D12, kemudian menentukan derajat keanggotaannya untuk masing-masing *fuzzy-set*. Input berupa sensor jarak tersebut dibagi menjadi tiga bagian yaitu : *very short*, *short*, *medium*, *long* dan *very long*. *Fuzzy-set* untuk *very short* dan *very long* mempunyai bentuk setengah trapesium sedangkan untuk yang lain berbentuk segitiga.



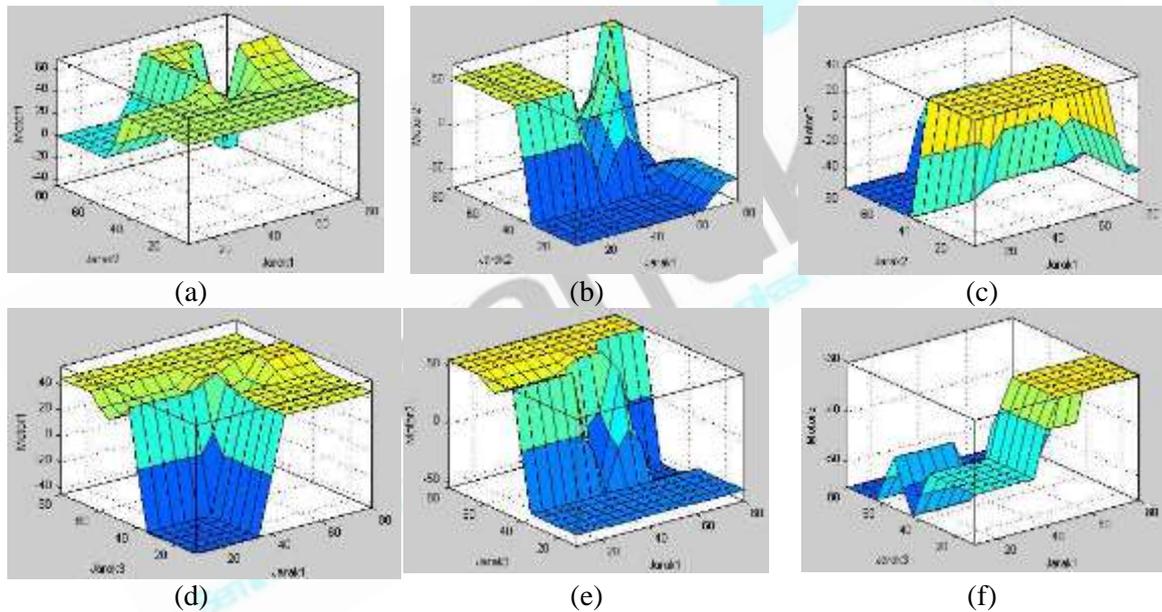
(a) (b)
Gambar 10: Fungsi Keanggotaan Fuzzy Input (a) output (b)

3.2 Pengujian kendali FLC.

Pada output FLC ini, *fuzzy-set* dibagi menjadi ada 5 bagian, diantaranya adalah *Negative Fast* (NF), *Negative Medium* (NM), *Negative Slow* (NS), *Negative Very Slow* (NVS), *Zero* (Z), *Positive Very Slow* (PVS), *Positive Slow* (PS), *Positive Medium* (PM) dan *Positive Fast* (PF). Setelah desain *membership function* untuk input-output ini maka *rule evaluation* bisa didesain sehingga hasilnya bisa di simulasikan dengan sebuah program, hasilnya adalah :



Gambar 10: Rule Simulator.

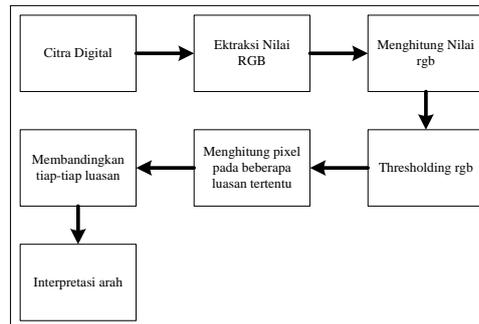


Gambar 11: *Input-Output surface* dari sistem kedali FLC: Jarak1 dan Jarak2 terhadap Motor1(a), terhadap Motor2(b) terhadap Motor3 (c), Jarak1 dan Jarak 3 terhadap Motor1(d) terhadap Motor2(e) terhadap Motor3(f).

3.3 Pengolahan citra digital untuk mengenali obyek.

Program yang digunakan untuk pengolahan citra digital pada penelitian ini menggunakan EmguCV dengan platform .NET pada Visual Studi 2008. Selain program tersebut, fasilitas yang digunakan adalah kamera webcam sebagai bentuk dari sensor vision, dimana struktur programnya seperti pada gambar 12. Stuktur program ini akan dijalankan selama 33 ms atau sekitar 30 fps (*frame per second*) sesuai standar NTSC[23]. Pada pengolahan citra digital untuk mendeteksi obyek, obyek yang memiliki warna putih dan hitam tidak

bisa dibedakan karena nilai r,g,b memiliki nilai prosentase hampir sama dan sulit dibedakan, oleh sebab itu pada penelitian ini kedua warna tersebut akan dianggap sebagai lantai ataupun dinding.



Gambar 12: Struktur program pengolahan citra untuk mengenali obyek



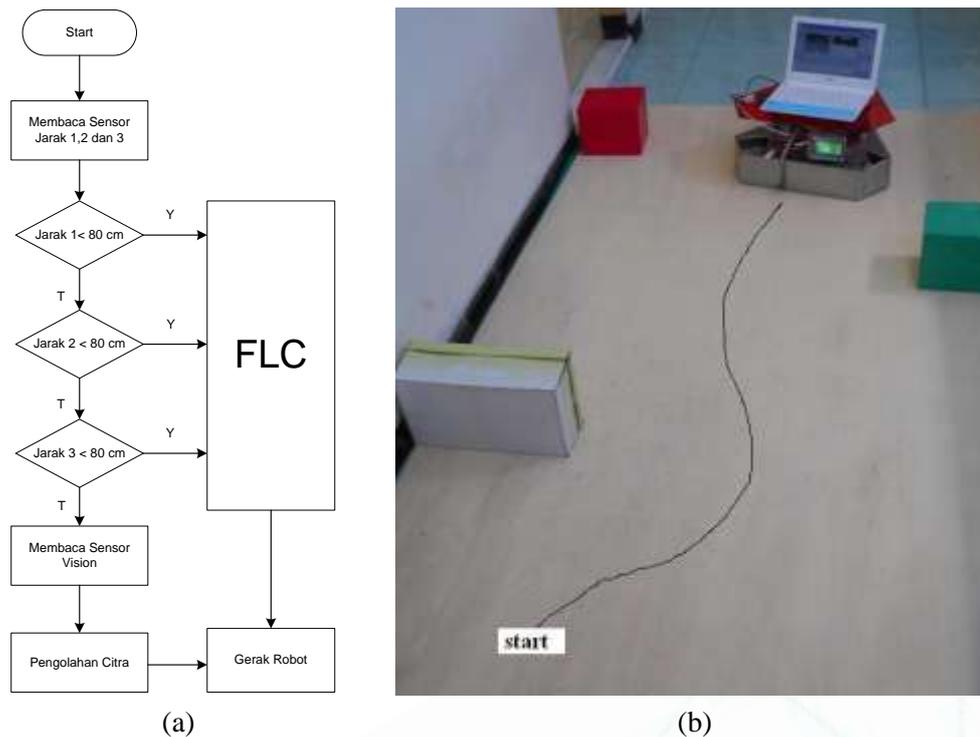
Gambar 13: Hasil pengolahan citra pada saat mendeteksi obyek sebelah kiri (a), obyek sebelah kanan (b).

Tabel 1: Hasil proses *normalized RGB*

Jenis Warna	r	g	b
Merah	0.6556903	0.1748044	0.1695055
Hijau	0.2268326	0.4299823	0.343185
Biru	0.2725676	0.2493136	0.4781188
Coklat	0.463363	0.2989612	0.2376759
Hitam	0.336211	0.3291586	0.3346305
Putih	0.3335534	0.333749	0.3326981

3.4 Pengujian *mobile robot*.

Sistem integrasi dari sensor PSD dengan vision sensor menggunakan model behavior base [1]. Sistem ini bekerja berdasarkan prioritas dan urutan yang dibutuhkan[24]. Agar diketahui kinerja *mobile robot* setelah integrasi sistem, perlu dilakukan uji coba untuk menghindari halangan. Lingkungan tempat robot berjalan diberi obyek sebagai halangan, dengan harapan robot dapat menghindari dari halangan tersebut. Lantai tempat berjalan robot diberi alas untuk menghindari roda plastic yang licin pada saat berjalan di lantai sesungguhnya.



Gambar 14: Struktur *behavior base mobile robot* (a), hasil pengujian *mobile robot* (b).

4. Penutup.

4.1 Kesimpulan.

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan, bahwa :

1. Obyek yang dapat dideteksi oleh *mobile robot* sesuai dengan jarak maksimal yang dapat diukur oleh sensor PSD yaitu sekitar 80 cm.
2. Penentuan nilai K_p , K_i dan K_d pada kendali PID akan sangat menentukan respon kontrol terhadap *setpoint*, dimana pada penelitian ini yang digunakan $K_p=0.2$, $K_i=0.02$ dan $K_d=0.2$ dengan error rata-rata 3,2%.
3. Desain *membership function* untuk input dan output mempunyai pengaruh terhadap pergerakan *mobile robot*.
4. Proses pengolahan citra pada *mobile robot* untuk mendeteksi obyek berdasarkan warna menggunakan metode *normalized rgb* masih memiliki kelemahan dimana warna putih dan hitam sulit dibedakan.

4.2 Rekomendasi.

Selama melakukan penelitian ini peneliti merekomendasikan

1. Berdasarkan penelitian yang telah dilakukan bahwa sensor vision nantinya dapat dikembangkan sebagai pengganti sensor PSD pada robot. Apabila tetap mempertahankan sensor PSD, maka gunakan sensor dengan kemampuan mengukur jarak lebih jauh.
2. Berdasarkan blok diagram system yang dibuat, memungkinkan untuk menerapkan beberapa metode kecerdasan buatan pada *mobile robot*, guna mengatasi dynamic obstacle.

DAFTAR PUSTAKA

- [1] Pitowarno, E., ROBOTIKA: “*Desain, Kontrol, dan Kecerdasan Buatan*”, Andi Yogyakarta, 2006, Bab???, hal ???
- [2] Siegwart, R., Nourbakhsh, Illah., R, “*Introduction to Autonomous Mobile Robots*”, A Bradford Book, The MIT Press Cambridge, 2004, Massachusetts.

- [3] A. Saffiotti, “*The Uses of Fuzzy Logic in Autonomous Robot Navigation*”, *Soft Computing* 1(4):180-197, 1997.
- [4] Rusu. P., Emil M. Petriu, Fellow, IEEE, Thom E. Whalen, Aurel Cornell, and Hans J. W. Spoelder, “*Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation*”, *IEEE Trans. on Instrumentation and Measurement* Vol. 52, No. 4, p. 1335 – 1340, 2003.
- [5] Takahashi. M, Developing a mobile robot for transport applications in the hospital domain, Elsevier. on *Robotics and Autonomous System*, doi:10.1016/j.robot.2010.03.010, 2010.
- [6] Braunl, T., *Embedded Robotics: “Mobile Robot Design and Application with Embedded Systems”*. Berlin: Springer-Verlag Berlin Heidelberg, Inc, 2008.
- [7] Rostami. V, Ebrahimijam. S, Khajehpoor. P, Mirzaei. P, Yousefiazar. M, *Cooperative Multi Agent Soccer Robot Team*, *World Academy of Science, Engineering and Technology* 9, 2005.
- [8] <http://www.acroname.com/robotics/parts/R130-8CM-POLY-ROLLER.html>, 2011.
- [9] Baede. T.A., *Motion Control of an Omnidirectional Mobile Robot*, Traineeship Report, National University of Singapore, Faculty of Engineering, Departement of Mechanical Engineering, Control and Mechatronics Group and Eindhoven University of Technology, Departement of Mechanical Engineering, Dynamics and Control Technology Group, 2006.
- [10] SHARP, Data Sheet GP2D12 - Compact, High Sensitive Distance Measuring Sensor, Sharp Co., data sheet, <http://www.sharp.co.jp/ecg/>, 2006.
- [11] <http://www.citrajaya.net/produk/genius/850-genius-webcam-islam-310.html>, 2011.
- [12] Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, “*Teori Pengolahan Citra Digital*”, Andi Yogyakarta dan UDINUS Semarang, 2009.
- [13] Purnomo Mauridhi Hery, Muntasa Arif, “*Konsep Pengolahan Citra Digital dan Ekstraksi Fitur*”, Graha Ilmu Yogyakarta, 2010.
- [14] Jähne Bernd, Haußecker Horst, “*Computer Vision and Applications*”, San Diego, California, Academic Press, 2000.
- [15] P. Kakumanu, S. Makrogiannis, N. Bourbakis, “*A survey of skin-color modeling and detection methods*”, *ScienceDirect on Pattern Recognition* 40 (2007) 1106 – 1122.
- [16] Soetedjo Aryuanto, Yamada Koichi, Limpraptono F. Yudi, “*Segmentation of Road Guidance Sign Symbols and Characters based on Normalized RGB Chromaticity Diagram*”, *International Journal of Computer Applications* (0975 – 8887) Volume 3 – No.3, June 2010
- [17] J. Yang, W. Lu, A. Waibel, “*Skin-color modeling and adaptation*”, *ACCV98*, 1998.
- [18] M.H. Yang, N. Ahuja, “*Gaussian Mixture model for human skin color and its application in image and video databases*”, *Proceedings of SPIE: Conference on Storage and Retrieval for Image and Video Databases*, vol. 3656, 1999, pp. 458–466.
- [19] Ang. K. H, Chong. G, Li Yun (2005), “*PID Control System Analysis, Design, and Technology*”, *IEEE Transactions On Control Systems Technology*, Vol. 13, No. 4, July 2005, 559.
- [20] Bachri Samsul M, (2004), “*Sistem Kendali Hybrid Pid - Logika Fuzzy Pada Pengaturan Kecepatan Motor DC*”, *Makara, Teknologi*, Vol. 8, No. 1, April 2004: 25-34.
- [21] Negvevitsky, M., “*Artificial Intelligence: A Guide to Intelligent System*”. Sydney : Addison Wesley. 2002.
- [22] Singh, M.K, D.R.Parhi, S.Bhowmik, S.K.Kashyap, “*Intelligent Controller for Mobile Robot: Fuzzy Logic Approach*”, *The 12th International Conference of International Association for Computer Methods and Advances in Geomechanics (IACMAG)*, Goa, India, 1-6 October, 2008.
- [23] http://www.paradiso-design.net/videostandards_en.html, 2011.
- [24] Skrzypczyk krzysztof, “*Behaviour Activity Based Maps Applied to Mobile Robot Navigation*”, *Proceedings of the 13th WSEAS International Conference on System*, ISSN : 1790-2769, ISBN : 978-960-474-097-0.