

IMPLEMENTASI METODE *HYBRID ARTIFICIAL NEURAL NETWORK* (ANN) – PID UNTUK PERBAIKAN PROSES BERJALAN PADA *PROTOTYPE ROBOT MATERIAL HANDLING*

Agus Salim¹⁾, Wahyu Setyo Pambudi²⁾,

Program Studi Teknik Elektro, Universitas Internasional Batam

Email : 18agus.salim@gmail.com¹⁾, pambudi@uib.ac.id²⁾

ABSTRAK

Penelitian pada *mobile robot* telah banyak dilakukan, salah satunya yaitu pada robot *material handling*. Berbagai metode sudah diterapkan untuk mengembangkan proses berjalan pada robot ini, diantaranya menggunakan metode PID. Salah satu kelemahan dari metode ini adalah proses penentuan parameter K_p , K_i dan K_d dengan metode *trial and error* sehingga nilai parameter yang dihasilkan belum tentu tepat. Metode *Artificial Neural Network* (ANN) merupakan suatu pendekatan model kecerdasan yang diilhami dari struktur otak manusia dan kemudian diimplementasikan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses *learning* berlangsung. Metode *hybrid ANN – PID* adalah satu rekayasa metode penggabungan antara ANN dan PID yang diharapkan dari metode ini penentuan parameter K_p , K_i dan K_d lebih realistis dan tidak lagi secara manual. Pada penelitian ini digunakan *prototype robot material handling* dan sebuah lintasan yang berbentuk oval yang akan dilalui oleh robot *material handling* tersebut. Sebelumnya dengan metode *trial and error* ditentukan nilai K_p , K_i dan K_d sebagai nilai awal, selanjutnya robot akan berjalan mengikuti jalur, diawal cara berjalan robot kurang sesuai jalur namun seiring pembelajaran yang dilakukan oleh robot (dengan metode ANN, perubahan nilai K_p , K_i dan K_d) robot berjalan lebih bagus atau stabil (sesuai jalur). Hasil penelitian menunjukkan bahwa dengan menggunakan metode *hybrid ANN-PID* robot *material handling* mampu berjalan melewati lintasan dengan penurunan nilai *error steady state* sebesar 33% saat 800 iterasi.

Kata Kunci: Robot *Material Handling*, Metode PID dan ANN, *trial and error*, *hybrid ANN – PID*

ABSTRACT

The research on mobile robots have been done, one of them is the material handling robot. There are several methods have been applied to develop a running process on the robot, included the PID method. The disadvantage of this method is the determining parameters process of K_i , K_p , K_d value by the trial and error method which the results of parameters value are not perfectly. Artificial Neural Network (ANN) methods is an intelligence models approach of human brain structure inspired and can be implemented using computer program. The computer program resolve a calculations process during the learning process. Hybrid ANN-PID method is the engineering method of merging between ANN and PID method. The purpose of this merging method is determining of the parameters of K_p , K_i , K_d are realistic and no longer manually. In this study used a prototype of material handling robot and an oval shaped track to be traversed by the material handling robot. By using the trial and error method for specified of K_p , K_i and K_d value as an initialitation, then the robot will run following the track, beginning by robot walking isare not suit of the track, but along the learning process by a robot (with the ANN method, changes the value of K_p , K_i and K_d) robot can run better or stabile (on track). The results, by using hybrid ANN-PID method the material handling robot can follow the track with value of error stady state decrease 33% when 800 iterations.

Key Words: Material Handling Robot, PID and ANN method, trial and error, Hybrid ANN-PID.

1. Pendahuluan

Penelitian pada *mobile robot* telah banyak dilakukan, salah satunya yaitu pada robot *material handling*. Proses berjalan pada robot ini menggunakan prinsip yang sama dengan robot *line follower*, yaitu robot yang dapat mengikuti garis dan didukung oleh rangkaian komponen elektronika serta dilengkapi dengan roda yang digerakkan oleh motor [1]. Berbagai macam metode sudah diterapkan untuk mengembangkan proses berjalan pada robot ini, salah satunya yaitu menggunakan metode PID.

PID (*Proportional Integral Derivative*) merupakan sebuah mekanisme pengontrolan yang biasa digunakan untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut [2]. Terdapat beberapa metode dalam penentuan parameter PID (K_p , K_i dan K_d), salah satunya yaitu dengan menggunakan metode *trial and error*. Proses penentuan parameter PID dengan metode *trial and error* masih dilakukan secara manual yang hasilnya belum tentu tepat. Dalam kasus ini, metode alternatif yang digunakan adalah metode *Artificial Neural Network* (ANN). Metode ANN ini digunakan untuk mempengaruhi nilai K_p , K_i dan K_d dari kontroler PID.

ANN merupakan paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi [3]. Berdasarkan fungsi-fungsi dan manfaat dari metode PID dan metode ANN, maka pada penelitian ini akan mengkombinasikan kedua metode tersebut agar nilai K_p , K_i dan K_d dapat bersifat variabel, sehingga saat metode kombinasi atau *hybrid ANN - PID* diaplikasikan pada *prototype robot material handling*. Harapannya adalah proses berjalan robot *material handling* yang sesuai konsep *line follower* dapat menyesuaikan dengan kondisi lingkungan.

2. Kajian Pustaka

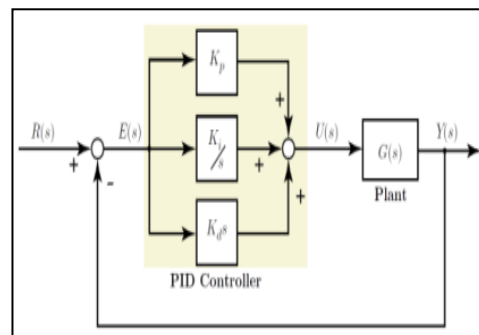
A. Robot Material Handling

Salah satu proses yang paling banyak digunakan dalam dunia industri adalah proses *material handling* atau penanganan material, dimana *material* harus dipindahkan dari satu lokasi ke lokasi lainnya. Material tersebut

harus berpindah dengan posisi yang tepat dan dalam waktu yang tepat pula. Dalam proses penanganan material, aplikasi dari sebuah *mobile robot* juga diperlukan untuk memudahkan pekerjaan ini, salah satunya yaitu menggunakan robot *material handling*. Robot *material handling* merupakan sebuah robot yang berfungsi untuk memindahkan suatu *material* dari suatu tempat ke tempat yang lain, dengan waktu yang akurat pada lokasi yang tepat [4].

B. Line Follower PID (Proportional Integral Derivative)

Sistem control yang digunakan pada robot *line follower*, dapat berupa *open loop* maupun *close loop*. Kontrol robot *line follower* dengan *close loop* dapat menggunakan PID [5,6,7]. PID (*Proportional - Integral - Derivativecontroller*) merupakan pengontrol untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Komponen kontrol PID ini terdiri dari tiga jenis, yaitu Proporsional, Integratif dan Derivatif. Ketiganya dapat digunakan bersamaan maupun sendiri - sendiri tergantung dari respon yang kita inginkan terhadap suatu sistem [7].



Gambar 1. Sistem Kontrol PID [8]

Penjelasan dari masing-masing komponen adalah sebagai berikut:

1. Pengendali Proporsional

Pengendali proporsional bertindak sebagai penguat yang mampu mengubah *output* dari sistem secara proporsional tanpa memberikan efek dinamik pada kinerja pengendali tersebut. Respon dari pengendali proporsional dapat dinyatakan dengan Persamaan (1) :

$$P_{out} = K_p \cdot E(t) \dots \dots \dots (1)$$

dimana :

P_{out} = *Output* dari pengendali proporsional

K_p = Konstanta gain
 $E(t)$ = *Error* yang dinyatakan dalam waktu secara kontinu

2. Pengendali Integral.

Pengendali integral merupakan pengendali yang berfungsi untuk memperbaiki respon tunak/*steadystate* dari sistem sehingga pengendali ini mampu memperkecil sistem *error*. Respon dari pengendali integral dapat dinyatakan dalam Persamaan (2).

$$I_{out} = K_i \int_0^t E(\tau) d\tau \dots\dots\dots(2)$$

dimana:

- I_{out} = *output* pengendali integral
- K_i = konstanta integral
- $E(\tau)$ = *error* yang dinyatakan secara kontinu
- τ = variabel integrasi

3. Pengendali *Derivative*

Pengendali *derivative* merupakan suatu pengendali yang berfungsi untuk memperbaiki respon transien dari sistem. Respon pengendali *derivative* dapat dinyatakan dalam Persamaan (3).

$$D_{out} = K_d \frac{d}{dt} E(t) \dots\dots\dots(3)$$

dimana:

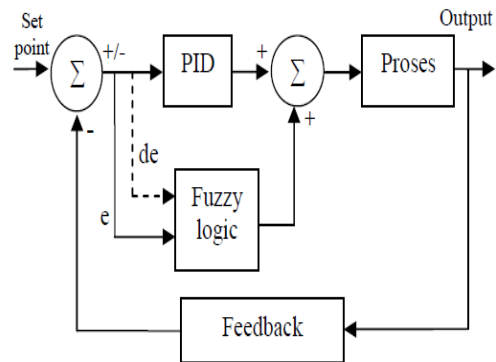
- D_{out} = *output* pengendali *derivative*
- K_d = konstanta *derivative*
- $E(t)$ = *error* yang dinyatakan dalam waktukontinu

Nilai *output* dari pengendali PID didapatkan berdasarkan penjumlahan dari masing-masing komponennya sebagai berikut:

$$MV(t) = K_p \cdot E(t) + K_i \int_0^t E(\tau) d\tau + K_d \frac{d}{dt} E(t) \dots\dots\dots(4)$$

Penelitian tentang penerapan PID pada robot *line follower* menghasilkan kesimpulan bahwa untuk menghasilkan kontrol yang optimal harus dilakukan *trial and error* untuk menentukan parameter K_p , K_i dan K_d . Metode penentuan nilai/*tuning* parameter PID ini bisa melalui cara konvensional (langsung ditentukan) atau melalui penerapan metode osilasi *Ziegler-Nichols* [5,6]. Pada proses *tuning* kontrol PID dengan menggunakan metode osilasi *Ziegler-Nichols* dimulai dengan memberikan nilai 0 pada parameter T_i dan T_d , kemudian nilai K_p dinaikan sampai didapat grafik posisi *tracer* yang berkesinambungan. Nilai T_i dan T_d ini dicari dengan menghitung

nilai K_{cr} yang merupakan nilai K_p saat osilasi dan P_{cr} yang merupakan periode kesinambungan dari grafik osilasi. Berdasarkan metode osilasi *Ziegler-Nichols* dapat diterapkan pada proses tuning parameter PID robot *auto-line follower*, proses *tuning* parameter PID bisa dilakukan lebih singkat tanpa perlu *trial and error* parameter K_i dan K_d [6]. Metode PID untuk kontrol robot *line follower* ini dapat juga digabungkan dengan sistem yang berbasis *expert system*, yaitu fuzzy. Seperti yang ditampilkan pada Gambar 2, konsep yang digunakan pada *hybird PID-Fuzzy Logic* ini, adalah dimana hasil dari *output* PID $u(t)$ dijumlahkan dengan *output* Fuzzy [7].



Gambar 2. Blok Hybrid PID-Fuzzy Logic [7]

Pada perancangan sistem kontrol *fuzzy logic* tentunya harus berdasarkan pada pengetahuan dari *universe of discourse* yang akan digunakan untuk menentukan *membership function* dari *fuzzification* dari *input* maupun *output*. Sistem kontrol dengan melibatkan *fuzzy logic* ini tidak melibatkan fungsi matematik kompleks untuk menyelesaikan masalah sistem kontrol, tapi dalam penentuannya tetap memerlukan *expert*. Berdasarkan hasil pengujian yang dilakukan, metode *fuzzy logic* yang digabungkan dengan kontroler PID pada robot *line follower* untuk mempercepat nilai *recovery time* dari sistem. Hal ini bisa dilakukan karena permasalahan telah dipetakan sedemikian rupa kedalam derajat keanggotaan *fuzzy*, sehingga tanggapan respon lebih cepat [7].

C. *Artificial Neural Network*

Berbeda dengan *fuzzy logic* yang masih mengandalkan pemikiran *expert*, *artificial neural network* (ANN) dapat secara mandiri untuk menentukan parameternya. ANN atau

Jaringan Syaraf Tiruan ditemukan pertama kali oleh McCulloch dan Pitts pada tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem *neural* akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*.

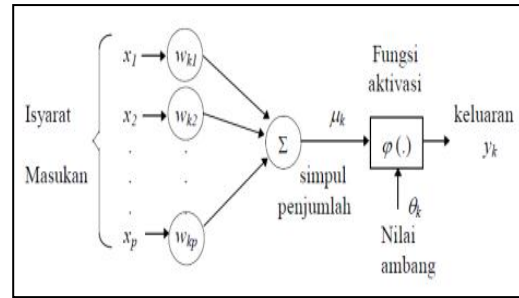
Selanjutnya pada tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan baru yang terdiri dari beberapa lapisan yang disebut *perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasi-nya. Widrow dan Hoff pada tahun 1960 mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan delta (kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan layer tunggal (*single layer*). Rumelhart bersama McClelland pada tahun 1986 mengembangkan *perceptron* menjadi *backpropagation*, yang memungkinkan jaringan diproses melalui beberapa layer [9]. Secara umum suatu jaringan syaraf dibentuk atas sejumlah *neuron* sebagai unit pengolah informasi sebagai dasar operasi untuk menjalankan fungsi atau tugasnya. Model susunan *neuron* dapat ditunjukkan pada (Gambar 3).

Secara matematis, untuk *neuronk* berlaku Persamaan (5):

$$\mu_k = \sum_{j=1}^p w_{kj} x_j \dots\dots\dots (5)$$

$$y_k = \varphi(\mu_k - \theta_k) \dots\dots\dots (6)$$

dengan x_1, x_2, \dots, x_p adalah isyarat masukan, $w_{k1}, w_{k2}, \dots, w_{kp}$ adalah bobot sinapsis untuk *neuronk*, μ_k adalah keluaran kombinasi linier, θ_k adalah nilai ambang, $\varphi(\cdot)$ adalah fungsi aktivasinya, dan y_k adalah isyarat keluaran *neuronk* [10].



Gambar 3. Model *Neuronk* [10]

Fungsi Aktivasi

Fungsi aktivasi merupakan bagian penting dalam tahapan perhitungan keluaran dari suatu algoritma ANN. Berikut ini fungsi aktivasi yang dipergunakan dalam ANN [13].

1. Fungsi *Sigmoid Biner*

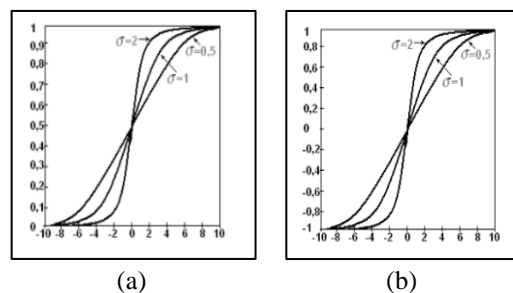
Fungsi *sigmoid biner* memiliki nilai range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk ANN yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Fungsi *sigmoid biner* dirumuskan seperti Persamaan (7).

$$y = \frac{1}{1+e^{-x}} \dots\dots\dots (7)$$

2. Fungsi *Sigmoid Bipolar*

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, hanya saja *output* dari fungsi ini memiliki *range* antara -1 sampai 1. Fungsi *sigmoid bipolar* dirumuskan sebagai berikut:

$$y = \frac{1-e^{-x}}{1+e^{-x}} \dots\dots\dots (9)$$



Gambar4. Fungsi Aktivasi [13]

Backpropagation Neural Network

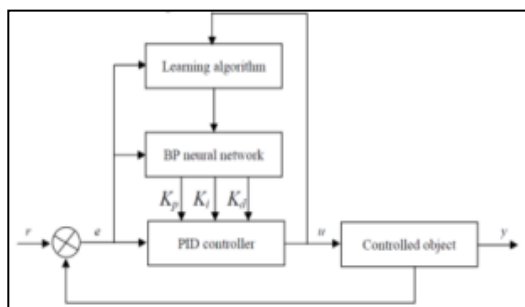
Metode pelatihan *Backpropagation Neural Network* (BP-NN) pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart & McClelland. *Backpropagation neural network* merupakan tipe jaringansaraf tiruan yang menggunakan metode pembelajaran terbimbing (*supervised learning*). Pada *supervised learning* terdapat

pasangan data *input* dan *output* yang dipakai untuk melatih ANN hingga diperoleh bobot penimbang (*weight*) yang diinginkan.

Penimbang itu sendiri adalah sambungan antar lapis dalam ANN. Algoritma ini memiliki proses pelatihan yang didasarkan pada interkoneksi yang sederhana, yaitu apabila keluaran memberikan hasil yang salah, maka penimbang dikoreksi agar *error* dapat diperkecil dan tanggapan ANN selanjutnya diharapkan dapat mendekati nilai yang benar. BP-NN juga berkemampuan untuk memperbaiki penimbang pada lapis tersembunyi (*hidden layer*) [10]. Penemuan NN dengan algoritma *backpropagation* menggunakan *multi layer perceptron* (MLP) menjadi salah satu alternatif penyelesaian permasalahan yang susah untuk diturunkan dalam model matematikanya [10]. BP-NN ini hanya memiliki koneksi (*synapse*) antara *input*, *hidden layer* dan *output*. Sedangkan untuk mendeteksi kesamaan pola *output* pembelajaran dan pola *output* yang dihasilkan, maka *error* yang dihasilkan kedua *output* tersebut akan dipropagasikan balik selama proses pelatihan. Agar mendapatkan keseimbangan antara kemampuan untuk mengenali *input* dan pola *output*, maka jaringan akan dilatih. Pelatihan tersebut akan berjalan sampai pola *output* tersebut serupa dengan pola yang digunakan selama pelatihan [10].

Hybrid ANN – PID

Sistem kontrol ini terdiri dari dua bagian, yaitu kontrol PID dan ANN, dimana kontrol PID langsung mengontrol objek yang dikontrol dengan *close loop*, dan parameter kontrol K_p , K_i dan K_d yang secara *online* disesuaikan dengan kondisi *output* ANN. Melalui jaringan saraf tiruan *self-learning*, *output* jaringan saraf tiruan akan menyesuaikan parameter sistem kontrol PID [11].



Gambar5. Sistem Kontrol ANN-PID [11]

Untuk model sistem yang belum ditentukan, metode eksperimental dapat digunakan untuk merancang kontrol PID [12]. Fungsi kontrol PID dapat ditulis sebagai berikut :

$$GS(s) = K_p(1 + \frac{1}{s\tau_i} + S\tau_d) \dots\dots\dots(10)$$

di mana K_p adalah gain proporsional, τ_i adalah integral waktu, dan τ_d adalah turunan waktu.

Output dari ANN ditambahkan ke kontrol PID konvensional untuk menyesuaikan sistem dari kontroler PID *online* sesuai dengan perubahan sinyal *error* [12]. Hal ini dilakukan untuk memastikan bahwa ANN akan memberikan perubahan yang diperlukan dalam sinyal kontrol (*u*)*online*. Tindakan kontrol dari kontroler PID setelah menggunakan *output* ANN $y(t)$ dapat dituliskan sebagai berikut :

$$u = K_{PNN} e(t) + K_{INN} \int e dt + K_{DNN} \frac{d e(t)}{dt} \dots\dots\dots(11)$$

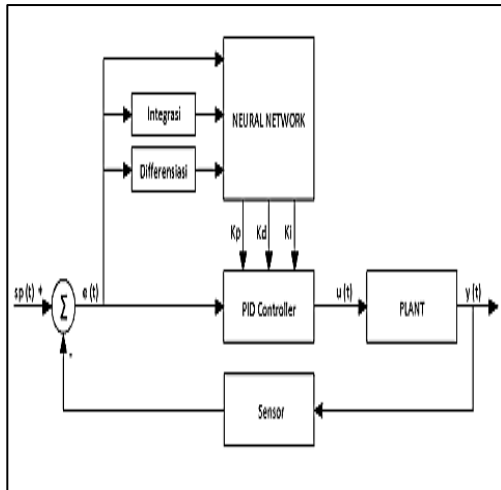
dimana :

- $K_{PNN} = y(t) \cdot K_p$
- $K_{INN} = y(t) \cdot K_i$
- $K_{DNN} = y(t) \cdot K_d$

3. Metode Penelitian

A. Perancangan Penelitian

Pada penelitian ini, rancangan penelitian dibagi menjadi beberapa bagian utama, yaitu perancangan sistem perangkat keras (*hardware*), perancangan sistem perangkat lunak (*software*) dan perancangan konstruksi alat. Metode yang digunakan untuk menunjang penelitian ini adalah *hybrid ANN – PID*, dimana keluaran ANN digunakan sebagai *auto tuner* PID untuk mengontrol proses berjalannya robot.

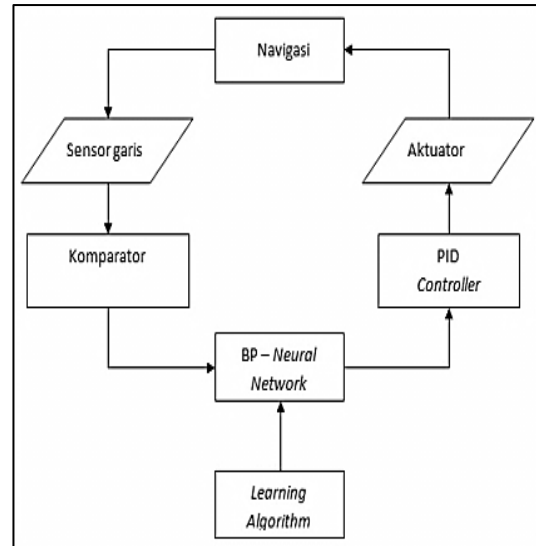


Gambar6. Blok Diagram Software

Nilai parameter PID akan ditentukan terlebih dahulu sebagai data inisialisasi. Nilai parameter ini didapatkan dengan metode *trial and error* dengan tujuan agar tahap pembelajaran ANN untuk mencari parameter yang tepat tidak terlalu lama. Keluaran dari metode ANN ini akan menambah atau mengurangi nilai dari parameter PID sampai didapatkan nilai *error* terkecil atau sampai pergerakan robot dirasakan sudah cukup stabil. *Input* dari sistem ini adalah nilai *error* yang didapatkan dari sensor garis. Sensor garis ini akan menunjukkan posisi robot, selanjutnya akan diketahui nilai *error* dari selisih nilai *set point* yang telah ditentukan sebelumnya. Nilai dari masukan akan diproses untuk selanjutnya digunakan sebagai pengontrol pergerakan robot dengan kontrol PID. Sinyal *error* juga digunakan sebagai masukan untuk proses *selflearning* pada metode ANN. Keluaran dari ANN akan mempengaruhi nilai konstanta K_p , K_d dan K_i untuk kontrol PID.

B. Flowchart Sistem Keseluruhan

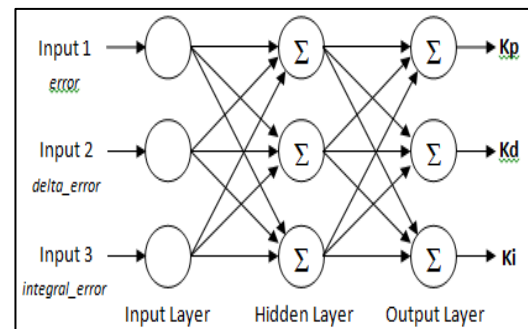
Berdasarkan *flowchart* sistem di atas, dijelaskan bahwa proses kerja robot dimulai saat sensor mendeteksi garis hitam. Jika sensor tidak mendeteksi garis hitam, maka robot akan bergerak mencari garis hitam. Apabila sensor mendeteksi adanya garis hitam, maka robot akan dikontrol pergerakannya untuk mengikuti garis hitam sebagai jalur lintasannya. Pada sistem navigasi robot terdapat beberapa proses yang akan dilakukan seperti proses pembelajaran metode ANN berdasarkan pola masukan dari sensor sehingga sistem ini dapat digunakan untuk mengontrol *actuator* berupa motor yang dapat menggerakkan posisi robot.



Gambar7. Flowchart System

C. Algoritma Program

Struktur ANN *backpropagation* pada program ini terdiri dari 3 bagian yang meliputi *input layer*, *hidden layer* dan *output layer*.



Gambar8. Struktur Jaringan ANN

Pada struktur jaringan ANN yang diimplementasikan pada penelitian ini menggunakan 3 *node input* (*error*, *delta error* dan *integral error*), 3 *node hidden* dan 3 *node output* yang merupakan variabel dari parameter K_p , K_d dan K_i untuk diterapkan pada pengontrol PID. Hal pertama yang dilakukan yaitu inisialisasi masukan untuk *node input* ANN. Langkah selanjutnya yaitu memberi nilai awal secara *random* bagi seluruh *weight* antara *input - hidden layer* dan *hidden layer - output*.

Feed Forward

Hitung semua keluaran jaringan di unit *input* dan *hidden layer* dengan Persamaan (12).

$$n1 = \sum_{i=0}^m \sum_{j=0}^n x_j * w_{ji} \dots\dots\dots(12)$$

Karena data keluarannya bernilai positif antara 0 sampai dengan 1, maka fungsi aktivasi *node*

hidden yang digunakan adalah fungsi biner *sigmoid* (Persamaan (13)).

$$f(n1) = \frac{1}{1 + e^{-\alpha n1}} \dots\dots\dots(13)$$

Kemudian menghitung semua keluaran jaringan di *node hidden* dan *output layer* dengan Persamaan (14).

$$n2 = \sum_{i=0}^m \sum_{j=0}^n y_j * u_{ji} \dots\dots\dots(14)$$

Karena data keluarannya bernilai positif antara 0 sampai dengan 1, maka fungsi aktivasi *node output* yang digunakan adalah fungsi biner *sigmoid*:

$$f(n2) = \frac{1}{1 + e^{-\alpha n2}} \dots\dots\dots(15)$$

Backward

Hitung nilai *error* berdasarkan kesalahan di setiap unit keluaran/*output* dengan Persamaan (16).

$$\delta_k = \delta_{net,k} f'(y_{net,k}) = \delta_{net,k} (y_i (1 - y_i)) \dots\dots\dots(16)$$

dimana δ_k merupakan kesalahan yang akan digunakan dalam perubahan bobot *layer* sebelumnya.

Menghitung nilai *error* berdasarkan kesalahan di setiap unit *hidden* dengan Persamaan (17)

$$\delta_j = \delta_{net,j} f'(z_{net,j}) = \delta_{net,j} (z_j (1 - z_j)) \dots\dots\dots(17)$$

dimana untuk menentukan nilai $\delta_{net,j}$ adalah dengan Persamaan (18).

$$\delta_{net,j} = \sum_{i=0}^m \sum_{j=0}^n \delta_k * u_{ij} \dots\dots\dots(18)$$

Modifikasi Bobot

Langkah selanjutnya yaitu menentukan bobot (*weight*) yang menuju ke *node hidden layer* dengan menggunakan Persamaan (19).

$$w_{ji}(k+1) = \sum_{j=0}^n \sum_{i=0}^m w_{ji}(k) + x_j * \delta_j \dots\dots\dots(19)$$

Menentukan bobot (*weight*) yang menuju ke *node output layer* dengan menggunakan Persamaan (20).

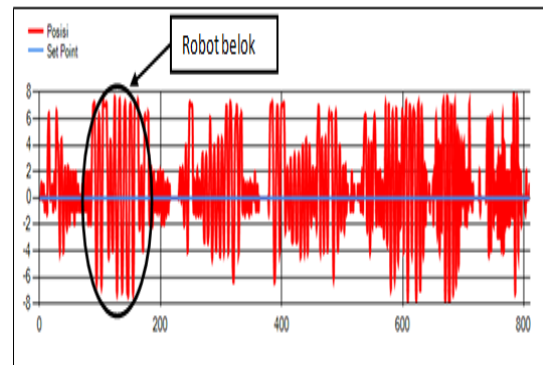
$$u_{ji}(k+1) = \sum_{i=0}^m \sum_{j=0}^n u_{ji}(k) + y_j * \delta_j \dots\dots\dots(20)$$

Keluaran dari ANN ini akan digunakan untuk mempengaruhi parameter PID yaitu nilai *Kp*, *Kd* dan *Ki* yang akan mengontrol kecepatan motor atau mengatur pergerakan robot.

4. Pengujian dan Analisa Data

A. Respon Kontrol Robot dengan Metode PID

Penentuan nilai parameter kontrol PID pada penelitian ini menggunakan metode *trial anderror*. Setelah melakukan beberapa percobaan dengan nilai parameter yang berbeda-beda, maka didapatkan nilai *Kp* = 10, *Ki* = 0.5 dan *Kd* = 5. Nilai parameter ini untuk selanjutnya akan digunakan sebagai data inisialisasi awal pada percobaan dengan metode *hybrid ANN-PID* agar proses pembelajaran yang dilakukan ANN untuk mencari parameter yang tepat tidak membutuhkan waktu yang terlalu lama. Hasil pengujian respon posisi robot menggunakan metode PID dapat dilihat pada (Gambar 9).



Gambar9. Grafik Respon Kontrol Robot Dengan Metode PID

Dari hasil pengujian robot seperti yang ditunjukkan pada (Gambar 10), dapat dilihat bahwa pergerakan robot sudah mendekati *set point* atau sumbu 0. Akan tetapi pola pergerakan robot masih belum stabil, dengan nilai *error steady state* rata-rata yaitu sebesar 2.89 saat iterasi = 200, dan nilai *error steady state* rata-rata sebesar 2.64 saat iterasi = 800. Nilai *error steady state* rata-rata pada setiap 200 iterasi dalam pengujian ini dapat ditunjukkan pada Tabel 1.

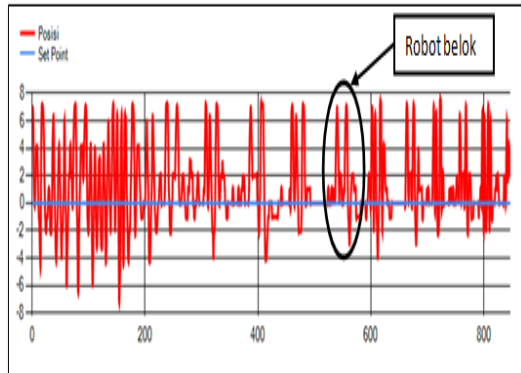
Tabel 1. Nilai *Error Steady State* Rata-Rata Pengujian PID

Pengujian	Error Steady State Rata-rata			
	iterasi 200	iterasi 400	iterasi 600	iterasi 800
PID	2.89	2.58	2.57	2.65

B. Respon Kontrol Robot dengan Metode Hybrid ANN - PID

Pengujian selanjutnya dilakukan untuk melihat bagaimana respon kontrol robot yang

menerapkan metode *hybrid* ANN–PID dan perubahan nilai parameter PID. Grafik respon kontrol dapat dilihat pada (Gambar 11).



Gambar 10. Respon Kontrol Dengan Metode *Hybrid* ANN-PID.

Tabel 2. Nilai *Error Steady State* Rata-Rata Pengujian *Hybrid* ANN-PID

Pengujian	Error Steady State Rata-rata			
	iterasi 200	iterasi 400	iterasi 600	iterasi 800
ANN-PID	3.21	2.5	2.42	2.41

Dari hasil pengujian robot seperti yang ditunjukkan pada (Gambar 11), dapat dilihat bahwa pergerakan robot juga sudah mendekati *set point* atau sumbu 0. Nilai *error steady state* rata-rata ditampilkan pada tiap pertambahan 200 iterasi untuk melihat seberapa besar pengurangan nilai *error steady state* atau proses perbaikan yang dijalankan oleh metode ANN. Nilai *error steady state* rata-rata pada pengujian metode *hybrid* ANN – PID dapat ditunjukkan pada Tabel 2.

Dari data pengujian seperti yang ditunjukkan pada Tabel 2, dapat dilihat bahwa nilai *error steady state* rata-rata mengalami pengurangan nilai setiap pertambahan iterasi. Saat iterasi = 200, nilai *error steady state* rata-rata sebesar 3,21. Kemudian saat iterasi = 800, nilai *error steady state* rata-rata sebesar 2,41. Jadi pada pengujian ini, nilai *error steady state* rata-rata mengalami penurunan sebesar 33%. Hal ini menunjukkan bahwa pergerakan robot akan semakin stabil seiring dengan bertambahnya waktu atau bertambah banyaknya iterasi yang telah dijalankan oleh program.

Pengujian untuk melihat perubahan nilai parameter PID dilakukan dengan menampilkan

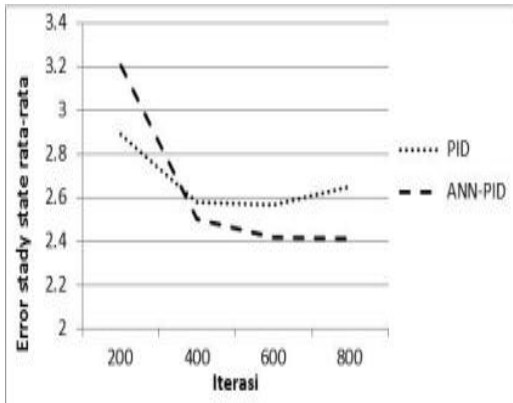
nilai K_p , K_i dan K_d setiap pertambahan 200 iterasi. Perubahan nilai parameter PID ditunjukkan pada Tabel 3.

Tabel 3. Perubahan Nilai Parameter PID

Iterasi	Parameter PID		
	K_p	K_i	K_d
0	10	0.5	5
200	11.32	0.51	5.04
400	12.03	0.2	7.81
600	12.04	0.23	8.02
800	12.08	0.21	8.01

Berdasarkan hasil pengujian seperti yang ditunjukkan pada Tabel 3, nilai parameter PID saat iterasi = 0 merupakan nilai inisialisasi awal. Kemudian untuk iterasi selanjutnya, nilai parameter PID akan dipengaruhi oleh keluaran dari ANN sehingga nilai parameter PID mengalami perubahan untuk mengurangi nilai *error steady state*. Nilai parameter PID tidak mengalami perubahan nilai yang cukup besar dari iterasi = 400 sampai iterasi = 800.

Setelah didapatkan semua nilai *error steady state* rata-rata maka hasil perbandingannya dapat dilihat pada (Gambar 12), dimana terlihat bahwa pada saat iterasi 200 nilai *Hybird* ANN-PID lebih besar daripada PID, tetapi pada saat system mulai berjalan mencapai iterasi ke 800 nilai *error steady state* rata-rata *Hybird* ANN-PID lebih rendah dari pada PID. Hal ini memperlihatkan bahwa pada saat sistem berjalan, proses *Hybird* ANN-PID akan selalu memperbaiki nilai *error steady state* untuk mencapai nilai minimalnya, atau mendekati nilai 0. Berdasarkan hasil ini dapat disimpulkan bahwa pada permasalahan di penelitian ini sistem *Hybird* ANN-PID lebih baik daripada PID, karena respon *error steady state* pada *Hybird* ANN-PID memiliki tren menurun pada saat sistem ini berjalan.



Gambar 11.Perbandingan *Error Steady State* rata-rata antara metode PID dengan *hybrid ANN-PID*

5. Kesimpulan

Berdasarkan perancangan, pengujian, dan analisis yang telah dilakukan pada penelitian ini, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Nilai keluaran ANN dapat digunakan untuk mempengaruhi nilai parameter PID sehingga dapat mengurangi nilai *error steady state*. Nilai akhir parameter PID yang dikombinasikan dengan metode ANN yaitu $K_p = 12.08$, $K_i = 0.21$ dan $K_d = 8.01$ dengan penurunan nilai *error steady state* sebesar 33% saat 800 iterasi.
2. Pada pengujian *prototype* robot *material handling* yang menggunakan metode *hybrid ANN-PID*, robot telah mampu berjalan mengikuti lintasan dengan nilai *error steady state* rata-rata mengalami penurunan seiring dengan bertambahnya waktu atau bertambah banyaknya iterasi yang telah dijalankan oleh program.

DAFTAR PUSTAKA

[1] Andriansya, (2009), *Kontrol PID pada Robot Line Follower*, Skripsi Departemen Teknik Elektro Universitas Komputer Indonesia.

[2] Jeni Agus Nurhuda, (2012), *Kontrol PID (Sebuah Teori)*, UKM Robotic STMIK Teknokrat Lampung.

[3] Ali Ridho Barakbah, (2011), *Kecerdasan Buatan*, Soft Computation Research Group EEPIS - Institut Teknologi Sepuluh Nopember Surabaya.

[4] Musa Bin Mailah, Hishamuddin, Endra Pitowarno, (2007), *Intelligent Material Handling Mobile Robot for Industrial Purpose*, Pusat Pengurusan Penyelidikan, Universiti Teknologi Malaysia.

[5] Saputro G, (2012), “ *Sistem Kontrol Motor Robot Line Follower Berbasis Mikrokontroler Atmega32 Menggunakan Algoritma PID (Proporsional Integral Derivatif)*”,Skripsi Sekolah Tinggi Manajemen Informatika Dan Komputer AMIKOM, Yogyakarta.

[6] Setiawan B I, () “ *Perancangan Robot Auto Line Follower Yang Menerapkan Metode Osilasi Ziegler-Nichols Untuk Tuning Parameter PID Pada Kontes Robot Indonesia* “, Jurusan Teknik Elektro Universitas Brawijaya, Malang.

[7] Pratama I Putu A M, Suwenden I N, Swamardika I B A, (2013), “ *Sistem Kontrol Pergerakan Pada Robot Line Follower Berbasis Hybrid PID-Fuzzy Logic* “, Prosiding Conference on Smart-Green Technology in Electrical and Information Systems, Bali, 14-15 November 2013.

[8] Robert A. Paz, (2001), *The Design of the PID Controller*, Klipsch School of Electrical and Computer Engineering.

[9] Carlos Gershenson, (2003), *Artificial Neural Networks for Beginners*, Springer, Berlin.

[10] Hidayatno, Achmad, (2012), *Pengantar Jaringan Syaraf*, Departemen Teknik Elektro UNDIP Semarang.

[11] Luoren Liu, Luo Jinling, (2011), *Research of PID Control Algorithm Based on Neural Network*, Department of Electronics and Information Engineering Loudi Vocational College, Loudi 41700, Hunan, China.

[12] Saad Zaghlul Saeed Al-Khayyt, (2013), *Tuning PID Controller by Neural Network for Robot Manipulator*,

Department of Mechatronics
Engineering, Mosul University – Iraq.

[13] Kusumadewi Sri, (2003), *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta