

SIMULASI FOLDING MACHINE DENGAN PID, P, PI, PD DAN FUZZY-PD (PROPORTIONAL DIFFERENTIAL)

Wahyu Setyo Pambudi¹, Jan Putra. B. A. S. Pelawi²
Jurusan Teknik Elektro, Universitas Internasional Batam
Jl. Gajahmada Baloi Sei Ladi Batam 29422
Telp. (0778) 7437111 ext. 130, Faks. (0778) 7437112
E-mail: 1) pambudi@uib.ac.id; yoe2fa@yahoo.com
2) janpoetra22@gmail.com

ABSTRAKS

Sebelum melakukan perancangan mesin pelipat baju (*Folding Machine*), maka perlu ditentukan terlebih dahulu sistem kontrol yang akan digunakan pada alat tersebut. Sistem kontrol yang digunakan merupakan sistem kontrol posisi dimana nantinya pelipat dari *folding machine* akan bergerak menuju sudut tertentu agar baju terlipat dengan baik. Sistem kontrol ini nantinya akan diterapkan pada motor DC. Untuk menentukan sistem kontrol yang terbaik, maka dipergunakan simulasi perlipat pada *folding machine*. Pada simulasi yang dilakukan digunakan konsep dinamik arm manipulator agar didapat hasil mendekati kondisi real. Pada simulasi digunakan berbagai sistem kontrol yaitu PID, P, PI, PD dan FuzzyPD. Dari simulasi yang dilakukan didapatkan hasil FuzzyPD memiliki respon baik dengan rise time 0,01 s dengan overshoot 3° dan error steady state sebesar 2° , saat dibandingkan dengan kontroler lainnya dengan nilai $K_p=0,5, K_i=0,001, K_d=3$. Namun setelah dilakukan proses tuning PID dengan menurunkan K_d menjadi 1. maka didapatkan kontroler PID yang terbaik dengan rise time 0,005s dengan tidak memiliki overshoot dan error steady state.

Kata Kunci: PID, Fuzzy-PD, Arm Manipulator

1. PENDAHULUAN

Jenis usaha mikro yang bergerak dibidang jasa salah satunya adalah usaha *laundry*. Usaha ini banyak muncul didaerah perkotaan, hal ini disebabkan karena sebagian besar masyarakat kota adalah pekerja sehingga sangat memerlukan jasa *laundry* ini. Proses *laundry* terdiri dari proses mencuci, mengeringkan dan menyetrika serta melipat pakaian. Sebagian besar pelaku usaha jenis ini berada di golongan usaha menengah ke bawah, sehingga perlu dilakukan usaha-usaha untuk membantu meningkatkan usaha jenis ini sehingga dapat bersaing.

Banyak yang perlu dilakukan untuk membantu para pelaku usaha *laundry* ini salah satunya adalah dengan memberikan sentuhan teknologi pada proses produksinya. Teknologi yang dapat diterapkan adalah dengan memanfaatkan teknologi *system control* untuk proses melipat pakaian. Alat untuk melipat baju atau *folding machine* ini menggunakan *flip* yang digerakkan oleh motor DC yang dikontrol berdasarkan posisi *flip* yang diinginkan. Prinsip kerja *flip* pada *folding machine* mirip dengan *arm manipulator* dengan 1 DOF (*degree of freedom*).

Sebelum dilakukan pembuatan alat yang sebenarnya, hendaknya perlu dilakukan penelitian awal berupa pembuatan simulasi *folding machine*. Pembuatan simulasi yang dibuat perlu menerapkan *kinematic* dan *dynamic* dari *system folding machine*, karena secara keseluruhan pengujian program simulasi terdapat hubungan antara *inverse kinematic*, *forward kinematic* serta *dynamic*. Ketiga persamaan

itu tidak bisa dilepaskan dalam analisa robot khususnya *arm manipulator* (Pambudi, W. S. et al. 2014)

Simulasi yang dirancang menggunakan simulasi *arm manipulator* 1 DOF untuk pelipat sisi samping pakaian dengan menggunakan model *dynamic* untuk memperoleh hasil yang mendekati kondisi riil. Pada simulasi digunakan beberapa kontroler yang akan dibandingkan responnya, yaitu kontrol PID, kontrol P, kontrol PI, kontrol PD dan Fuzzy PD. PID dipilih karena *system control* ini merupakan *system control* konvensional yang umum digunakan. Sedangkan untuk penggunaan Fuzzy PD digunakan metode sistem kendali Fuzzy merupakan alternatif sistem kendali modern yang mudah karena tidak perlu mencari model matematis dari suatu sistem, tetapi tetap efektif karena memiliki respon sistem yang stabil, *redundant* atau *fault tolerant* yang artinya masih dapat bekerja meskipun adanya pengurangan beberapa *rules*, maupun terjadi kesalahan-kesalahan kecil dalam pemrogramannya tanpa adanya perubahan signifikan (Resmana et al. 1999). Kontrol yang digunakan diharapkan memiliki respon waktu yang cepat namun tidak terjadi *overshoot* dan osilasi. Pada simulasi ini, keluaran *system control* berupa nilai torsi untuk dapat mencapai sudut target yang diberikan.

Berdasarkan hal ini maka perlu dilakukan penelitian awal melalui pembuatan simulasi robot *Folding Machine* serta analisisnya sebelum dilakukan pembuatan atau realisasi. Sistem kontrol yang digunakan adalah PID, P, PI, PD dan fuzzy-PD, harapannya adalah dengan adanya program

simulasi *Folding Machine* dapat menjadi dasar untuk menentukan parameter-parameter yang diperlukan pada proses realisasinya nanti.

2. LANDASAN TEORI

2.1 Model Dinamik Arm Manipulator

Umumnya, robot manipulator digerakkan oleh aktuatur *electric, hydraulic* atau *pneumatic*, yang mana menerapkan torsi yang dihasilkan oleh aktuatur pada joint (sendi) robot untuk menggerakkan lengan manipulator. Model *dynamic arm manipulator* menjelaskan bagaimana robot bergerak terhadap respon atau pengaruh dari torsi ini (Pambudi, W. S. et al. 2014).

Model *dynamic* dari suatu robot menjelaskan hubungan pergerakan dan gaya (torsi) yang mempengaruhi robot. Berbeda dengan model *kinematic* yang hanya menghubungkan sudut dan posisi, maka model *dynamic* ini menghubungkan gaya dan torsi terhadap pergerakan yang dalam *arm manipulator* berhubungan dengan percepatan sudut, kecepatan sudut dan besar sudut itu sendiri, akibat dari model pergerakan rotasi dari *arm manipulator*. *Dynamic* ini membahas hubungan secara matematis dari besar sudut pada *joint* yang bergantung kepada besar torsi yang diberikan untuk menggerakkan lengan / *arm* tersebut serta parameter dimensional dari robot itu sendiri (David, I. et al. 2012).

Terdapat 2 macam pendekatan dalam model dinamik robot, yaitu *direct* dan *inverse dynamic*.

2.1.1 Direct Dynamic Model

Model ini merupakan model perhitungan untuk memperoleh besar percepatan sudut terhadap waktu dari pemberian gaya atau torsi tertentu pada *joint* dan gaya tambahan dari luar pada *end-effector*. Dari percepatan sudut ini maka akan didapat kecepatan dan besar sudut di *joint* tersebut pada Persamaan (1) (Melchiorri, C).

$$\tau(t) \Rightarrow \ddot{\theta}(t) (\dot{\theta}(t), \theta(t)) \dots\dots\dots (1)$$

2.1.1.1 Inverse Dynamic Model

Model ini merupakan perhitungan vektor torsi atau gaya yang diperoleh berdasarkan *trajectory* $\ddot{\theta}(t), \dot{\theta}(t), \theta(t)$ sesuai dengan Persamaan (2).

$$\ddot{\theta}(t), \dot{\theta}(t), \theta(t) \Rightarrow \tau(t) \dots\dots\dots (2)$$

Model *dynamic* dapat dipergunakan untuk simulasi pengujian gerak tanpa menggunakan eksperimen secara riil, untuk analisis dan sintesis sistem control yang cocok untuk *plant* serta analisa dari properti skrutural manipulator pada tahap desain. Dalam menentukan model dinamik dapat dipergunakan 2 macam pendekatan yang umum dipergunakan, yaitu:

2.1.1.1.1 Euler-Lagrange

Model pendekatan ini lebih sederhana, dan lebih cocok dipergunakan untuk memahami pengaruh perubahan parameter mekanik. Setiap lengan dianggap sebagai satu kesatuan dan model didapat secara analitik. Model ini diperoleh dari Energi Kinetik dan Energi Potensial pada manipulator. Akan tetapi model ini tidak efisien untuk komputasional.

2.1.1.1.2 Newton-Euler

Model ini efisien secara komputasional yang memanfaatkan struktur serial manipulator industri. Akan tetapi, model matematika tidak dinyatakan dalam bentuk tertutup. Kedua teknik tersebut menghasilkan hasil yang sama. Dalam pembahasan ini akan dipergunakan model pendekatan Euler-Lagrange untuk menentukan model *dynamic* secara matematis.

2.2 Persamaan Euler-Lagrange

Euler-Lagrange merupakan salah satu metode untuk menghasilkan model *dynamic* pada sistem mekanik. Semua metode menghasilkan set persamaan yang sama, akan tetapi memiliki bentuk yang berbeda yang mungkin lebih cocok untuk komputasi atau analisis. Dipergunakan analisis Lagrange untuk menurunkan model matematis karena model ini bergantung kepada properti energi dari suatu sistem mekanik untuk menghitung persamaan geraknya. Dengan analisa ini, akan lebih mudah untuk memperoleh persamaan tertutupnya dan mempermudah dalam analisa mendetail dari properti sistem.

Lagrange sebagai persamaan gerak didefinisikan sebagai besar selisih antara energi kinetik dan energi potensial dari suatu sistem, secara matematik dituliskan dalam Persamaan (3) sebagai berikut.

$$L(\theta(t), \dot{\theta}(t)) = K_E(\theta(t), \dot{\theta}(t)) - P_E\theta(t) \dots\dots\dots (3)$$

dimana :

L = Lagrange

K_E = Energi Kinetik

P_E = Energi Potensial

Persamaan gerak dari suatu mekanik sistem pada suatu koordinat atau posisi dan hubungannya dengan Lagrange dapat dinyatakan dalam Persamaan (4) sebagai berikut.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau \dots\dots\dots (4)$$

dimana:

L = Lagrange

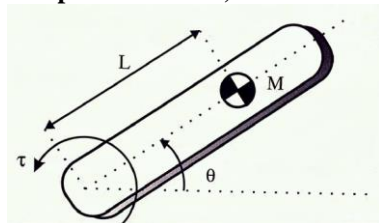
$\dot{\theta}$ = Kecepatan Sudut

θ = Besar Sudut

τ = Gaya luar yang muncul pada koordinat tersebut

Persamaan ini disebut Euler-Lagrange. Pada robot manipulator, yang merupakan gaya dapat dinyatakan sebagai torsi (τ) akibat pergerakan rotasi.

2.3 Model Dinamik pada Pendulum (Manipulator 1 DOF)



Gambar 1. Manipulator 1 DOF
Sumber : (David, I. et al. 2012)

Dalam menentukan model *dynamic* untuk manipulator 1 DOF seperti Gambar 1 diatas, maka dipergunakan persamaan (3) dan (4). Persamaan(3) untuk memperoleh Lagrange dari *system* dan persamaan (4) untuk memperoleh torsi system berdasarkan Lagrange yang diperoleh. Untuk itu diperlukan analisa untuk menentukan energi kinetik energi potensial yang bekerja pada system tersebut. Energi Kinetik dan Energi Potensial dapat ditentukan menggunakan persamaan Energi Mekanik, pada Persamaan (5) dan (6).

$$K_E = \frac{1}{2} Mv^2 \dots\dots\dots (5)$$

$$P_E = Mgh \dots\dots\dots (6)$$

dimana:

- M = massa lengan
- v = Kecepatan Linear
- h = Ketinggian
- g = Percepatan Gravitasi

Mengingat gerak manipulator merupakan gerak rotasi maka, kecepatan yang berlaku adalah kecepatan sudut, dan massa yang diterapkan dalam gerak rotasi adalah momen Inersia, maka untuk menghitung keduanya dipergunakan Persamaan (7) dan (8) berikut ini.

$$v = \dot{\theta}L \dots\dots\dots (7)$$

$$I = ML^2 \dots\dots\dots (8)$$

Dengan mensubstitusikan Persamaan (7) dan (8) pada Persamaan (5), maka didapatkan Persamaan (9) sebagai Persamaan energi *kinetic* untuk gerak rotasi sebagai berikut

$$K_E = \frac{1}{2} I\dot{\theta}^2 \dots\dots\dots (9)$$

Untuk energi potensial, h ditentukan berdasarkan letak *end-effector*, sehingga $h = L \sin\theta$, maka Persamaan energi potensialnya seperti pada Persamaan (10).

$$P_E = MgL \sin \theta \dots\dots\dots (10)$$

Setelah mendapat nilai energi potensial dan *kinetic*, maka persamaan (9) dan (10) disubstitusikan kedalam persamaan (3), sehingga Persamaan seperti pada Persamaan (11).

$$L(\theta(t), \dot{\theta}(t)) = \frac{1}{2} I\dot{\theta}^2 - MgL \sin \theta \dots\dots\dots (11)$$

Guna menentukan besar torsi maka Persamaan (11) harus dimasukkan dalam persamaan Euler-Lagrange, dengan menurunkan terlebih dahulu L terhadap θ dan $\dot{\theta}$ serta t , sehingga menjadi Persamaan (12) dibawah ini.

$$\frac{\partial L}{\partial \theta} = I\ddot{\theta}$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = I\ddot{\theta} \dots\dots\dots (12)$$

$$\frac{\partial L}{\partial \theta} = -MgL \cos \theta$$

$$\tau = I\ddot{\theta} + MgL \cos \theta$$

Bentuk umum dari *dynamic* dinyatakan dalam Persamaan (13) berikut ini.

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + g(\theta) \dots\dots\dots (13)$$

dimana :

- $M(\theta)\ddot{\theta}$ = Matrix Inersia
- $C(\theta, \dot{\theta})$ = Gaya Sentrifugal
- $g(\theta)$ = Gaya Gravitasi

2.4 Direct Dynamic

Persamaan Direct dynamic merupakan model untuk menentukan posisi atau besar sudut pada joint dengan hubungannya dengan torsi. Persamaan *direct dynamic* didapatkan dari modifikasi Persamaan (12) sehingga menjadi Persamaan (14).

$$\ddot{\theta} = \frac{(\tau - MgL \cos \theta)}{I} \dots\dots\dots (14)$$

Dengan mensubstitusikan Persamaan (8) pada persamaan diatas, maka didapatkan Persamaan (15) seperti dibawah ini.

$$\ddot{\theta} = \frac{(\tau - MgL \cos \theta)}{ML^2} \dots\dots\dots (15)$$

Sedangkan untuk menentukan kecepatan sudut, maka dilakukan integral terhadap percepatan sudut (Persamaan (16)) dan untuk menentukan besar sudut maka dilakukan integral terhadap kecepatan sudut (Persamaan (17)).

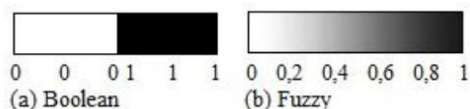
$$\dot{\theta} = \dot{\theta} + \int \ddot{\theta} dt \dots\dots\dots (16)$$

$$\theta = \theta + \int \dot{\theta} dt \dots\dots\dots (17)$$

2.5 Fuzzy Logic

Fuzzy secara harfiah berarti samar, sedangkan kebalikannya dalam hal ini adalah Crisp yang secara harfiah berarti tegas. Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut

himpunan kabur (fuzzy Set). Himpunan fuzzy di dasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan real pada interval [0,1]. Nilai keanggotaannya menunjukkan bahwa suatu nilai dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya (Negvevitsky, M. et al. 2002). Perbandingan logika fuzzy dengan Boolean ditunjukkan pada Gambar 2 dibawah ini.



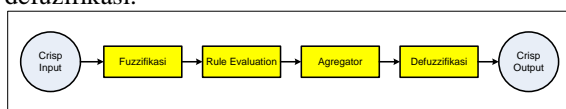
Gambar 2. Perbandingan Boolean dan Fuzzy
Sumber : (Negvevitsky, M. et al. 2002)

Beberapa alasan digunakannya fuzzy logic adalah:

1. Konsep fuzzy logic mudah dimengerti.
2. Fuzzy logic sangat fleksibel.
3. Fuzzy logic memiliki toleransi terhadap data yang kurang tepat.
4. Fuzzy logic mampu memodelkan fungsi nonlinier yang kompleks.
5. Fuzzy logic didasari pada ahasa alami.

2.5.1 Sistem Kendali Fuzzy.

Sistem kendali logika fuzzy yang terdiri atas empat komponen utama yang menyusun proses kendali logika fuzzy, yaitu proses fuzifikasi, basis pengetahuan, logika pengambil keputusan, dan defuzifikasi.

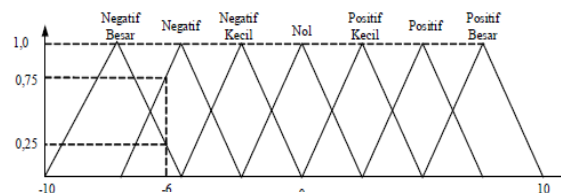


Gambar 3. Tahapan pada Fuzzy Logic Controller

Sumber : (Pambudi, W. S. et al. 2014).

2.5.1.1 Fuzzifikasi

Komponen fuzzifikasi berfungsi untuk memetakan masukan data tegas ke dalam himpunan fuzzy menjadi nilai fuzzy dari beberapa variabel linguistik masukan. Gambar 4 menunjukkan suatu contoh proses fuzzifikasi dimana nilai masukan tegas sebesar -6 dipetakan ke dalam fungsi keanggotaan segitiga dengan label (umumnya berupa variabel linguistik fuzzy) sebagai berikut: Negatif Besar, Negatif, Negatif Kecil, Nol, Positif Kecil, Positif dan Positif Besar. Variabel yang digunakan sebagai contoh adalah besar perubahan pengukuran data dengan jangkauan antara -10 sampai 10. Hasil proses fuzzifikasi dari data -6 tersebut adalah derajat keanggotaan 0,25 untuk Negatif Besar, derajat keanggotaan 0,75 untuk Negatif serta derajat keanggotaan 0 untuk masing-masing Negatif Kecil, Nol, Positif Kecil, Positif dan Positif Besar (Setiawan, I. et al. 2011)



Gambar 4. Proses Fuzzifikasi

Sumber : (Setiawan, I. et al. 2011)

Proses fuzzifikasi diekspresikan seperti Persamaan (18) berikut ini (Fauzi, I. et al. 2014).

$$x = \text{fuzzifier}(x_0) \dots\dots\dots (18)$$

dengan:

x_0 = nilai crisp variabel masukan.

x = himpunan fuzzy variabel yang terdefinisi.

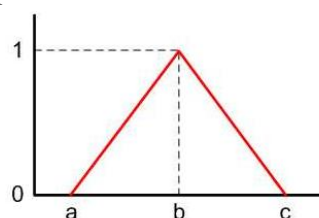
fuzzifier = operator fuzzifikasi yang memetakan himpunan crisp ke himpunan fuzzy.

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Jun Yan, menggunakan:

1. Himpunan fuzzy dengan distribusi simetris.
2. Gunakan himpunan fuzzy dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (rules).
3. Mengatur himpunan fuzzy agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

Salah satu metode yang paling lazim digunakan untuk menentukan fungsi keanggotaan adalah menggunakan bentuk segitiga, yang mengikuti fungsi pada Persamaan (19) dibawah ini.

$$T_{(u,a,b,c)} = \begin{cases} 0 & \text{for } u < a \\ \frac{u-a}{b-a} & \text{for } a \leq u \leq b \\ \frac{c-u}{c-b} & \text{for } b \leq u \leq c \\ 0 & \text{for } u > c \end{cases} \dots\dots\dots (19)$$



Gambar 5. Keanggotaan Segitiga

2.5.1.2 Basis pengetahuan (Rule)

Setelah menentukan *membership function* dan *fuzzy-set* untuk input maka, langkah selanjutnya membuat *rule evaluation*. Bentuk utama *rule evaluation* pada kendali fuzzy ini adalah sebagai berikut :

IF error is a0n AND Δerror a1n THEN Speed is c0

2.5.1.3 Defuzzifikasi

Berdasarkan basis aturan yang telah dibuat, langkah terakhir dari desain fuzzy ini adalah

defuzzifikasi. Defuzzifikasi adalah proses pengubahan besaran Fuzzy yang disajikan dalam himpunan fuzzy ke sinyal yang bersifat bukan fuzzy, dimana hasil output fuzzy akan diolah menjadi data output. Proses defuzzifikasi ini bisa menggunakan metode centroid atau *Center of Gravity* (COG) seperti yang ditunjukkan dalam Persamaan (20) dibawah ini.

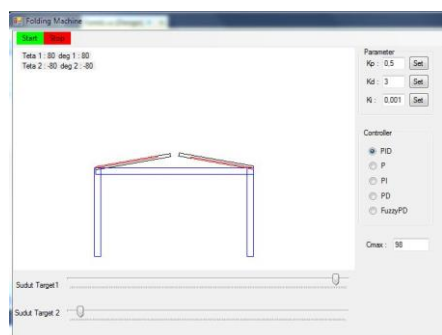
$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \dots\dots\dots (20)$$

dengan:

- U = Keluaran
- wi = Bobot nilai fuzzy
- ui = Nilai linguistik pada fungsi keanggotaan keluaran
- n = Banyak derajat keanggotaan

3. PERANCANGAN

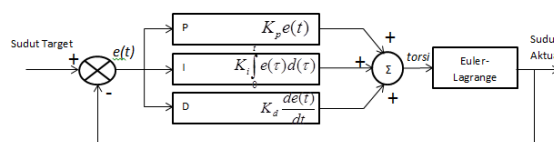
Folding Machine merupakan suatu alat yang berfungsi untuk melipat baju secara semi-automatic. Dalam perancangan sistem kontrolnya dipergunakan simulasi. Simulasi ini dibuat menggunakan program Visual Studio dengan bahasa pemrograman C#. Pada simulasi diimplementasikan 5 macam sistem kontrol dengan tujuan membandingkan sistem kontrol yang tepat untuk plant tersebut. Kelima sistem kontrol tersebut adalah, PID, P, PI, PD dan Fuzzy Logic Controller. Pada Simulasi, Folding Machine dipergunakan 2 arm manipulator 1 DOF sebagai pelipat, satu untuk sisi kanan, dan satu untuk sisi kiri. Dalam simulasi diimplementasikan persamaan dinamik arm manipulator 1 DOF untuk mendapat hasil simulasi yang mendekati keadaan sesungguhnya. Yang dikontrol pada simulasi ini adalah sudut dari manipulator agar mencapai nilai yang diinginkan sehingga dapat dikatakan sebagai control posisi. Berikut tampilan *user interface* dari simulasi yang digunakan.



Gambar 6. Interface simulasi

Berikut diagram blok dan perancangan dari system kontrol yang digunakan.

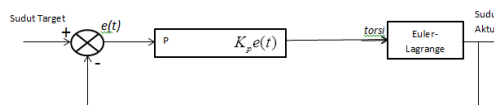
3.1 Kontrol PID



Gambar 7. Diagram Block Control PID

Pada PID controller (Gambar 7), diperlukan informasi nilai error yang didapat dari selisih nilai sudut actual dengan sudut target. Controller ini bertujuan mengurangi nilai error yang ada dengan operasi tertentu. PID adalah penggabungan controller P, I dan D, yang hasil perhitungan dari ketiganya dijumlahnya menjadi satu nilai saja. Controller P merupakan hasil perkalian nilai error dengan Kp yang ditentukan. Controller I merupakan hasil perkalian Ki dengan integral error. D controller merupakan hasil perkalian Kd dengan turunan error. Kp, Ki dan Kd adalah nilai gain yang ditetapkan agar nilai sudut actual sama dengan sudut target. Jika diinterpretasikan dalam waktu, maka P menggunakan error waktu tersebut (present), I menggunakan akumulasi error (past) dan D controller menggunakan prediksi error yang akan datang (future) berdasarkan rasio perubahan untuk mendapatkan nilai sudut actual yang sama dengan sudut target. Keluaran controller pada simulasi ini adalah nilai torsi yang mana nilai ini yang akan dipergunakan dalam persamaan dinamik Euler-Lagrange untuk mendapat nilai sudut actual. Berikut coding untuk PID controller.

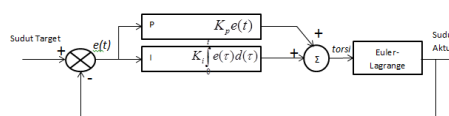
3.2 Kontrol P



Gambar 8. Diagram block control P

P Controller (Gambar 8) hanya menggunakan nilai Present dalam proses mengurangi error yang ada. Untuk coding controller ini, dapat dilakukan modifikasi pada PID dengan menghilangkan I dan D Controller.

3.3 Kontrol PI

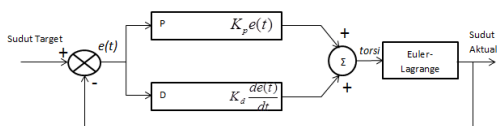


Gambar 9. Diagram block control PI

Sumber : ()

PI Controller (Gambar 9) menggunakan nilai Present dan nilai penjumlahan error sebelumnya. Coding tersebut didapatkan dari modifikasi coding PID dengan menghilangkan D controller.

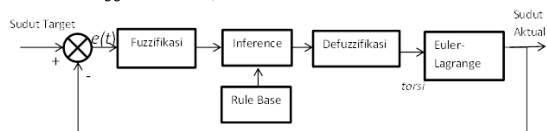
3.4 Kontrol PD



Gambar 10. Diagram block control PD

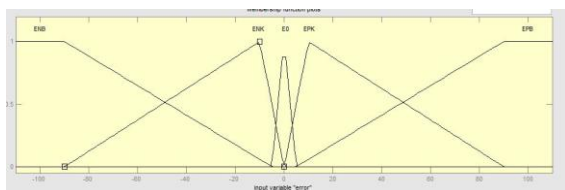
PD controller menggunakan nilai *present* dan prediksi *error* berdasarkan rasio perubahan. Coding untuk controller ini didapatkan dengan memodifikasi PID dengan menghilangkan I Controller.

3.5 Kontrol Fuzzy-PD (Proportional Defferential).

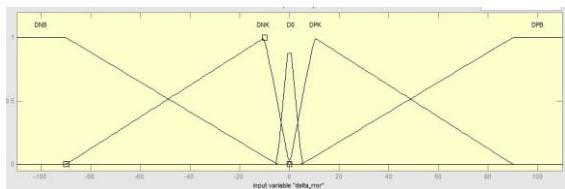


Gambar 11. Diagram Block Fuzzy PD

Fuzzy-PD adalah controller yang menggunakan logika fuzzy, blok system serpti yang ditunjukkan pada Gambar 11. Pada perancangan fuzzy, dipergunakan 2 input yaitu nilai error dan delta error antara *Present Error* dengan *Last Error*. Berikut *membership function* untuk Fuzzy-PD yang dirancang. Nilai error dan delta error ini didapat dari perhitungan *error* dan *delta error* dari PID.



Gambar 12. Membership Function Input Error

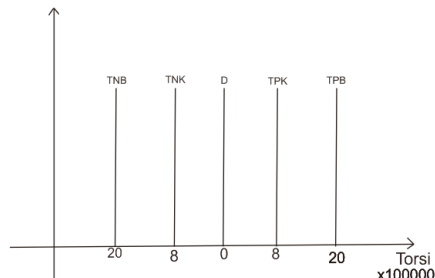


Gambar 13. Membership Function Input Delta Error

Pada fuzzifikasi dipergunakan metode segitiga untuk menentukan nilai fuzzy dari suatu kondisi. Dalam perancangan disusun 5 membership untuk input error dan delta error. Membership input error adalah sebagai berikut: ENB (Error Negatif Besar), ENK (Error Negatif Kecil), E0 (Error=0), EPK (Error Positif Kecil) dan EPB (Error Positif Besar). Sedangkan input delta error, sebagai berikut: DNB (Delta Error Negatif Besar), DNK (Delta Error Negatif Kecil), D0 (Delta Error=0), DPK (Delta Error Positif Kecil) dan DPB (Delta Error Positif Besar).

Besar). Untuk detail setiap membership function dapat diperhatikan pada gambar diatas. Dalam menentukan fuzzy, fuzzifikasi dilakukan dengan metode segitiga.

Untuk *membership function* output beserta *rule base* yang digunakan dapat dilihat pada Gambar 11 dan Tabel 1, berikut ini.



Gambar 14. Membership Function Output

Tabel 1 Rule Base Fuzzy Logic Controller Folding Machine

		error				
		ENB	ENK	E0	EPK	EPB
delta error	DNB	TNK	TPB	TPB	TNB	TPK
	DNK	TNK	TPB	TNK	TNB	TPK
	D0	TNK	TNK	D	TPK	TPK
	DPK	TNK	TPB	TPK	TNB	TPK
	DPB	TNK	TPB	TNB	TNB	TPK

Ket: TNB (Torsi Negatif Besar) = -2000000
 TNK (Torsi Negatif Kecil) = -800000
 D (Diam) = 0
 TPK (Torsi Positif Kecil) = 800000
 TPB (Torsi Positif Besar) = 2000000

4. PENGUJIAN & PENGAMATAN

Pada program simulasi yang dibuat ini, menggunakan dinamika gerak pada arm manipulator, dimana dengan dinamik maka simulasi akan mensimulasikan pergerakan mendekati real. Hal ini, karena dengan konsep dinamik, sudut ditentukan berdasarkan torsi, massa, inersia, gravitasi serta panjang lengan.

Pada simulasi ini, torsi yang menjadi patokan dalam analisa, karena system control yang digunakan akan menghasilkan torsi dan torsi ini yang akan dimasukkan dalam persamaan dinamik untuk menghasilkan besar sudut. Sedangkan besaran lainnya dianggap tetap.

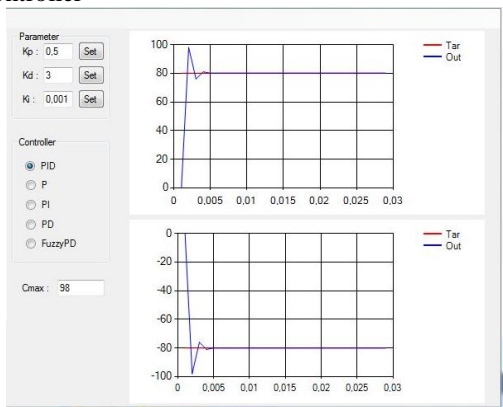
Pada simulasi dibandingkan 5 jenis controller, yaitu PID, P, PI, PD dan Fuzzy-PD. Kelima controller ini menggunakan informasi nilai error yang kemudian menghasilkan nilai torsi agar sudut manipulator dapat mengikuti sudut target. Nilai torsi ini yang akan menentukan pergerakan manipulator, nilai torsi positif untuk gerak rotasi searah jarum jam dan nilai torsi negative untuk gerak totasi berlawanan arah jarum jam.

Untuk pengujian menggunakan PID, P, PI dan PD tidak dilakukan tuning parameter K_p , K_i dan K_d . Untuk itu nilai ketiganya ditentukan sebagai berikut: $K_p = 0,5$; $K_i = 0,001$; $K_d = 3$. Pada simulasi

dipergunakan sudut 80° dengan posisi awal 0° sehingga error awal sebesar 80° .

4.1 PID Controller

Berikut hasil pengujian menggunakan PID Controller



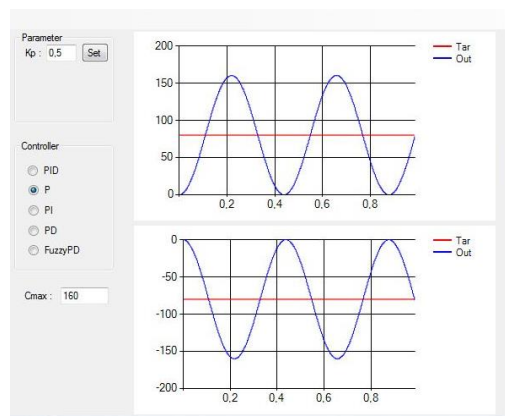
Gambar 15. Hasil Simulasi PID Controller

Dari hasil tersebut dapat diamati pada Gambar 15, pada PID controller dengan nilai $K_p=0,5$; $K_d=3$; $k_i=0,001$; terjadi overshoot sebesar 18° . Hal ini dapat diamati dari nilai maksimum yang dicapai yaitu sebesar 98° dibanding *steady state* sebesar 80° . Namun pada *controller* ini tidak terjadi *error steady state*. Dari grafik tersebut, *rise time* dapat diperkirakan bernilai sekitar $0,002$ s dengan adanya *delay system* sebesar $0,001$ s. Dari grafik dapat diamati bahwa, meskipun terdapat *delay system* sebesar $0,001$ yang diakibatkan bahwa dilakukan perhitungan terhadap dinamik sebelum manipulator bergerak, respon waktu system tetap cepat. System juga mencapai posisi *steady state* sebelum $0,005$ s (*settling time*).

Dari data diatas dapat diamati berdasarkan error tersebut, nilai torsi sangat besar sehingga terjadi overshoot yang mengakibatkan saat gerakan pertama kali, manipulator mencapai sudut 98° . Akan tetapi meskipun terjadi, overshoot system tetap mampu mengembalikan manipulator pada *steady state* dengan menghasilkan torsi yang nilainya berlawanan arah dengan torsi sebelumnya, sehingga manipulator bergerak kembali menuju *steady state* nya. PID menghasilkan respon waktu yang cepat namun terjadi overshoot.

4.2 P Controller

Berikut Hasil Pengujian P Controller, yang ditunjukkan pada Gambar 16 dibawah ini.

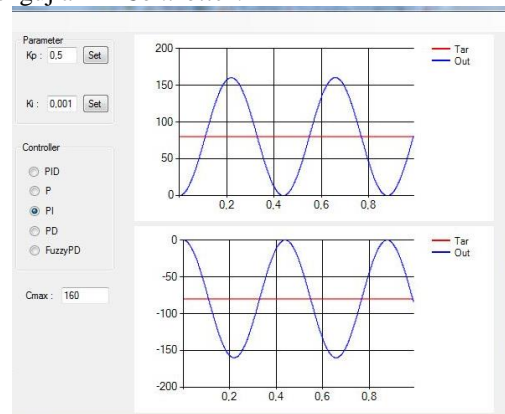


Gambar 16. Hasil Simulasi P Controller

Pada P Controller dengan nilai $K_p=0,5$; system memiliki *rise time* dengan nilai mendekati $0,18$ s dan system beresilasi dengan simpangan sebesar sudut target. System ini tidak mampu mencapai *steady state*.

4.3 PI Controller

Gambar 17 berikut ini merupakan hasil pengujian PI Controller.

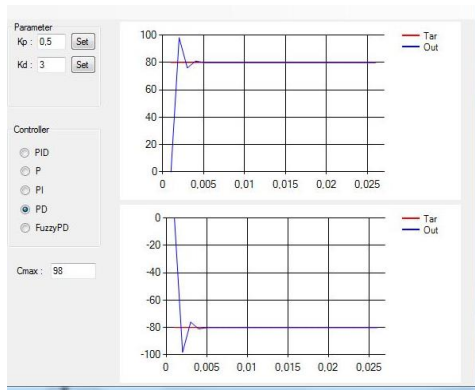


Gambar 17. Hasil Simulasi PI Controller

Simulasi menggunakan PI dengan $K_p=0,5$; $K_i=0,001$; menunjukkan hasil yang sama baik untuk *rise time* maupun respon terhadap target. Pada PI controller, I Controller tampak tidak memberikan pengaruh pada *plant* ini. Hal ini tampak dengan tidak berubahnya *rise time*, yang tetap bernilai sekitar $0,18$ s dan system yang beresilasi dengan simpangan sebesar sudut target dan tidak mampu mencapai *steady state*.

4.4 PD Controller

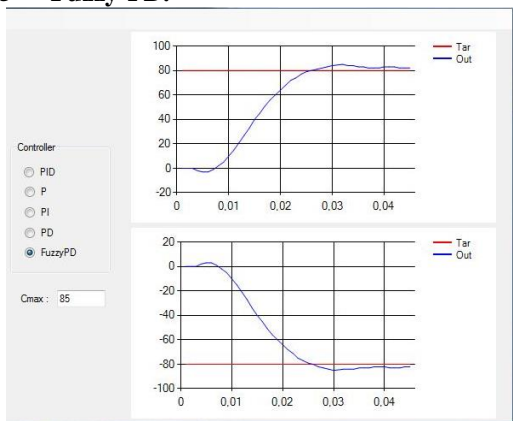
Berikut hasil pengujian dengan PD Controller, yang ditunjukkan pada Gambar 18 dibawah ini.



Gambar 18. Hasil Simulasi PD Controller

Pada PD controller dengan $K_p=0,5$ dan $K_d=3$, respon systemnya menunjukkan angka yang sama dengan PID controller baik untuk *rise time*, *overshoot* dan *error steady state*. Hal ini menunjukkan I Controller tidak berpengaruh sama sekali dalam proses control. Sistem menunjukkan *rise time* sekitar 0,002; *overshoot* 18° dan tidak adanya *error steady state*.

4.5 Fuzzy-PD.



Gambar 19. Hasil Pengujian Fuzzy Logic Controller.

Berdasarkan Gambar 19 hasil pengujian pada Fuzzy-PD dengan *rule* yang telah dibahas sebelumnya, maka dapat diamati system memiliki *rise time* 0,01 s dengan terjadi *delay* sekitar 0,01s. Pada *system* terjadi *Overshoot* sebesar 3° akibat nilai maksimum 85° dan *steady state system* sebesar 82° . Sistem memiliki *error steady state* sebesar 2° . Dari hasil tersebut, tampak bahwa hasil dari FLC belum maksimal dengan respon yang tidak terlalu cepat dan terjadinya *error steady state*. Namun besar *error steady state* ini dapat ditoleransi mengingat nilainya yang sangat kecil dan pada *folding machine* ini tidak diperlukan kepresisian sudut yang tinggi. Tidak maksimalnya Fuzzy-PD ini diakibatkan *rule* yang masih belum cocok dengan *plant* tersebut.

4.6 Perbandingan Respon Kontrol.

Dari data diatas maka dapat dilakukan perbandingan sebagai berikut.

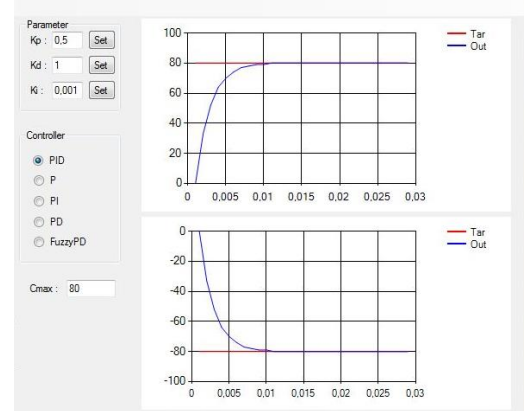
Tabel 2 Perbandingan Respon Control

No	Jenis Controller	Rise Time	Overshoot	Error Steady State	Keterangan
1	PID	0,002 s	18°	0°	
2	P	0,18 s	-	-	Berosilasi
3	PI	0,18 s	-	-	Berosilasi
4	PD	0,002 s	18°	0°	
5	FLC	0,01 s	3°	2°	

Dari Tabel 2 diatas dapat diamati bahwa dalam plant ini I controller tidak memberikan pengaruh. Hal ini tampak dari tidak adanya perbedaan antara P dan PI serta PID dan PD. Tanpa D Controller respon system menjadi semakin lambat dan tidak mencapai *steady state*.

Suatu system dikatakan baik apabila memiliki respon yang cepat namun memiliki sekecil mungkin atau bahkan tidak memiliki *overshoot*, maka dari data diatas dapat diamati bahwa yang memiliki respon terbaik adalah Fuzzy-PD, karena meskipun lebih lambat dari PID, *Overshoot* pada Fuzzy-PD sangat kecil dibanding dengan PID maupun PD. Meskipun memiliki *error steady state*, akibat nilainya yang kecil maka nilai ini dapat ditoleransi mengingat pada *Folding Machine* tidak diperlukan kepresisian yang tinggi. Perbedaan *rise time* diantara keduanya juga tidak memiliki selisih yang terlalu besar.

Namun mengingat Fuzzy-PD belum maksimal dalam perancangan *rule*, maka dilakukan tuning parameter PID. Mengingat respon pada PID mengalami *overshoot* akibat respon system yang terlalu cepat maka dilakukan pengujian dengan menurunkan nilai K_d menjadi 1 sedangkan yang lainnya tetap. Pengujian dilakukan juga akibat pada PID tidak memiliki *error steady state*. Hasil pengujian yang didapatkan seperti pada Gambar 20.



Gambar 20. Hasil pengujian PID dengan Tuning parameter

Dari Gambar 20 diatas, dapat diamati system tidak memiliki *overshoot* dan *error steady state*. Hal ini dikarenakan dengan menurunkan nilai K_d maka system melambat yang ditunjukkan dengan nilai *rise time* yang menurun menjadi 0,005 dari sebelumnya

0,002. Penurunan inilah yang mengakibatkan perbaikan respon *system*, sehingga apabila dibandingkan dengan respon sebelumnya maka didapat data sebagai berikut seperti pada Tabel 3.

Tabel 3. Perbandingan Respon *Control* Setelah Tuning PID

No	Jenis Controller	Rise Time	Overshoot	Error Steady State	Keterangan
1	PID	0,002 s	18°	0°	-
2	P	0,18 s	-	-	Berosilasi
3	PI	0,18 s	-	-	Berosilasi
4	PD	0,002 s	18°	0°	-
5	FLC	0,01 s	3°	2°	-
6	PID (Tuning Parameter)	0,005 s	0°	0°	-

Dari data diatas maka, dapat diamati bahwa *system* yang memiliki respon terbaik adalah PID *controller* dengan hasil Tuning PID ($K_p=0,5; k_i=0,001; k_d=1$). Hal ini tampak dengan respon waktu yang lebih cepat dibanding Fuzzy-PD namun tidak memiliki *overshoot* maupun *error steady state*.

5. KESIMPULAN

Konsep gerak dinamik pada *arm manipulator* 1 DOF yang digunakan pada simulasi *folding machine* memberikan gambaran secara riil terhadap gerak *manipulator* karena pada perhitungannya nilai torsi, massa, panjang lengan, gravitasi dan inersia dilibatkan dalam menentukan sudut.

Pada pengujian *system control* yang digunakan pada *folding machine*, tanpa dilakukan tuning PID dengan nilai $K_p = 0,5; K_i=0,001; K_d=3$, maka didapat PID dan PD memiliki respon waktu yang paling cepat dengan *rise time* 0,002 namun terjadi *overshoot* sebesar 18°. Berbeda dengan P dan PI yang memiliki *rise time* 0,18 s namun berosilasi dan tidak mencapai *steady state*. *I controller* tidak memberikan pengaruh. Hal ini tampak dari tidak adanya perbedaan antara P dan PI serta PID dan PD. Tanpa *D Controller* respon *system* menjadi semakin lambat dan tidak mencapai *steady state*.

Respon terbaik jika tidak dilakukan tuning parameter PID adalah dengan Fuzzy-PD karena meskipun lebih lambat dari PID, *overshoot* pada Fuzzy-PD sangat kecil dibanding dengan PID maupun PD meskipun memiliki *error steady state*, karena nilai *error* ini cukup kecil maka nilai ini dapat ditoleransi mengingat pada *folding machine* tidak diperlukan kepresisian yang tinggi. Perbedaan *rise time* diantara keduanya juga tidak memiliki selisih yang terlalu besar. Fuzzy-PD memiliki *rise time* 0,01 s dengan *overshoot* 3° dan *error steady state* 2°.

Dengan Tuning PID dengan menurunkan nilai D menjadi 1, maka *system* inilah yang yang

memiliki respon terbaik karena memiliki *rise time* yang lebih cepat yaitu 0,005 dibanding Fuzzy-PD namun tidak memiliki *error steady state* dan *overshoot*.

6. SARAN

Pada simulasi ini hanya dilakukan secara pendekatan riil dari sistem, sehingga apabila ingin mengetahui secara riil perlu implementasi lebih lanjut. Apa yang didapat dari penelitian ini bisa digunakan sebagai dasar untuk menentukan parameter dan jenis control yang sesuai untuk diimplementasikan dalam *plant folding machine*.

7. DAFTAR PUSTAKA

- David I. Robles. 2012, *PID control dynamis of a robotic arm manipulator with two degree of freedom*.
- Fauzi, Imam., Y,Erni.,Purwanto.2014.*Pengaturan kecepatan Motor dalam Mempertahankan Batas Tepi Badan Robot Line Follower terhadap Line Menggunakan Fuzzy Logic Controller*. Universitas Brawijaya
- Hanafi, Dirman., Abueejela, YousefMoh., Zakaria, Mohammad Fauzi. .2013.*Wall Followr Autonomous Robot Development Applying Fuzzy Incremental Controller*, Intelligent Control and Automation ,2013,Vol. 4 No. 1,pp. 18-25.
- Melchiorri, Claudio. “*Dynamic model of robot manipulator*”, Dipartimento di Ingegneria dell’Energia Elettrica e dell’Informazione (DEI), Universita di Bologna.
- Negvevitsky, M. (2002), *Artificial Intelligence: A Guide to Intelligent System*. Sydney : Addison Wesley.
- Pambudi W. S., Sumanang. N. M. A., (2014), “Implementasi Fuzzy-PD untuk Menentukan Posisi Obyek pada Model Simulasi Robot Arm Manipulator 3 DOF (Degree of Freedom)dalam Bidang 2 Dimensi”, Jurnal Mikrotek, Universitas Trunojoyo Madura, Vol. xx, No. xx, ISSN : 2338-9460.
- Robot Dynamics and Control.(Online). (http://www.cds.caltech.edu/~murray/books/MLS/pdf/mls94-manipdyn_v1_2.pdf, diakses 20 Desember 2014).
- Setiawan,Iwan.,Darjat.,Septiaji,Ashadi.2011.*Perancangan Robot Mobil Penjejakl Dinding Koridor Menggunakan Kendali Logika Fuzzy*. Universitas Diponegoro, Semarang.