

Sintesis Kendali PID *Digital* dengan Diskritisasi Langsung dan *Backward Difference*

FENI ISDARYANI, MOHAMAD FADHILAH VIERI HESYA, FERIYONIKA

Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung, Indonesia
Email : feni.isdaryani@polban.ac.id

Received 29 Januari 2021 | Revised 22 Februari 2021 | Accepted 4 Maret 2021

ABSTRAK

Kendali PID analog, yang realisasinya menggunakan komponen elektronika, memiliki keterbatasan yaitu nilai toleransi yang terbatas. Saat ini spesifikasi kontroler dituntut untuk dapat berkomunikasi dengan sistem yang lebih besar seperti SCADA dan DCS sehingga lebih cocok menggunakan pengendali digital. Penelitian ini menganalisis metode konversi PID analog ke digital agar dihasilkan difference equation yang dapat direalisasikan kedalam pemrograman komputer. Metode yang dipakai adalah diskritisasi langsung dan Backward Difference. Perbandingan kedua metode dilakukan dengan menganalisis respon berdasarkan initial parameter yang dihasilkan oleh metode Ziegler Nichols. Hasil pengujian menunjukkan kendali PID diskrit menggunakan Backward Difference menghasilkan respon sistem yang lebih baik dibandingkan metode diskritisasi langsung dengan nilai K_p , T_i , dan T_d adalah 50, 80 dan 0,001 menghasilkan respon dengan nilai rise time, settling time dan overshoot berturut-turut sebesar 33,66s, 90,39s dan 0,9%.

Kata kunci: PID diskrit, diskritisasi langsung, Backward Difference, Ziegler Nichols

ABSTRACT

The analog PID control, where its parameters are realised using the electronic component, has disadvantages due to the limitation of its tolerance value. Currently, the specifications of controller are required to be able to communicate with larger systems such as SCADA and DCS, therefore digital controller is more appropriate to use. This study analyzes the analog to digital PID conversion method to generate a difference equation that can be realized in computer programming. The direct discretization and Backward Difference method are used. Comparison of both methods is by analyzing response based on initial parameters obtained of Ziegler Nichols method. The results show that discrete PID control using the Backward Difference indicates a better response than using the direct discretization method with K_p , T_i , and T_d values are 50, 80, and 0,001, respectively. Those parameters generate response with rise time, settling time, and overshoot values of 33,66s, 90,39s, and 0,9%, respectively.

Keywords: discrete PID, direct discretization, Backward Difference, Ziegler Nichols

1. PENDAHULUAN

Kontroler PID merupakan algoritma kendali yang banyak diaplikasikan pada peralatan industri maupun peralatan yang didesain khusus untuk keperluan tertentu. Fitur yang penting pada kontroler ini adalah tidak memerlukan analisis model yang presisi dari sistem yang akan dikontrol. Dimana pemodelan dapat dilakukan dengan istilah yang sering dipakai yaitu metode *black box*. Oleh karena itu kontroler PID telah banyak digunakan dalam bidang robotika, automasi, kontrol proses, manufaktur, transportasi. Dalam menyelesaikan permasalahan kontrol, sangat penting untuk mengerti apa tujuan utama dari kontrol tersebut. Pada umumnya terdapat dua permasalahan kontrol, yaitu untuk mengikuti *setpoint* dan untuk menolak gangguan. Kendali PID analog yang terdiri dari rangkaian komponen elektronika, memiliki kekurangan yaitu keterbatasan nilai komponen elektronika untuk merealisasikan parameter pengendali itu sendiri. Saat ini, pengendali dituntut untuk dapat berkomunikasi dengan sistem yang lebih besar misal seperti SCADA dan DCS sehingga lebih cocok jika menggunakan pengendali *digital*.

Pada tahun 2018 penelitian yang dilakukan Podrżaj (**Podrżaj, 2018**) menjelaskan dasar pengendali PID analog dan diskrit namun tidak menjelaskan bagaimana merealisasikan algoritma PID kedalam *embedded system*. Formula untuk menemukan *gain proportional*, integral dan *derivative* menggunakan PID diskrit dijabarkan oleh Narkvitul dan timnya di tahun 2014 (**Narkvitul, et al., 2014**) dimana diskritisasi dilakukan menggunakan pendekatan trapezoidal. Hasil simulasi menunjukkan bahwa pengendali PID waktu diskrit menghasilkan respon yang lebih cepat dibandingkan respon PID waktu kontinyu. Pada tahun 2019, penelitian yang dilakukan Angle (**Angel, et al., 2019**) dan timnya menghasilkan pengendali PID diskrit yang dibangun menggunakan metode *pole placement* untuk menghasilkan anti *windup controller* pada motor DC. Adapun proses diskritisasi PID menggunakan metode *Backward Difference* hasilnya menunjukkan performansi yang baik untuk perubahan *setpoint* dari sistem begitupula saat perubahan sinyal *ramp*. Pada tahun 2020, Das (**Das, et al., 2020**) dan timnya menyimpulkan bahwa pengendali PID juga dapat digunakan untuk mengukur respon *delay* dari sistem yang sering terjadi pada teknik *pole placement*. Peneliti Almanda dan timnya pada tahun 2016 (**Almanda, et al., 2016**), melakukan penelitian pengendali ketinggian air menggunakan PID adaptif, hasilnya menunjukkan bahwa sistem mampu beradaptasi dengan *setpoint* yang berubah ubah, namun pada penelitian ini sebetulnya dapat dilakukan reduksi perhitungan analitikal dengan menggunakan metode yang tidak memerlukan perhitungan yang rumit. Pada tahun 2020, Babu (**Babu, et al., 2020**) dan timnya, melakukan penelitian yang berfokus pengendalian *water level* pada tangki penyimpanan menggunakan kendali PID, hasil penelitian hanya dilakukan berdasarkan simulasi tidak dijelaskan realisasi/implementasi pada *plant* sesungguhnya.

Terdapat permasalahan lain pada pengendali PID adalah bagaimana mendapatkan parameter kendali PID yang terdiri dari *gain proportional* (K_p), integral (K_i) dan *derivative* (K_d). Salah satu caranya adalah menggunakan metode *tuning* parameter. Berbagai metode *tuning* PID seperti *response-based design*, *internal model control* dan optimisasi kontroler dengan algoritma stokastik telah banyak dibahas dalam penelitian. Penelitian yang dilakukan oleh Chen dan timnya, pada tahun 2017 (**Chen, et al., 2017**), yaitu membangun metode *extremum seeking* (SE) untuk *tuning* parameter PID secara *on-line* untuk meningkatkan performansi *steady state* pada *plant magnetic levitation*. Pada tahun 2018, Kumar dan Pan (**Kumar & Pan, 2018**), mencoba untuk merancang pengendali PID berdasarkan spesifikasi kestabilan margin yang dapat meningkatkan respon transient dari sistem. Perhitungan matematika dan analisis dilakukan menggunakan respon frekuensi, hasil penelitian menunjukkan respon sesuai dengan spesifikasi yang diinginkan. Sedangkan, penelitian yang dilakukan Verma dan Kumar pada

tahun 2019 (**Verma & Kumar, 2019**), berfokus pada *on-line tuning* parameter menggunakan aturan *tuning* yang optimal sehingga menghasilkan parameter PID yang *robust*. Ada juga penelitian yang dilakukan oleh Chopade dan timnya pada tahun 2016 (**Chopade, et al., 2016**), menggunakan metode pendekatan optimasi *pole-zero*, dimana nilai *pole-zero* diperoleh menggunakan pendekatan Tustin. Dalam praktiknya, metode ini cocok digunakan untuk *plant* yang nonlinier dan tidak stabil. Penelitian yang dilakukan tahun 2020 oleh Isdaryani dan timnya (**Isdaryani, et al., 2020**), membandingkan dua metode *response-based tuning* parameter PID, yaitu Ziegler-Nichols dan Cohen-Coon telah dilakukan. Hasil pengujian menunjukkan bahwa *tuning* PID menggunakan metode Ziegler Nichols menghasilkan respon lebih baik dibandingkan dengan metode Cohen-Coon untuk *plant magnetic levitation*.

Performansi pengendali PID tergantung pada saat perancangan baik menggunakan simulasi ataupun konfigurasi *digital* secara langsung. Dalam banyak kasus, faktor paling penting dalam menghasilkan performansi dari kontroler PID diskrit atau kontroler waktu diskrit lainnya adalah waktu cuplik (*sampling time*) dan teknik numerik integrasi. Terdapat beberapa pendekatan untuk mengkonversi kontroler PID waktu kontinyu (analog) menjadi kontroler PID waktu diskrit. Pendekatan tersebut seperti *forward rule*, *backward rule*, *trapezoidal rule*, *bilinear approximation*. Penelitian yang dilakukan oleh El-sharif dan tim pada tahun 2014 (**El-sharif, et al., 2014**), hanya menggunakan metode transformasi Z untuk melakukan diskritisasi PID, namun pada penelitian ini dijelaskan bagaimana merealisasikan persamaan PID ke dalam perangkat *digital* dengan baik. Pada tahun 2013, peneliti Coughi dan Ntogramatzidis (**Coughi & Ntogramatzidis, 2013**), menyimpulkan bahwa teknik sintesis PID diskrit yang dibangun dapat mengurangi prosedur *trial and error* dalam respon frekuensi desain namun memerlukan teknik perhitungan yang cukup rumit karena menggunakan respon frekuensi.

Penelitian ini menganalisis metode konversi PID analog ke *digital* agar dihasilkan *difference equation* yang bentuknya dapat direalisasikan kedalam pemrograman komputer. Metode yang dipakai adalah diskritisasi langsung (*direct discretization*) berdasarkan aturan *discrete approximation* dan metode *Backward Difference* dimana bagian integral dari persamaan PID terlebih dahulu dihilangkan. Adapun, parameter kendali PID (K_p , T_i , T_d) diperoleh menggunakan metode Ziegler Nichols tipe 1 (ZN1) dan tipe 2 (ZN2). Selanjutnya kontroler PID yang telah didesain diaplikasikan pada *plant level air*.

2. METODOLOGI PENELITIAN

2.1 PID Control

Persamaan (1) adalah bentuk umum dari kendali PID,

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (1)$$

dimana K_p adalah *gain proportional*, T_i adalah integral waktu, T_d adalah diferensial waktu, $e(t)$ adalah sinyal *error* dan $u(t)$ adalah sinyal kontrol.

Berdasarkan Persamaan (1) terlihat bahwa aksi integral berfungsi untuk meningkatkan *gain* dengan frekuensi rendah sehingga dapat mengurangi *error steady state*. Aksi *derivative* membuat sistem memiliki karakteristik *phase lead*, yang mana dapat meningkatkan kestabilan dan *bandwidth* sistem.

2.2 Kendali PID dengan *Backward Difference*

Dalam kendali PID, model *Backward Difference* merupakan salah satu jenis pengontrol PID yang menghilangkan bagian integral pada persamaannya. Dalam hal ini titik yang akan dijadikan objek dalam perhitungannya yaitu nilai *error*.

Adapun persamaan umum untuk mengubah persamaan domain waktu kontinyu menjadi domain waktu diskrit ditunjukkan oleh Persamaan (2).

$$\begin{aligned}\frac{dy(t)}{dt} &= \frac{y(k) - y(k-1)}{T}; \\ \frac{d^2y(t)}{dt^2} &= \frac{y(k) - 2y(k-1) + y(k-2)}{T^2}\end{aligned}\quad (2)$$

Untuk menghilangkan bagian integral pada Persamaan (1) dilakukan proses diferensial sehingga menjadi Persamaan (3)

$$\frac{du(t)}{dt} = K_p \frac{de(t)}{dt} + \frac{K_p}{T_i} e(t) + K_p T_d \frac{d^2e(t)}{dt^2} \quad (3)$$

kemudian, substitusi Persamaan (2) kedalam Persamaan (3), maka didapat Persamaan (4)

$$\frac{u(k) - u(k-1)}{T} = K_p \frac{e(k) - e(k-1)}{T} + \frac{K_p}{T_i} e(k) + K_p T_d \left(\frac{e(k) - 2e(k-1) + e(k-2)}{T^2} \right) \quad (4)$$

dengan

$K_i = \frac{K_p}{T_i}$; dan $K_d = K_p T_d$ dan T adalah *time sampling* dari Analog ke *Digital* (A/D).

Selanjutnya dilakukan perhitungan untuk mendapatkan persamaan sinyal kontrol ($u(k)$)

$$\begin{aligned}u(k) - u(k-1) &= K_p(e(k) - e(k-1)) + K_i T e(k) + K_d \left(\frac{e(k) - 2e(k-1) + e(k-2)}{T} \right) \\ u(k) &= u(k-1) + K_p(e(k) - e(k-1)) + K_i T e(k) \\ &\quad + K_d \left(\frac{e(k) - 2e(k-1) + e(k-2)}{T} \right)\end{aligned}\quad (5)$$

Persamaan (5) dapat disederhanakan menjadi Persamaan (6).

$$u(k) = u(k-1) + (K_p + K_i T + K_d/T)e(k) - \left(K_p + \frac{2K_d}{T} \right) e(k-1) + \frac{K_d}{T} e(k-2) \quad (6)$$

sehingga didapat bentuk sederhana dari sinyal kontrol ($u(k)$) pada Persamaan (7)

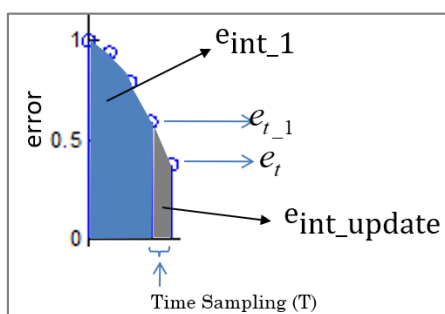
$$u(k) = u(k-1) + K_3 e(k) - K_4 e(k-1) + K_2 e(k-2) \quad (7)$$

dengan nilai,

$$K_1 = \frac{K_p T}{T_i}; \quad K_2 = \frac{K_p T_d}{T}; \quad K_3 = (K_p + K_2 + K_1); \quad K_4 = (K_p + 2K_2).$$

2.3 Kendali PID dengan Diskritisasi Langsung

Untuk mengimplementasikan pengendali PID menggunakan komputer *digital*, perlu mengubah persamaan kontinyu (1) ke persamaan *difference* dengan metode persamaan diskritisasi langsung. Bagian integral didefinisikan sebagai jumlah luasan total yang dibentuk oleh *error* dari awal mulai sampai *error* terakhir, seperti yang deskripsikan pada Gambar 1. Berdasarkan Persamaan (8), penjumlahan luasan *error* didefinisikan sebagai luas sebelum adanya e_t , biasa disebut e_{int_1} , dijumlahkan dengan luas area baru yang dibentuk oleh e_t dan e_{t-1} (disebut e_{int_update}). Penjumlahan luasan area baru ini menggunakan rumus trapesium dimana kedua sisi sejajar adalah e_t dan e_{t-1} dan tingginya adalah T .



Gambar 1. Deskripsi perhitungan integral *error*

$$\int_0^t e(t)dt \approx e_{int_1} + \left(\frac{(e_t + e_{t-1})T}{2} \right) \quad (8)$$

Untuk bagian *derivative* didefinisikan sebagai perubahan *error* terhadap perubahan waktu seperti yang ditunjukkan oleh Persamaan (8).

$$\frac{de(t)}{dt} \approx \frac{e(nT) - e(nT - T)}{T} \quad (9)$$

substitusi Persamaan (8) dan (9) diatas ke Persamaan (1) maka didapat

$$u(nT) = K_p \left(e(nT) + \frac{1}{T_i} \left(e_{int_1} + \left(\frac{(e_t + e_{t-1})T}{2} \right) \right) \right) + T_d \frac{e_n - e_{n-1}}{T} \quad (10)$$

pada Persamaan (10) untuk menghitung *output* pengendali u_n maka membutuhkan nilai *error* awal ($t = 0$) sampai n (waktu perhitungan *error* terakhir). Adapun perhitungan integral *error* untuk realisasi pada *embedded system* ditunjukkan pada Persamaan (11).

$$e_{int} = e_{int_1} + e_{int_update} \quad (11)$$

dengan,

e_t	=	<i>error</i> saat ini;	e_{int_1}	=	Luas area <i>error</i> sebelum perhitungan <i>error</i> terbaru e_t
e_{t-1}	=	<i>error</i> sebelum;	e_{int_update}	=	Luas area yg dibentuk <i>error</i> saat ini dan <i>error</i> sebelumnya
e_{int}	=	<i>error</i> integral;	T	=	<i>time sampling</i>

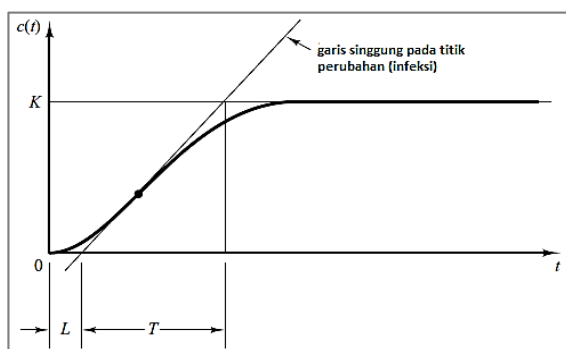
2.4 Metode Ziegler-Nichols Tipe 1

Metode Ziegler-Nichols tipe 1 (ZN1) dikenal juga sebagai metode penalaan lingkaran terbuka dimana pengukuran sistem yang akan dikendalikan dilakukan dengan keadaan tanpa umpan balik. Respon ini akan dibandingkan dengan *setpoint* sehingga *error* (perbedaan dengan *setpoint*) dari respon akan terlihat. Aturan *tuning* ZN1 ditujukan untuk menyatakan nilai *gain proportional* K_p , waktu integral T_i dan waktu *derivative* T_d berdasarkan karakteristik respon *transient* yang diberikan oleh *plant*.

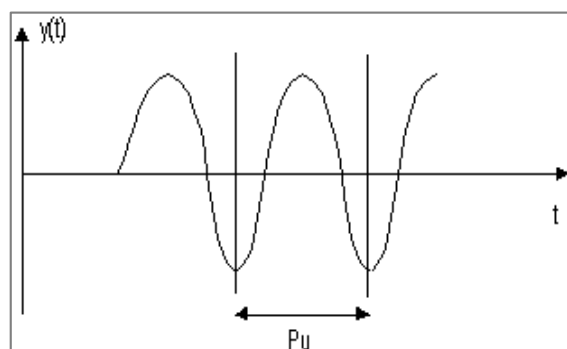
Parameter pengendali PID di dapat menggunakan metode ZN1, dimana *plant* diberi *input* kemudian *output*-nya di analisis untuk mendapatkan nilai parameter *dead time* (T) dan *time constant* (L). Nilai L merupakan waktu dari mulainya *setpoint* diaktifkan sampai dengan nilai sumbu x dari titik potong antara garis vertikal pertama dengan titik nilai tegangan awal sebelum *setpoint* aktif. Sementara nilai T merupakan nilai sumbu x titik perpotongan antara garis vertikal kedua dengan nilai posisi mantap (*steady state*) dari respon dikurangi dengan nilai x dari titik perpotongan antara garis vertikal pertama dengan titik awal seperti yang terlihat pada Gambar 2. Adapun tabel aturan *tuning* dari ZN1 diperoleh dari publikasi yang dilakukan oleh Patel, tahun 2020 (Patel, 2020) dan dapat dilihat pada Tabel 1.

Tabel 1. Aturan *Tuning* Kendali PID menggunakan Ziegler Nichols Tipe 1.

Jenis Kontroler	K_p	T_i	T_d
P	T/L	∞	0
PI	$0,9T/L$	$L/0,3$	0
PID	$1,2T/L$	$2L$	$0,5L$



Gambar 2. Ilustrasi Menentukan *Time Constant* (L) dan *Dead Time* (T) dari Respon Sistem Lup Terbuka



Gambar 3. Ilustrasi Respon Berosilasi dengan Simpangan Tetap

2.5 Metode Ziegler-Nichols Tipe 2

Pada Metode Ziegler Nichols Tipe 2 (ZN2), identifikasi parameter dilakukan menggunakan respon dari sistem lup tertutup (menggunakan umpan balik) serta penguat tegangan *input* sebelum masuk ke aktuator. Pada penelitian ini, metode ZN2 digunakan untuk mencari parameter kendali K_p , K_i , dan K_d dari *plant level* air. Dalam praktiknya, akan dicari nilai penguatan *proportional* (K_p) sampai sistem berosilasi dengan simpangan tetap (*sustain oscillation*) dengan parameter K_i dan K_d diatur agar bernilai nol. Kondisi *sustain oscillation* dilustrasikan pada Gambar 3. Parameter *proportional* (K_p) akan dinaikkan dimulai dari nol sampai mencapai nilai yang mengakibatkan sistem dalam kondisi *sustain oscillation*. Besarnya

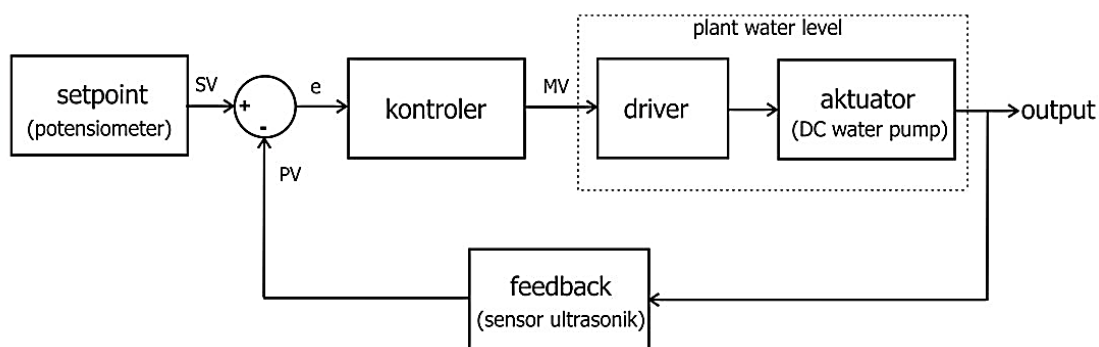
gain proportional saat sistem mencapai kondisi beresilasi dengan simpangan tetap disebut *ultimate gain* (K_u), sedangkan periodenya disebut *ultimate period* (P_u). Adapun tabel aturan *tuning* untuk ZN2 sesuai dengan yang ditulis oleh Chia pada tahun 2018 (Chia, 2018) dapat dilihat pada Tabel 2.

Tabel 2. Aturan *Tuning* Kendali PID Ziegler Nichols Tipe 2

Parameter Kendali	Aturan
K_p	$0,6 \times K_u$
K_i	$K_p / (0,5 \times P_u)$
K_d	$0,125 \times K_p \times P_u$
T_i	$0,5 \times P_u$
T_d	$0,125 \times P_u$

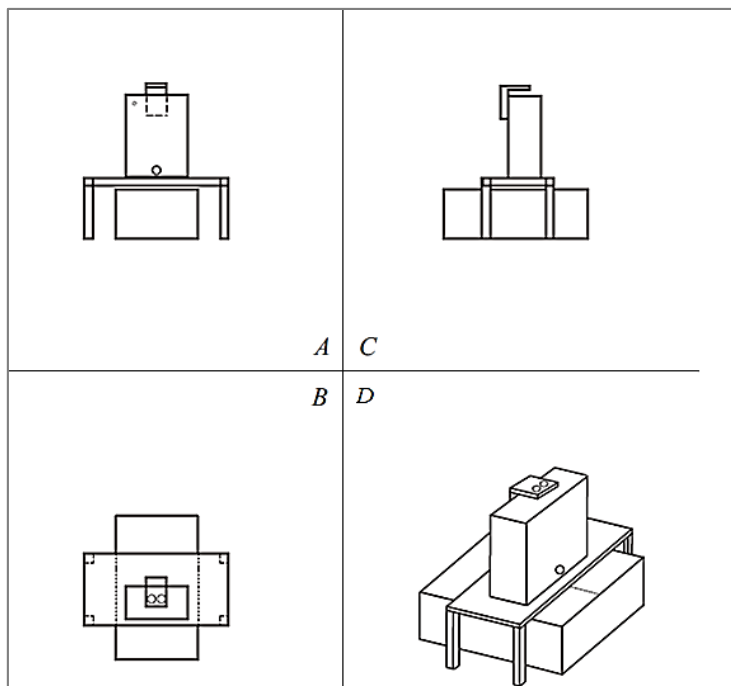
2.6 Perancangan Sistem Kendali *Level* Air

Pada penelitian ini, dibuat sebuah sistem pengendali *level* air menggunakan metode PID dengan teknik diskritisasi langsung dan teknik *Backward Difference*. Diagram blok dari sistem ditunjukkan oleh Gambar 4. Sistem menggunakan struktur lup tertutup dimana terdiri dari blok *setpoint value* (SV), kontroler dengan sinyal *output* dinotasikan oleh *manipulated value* (MV), blok *plant* terdiri dari *driver* dan aktuator yang berfungsi memompa air kedalam tangki. Ketika *setpoint* berupa potensiometer di tetapkan, maka pompa air akan membuat *level* air sesuai dengan nilai *setpoint*. Pompa akan mempertahankan keadaan *level* air ketika mencapai nilai sensor ultrasonik sebagai *feedback* dari sistem ini berfungsi untuk mengukur ketinggian/ *level* air dalam tangki. Sinyal *output* dari sensor ultrasonik dinotasikan oleh *process variable* (PV).

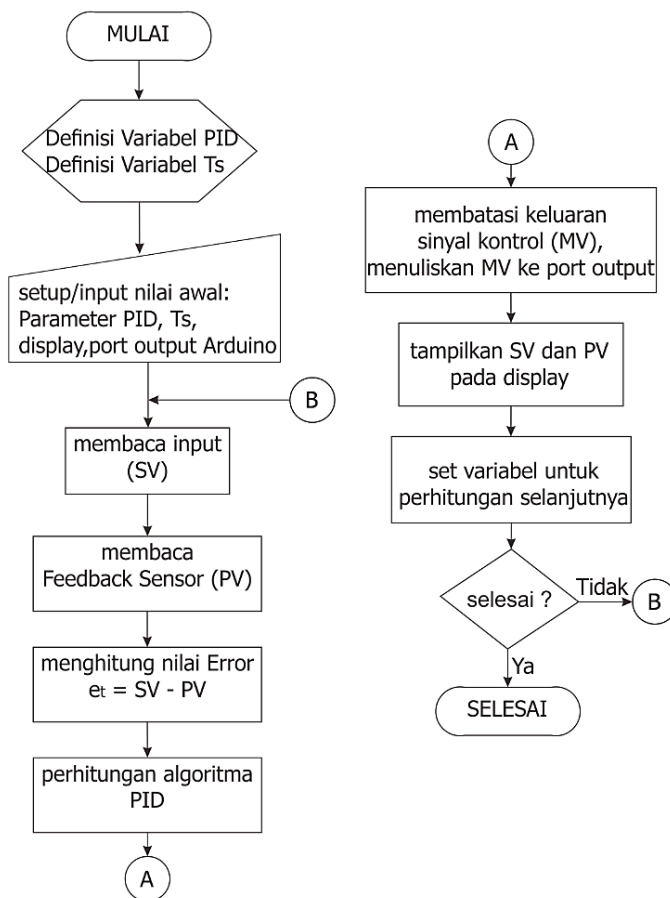


Gambar 4. Diagram Blok Sistem

Plant level air terdiri dari dua buah tangki yaitu terdiri dari tangki utama (yang digunakan untuk pengukuran *level* air) dan tangki penampungan air. Terdapat keran air pada tangki utama yang berfungsi sebagai gangguan pada *level* air. Adapun desain *plant level* air dapat dilihat pada Gambar 5, dimana A merupakan tampak depan, B tampak atas, C tampak samping dan D bentuk 3D dari *plant*. Perancangan diagram alir pengendali untuk *plant level* air menggunakan metode PID ditunjukkan pada Gambar 6. Untuk implementasi metode *Backward Difference* maka substitusi Persamaan (7) kedalam blok perhitungan algoritma PID pada diagram alir sedangkan untuk metode diskritisasi langsung substitusi Persamaan (10). Untuk menghitung integral *error* substitusi dari Persamaan (11).



Gambar 5. Desain *plant level* air. Bagian A merupakan Tampak Depan, B Tampak Atas, C Tampak Samping dan D Bentuk 3D dari *Plant*



Gambar 6. Diagram Alir Kendali *Plant Level* Air menggunakan PID

3. HASIL DAN PEMBAHASAN

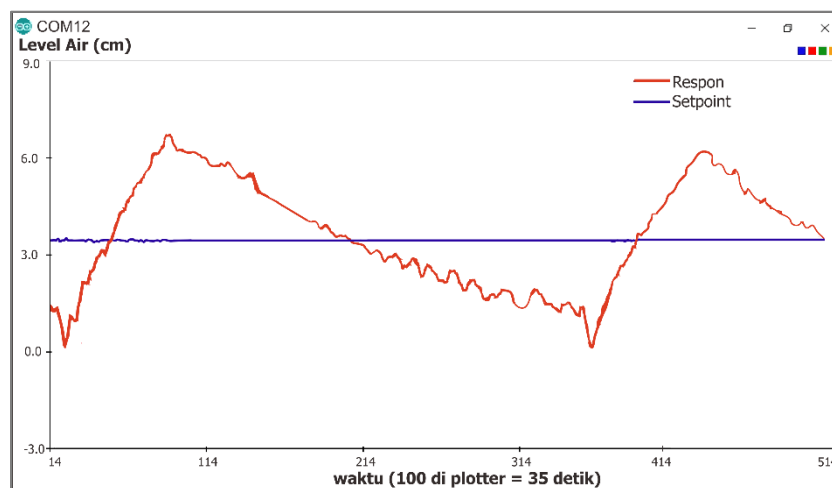
Kedua metode diskritisasi tersebut dibandingkan berdasarkan parameter awal yang didapat dari proses desain kendali menggunakan Ziegler Nichols tipe 1 (*open loop*) dan tipe 2 (*closed loop*). Selain itu, dilakukan pengujian terhadap perubahan *setpoint* yang ditujukan untuk melihat efek dari *differential error* dan juga pengujian terhadap gangguan. Semua pengujian baik metode diskritisasi langsung atau *Backward Difference* menggunakan *time sampling* sebesar 0,01 detik. *Time sampling* (T) adalah waktu yang dibutuhkan untuk mencuplik/mengambil satu data sampai perhitungan atau pengambilan data terbaru.

3.1 Uji Parameter Awal ZN1 dan ZN2

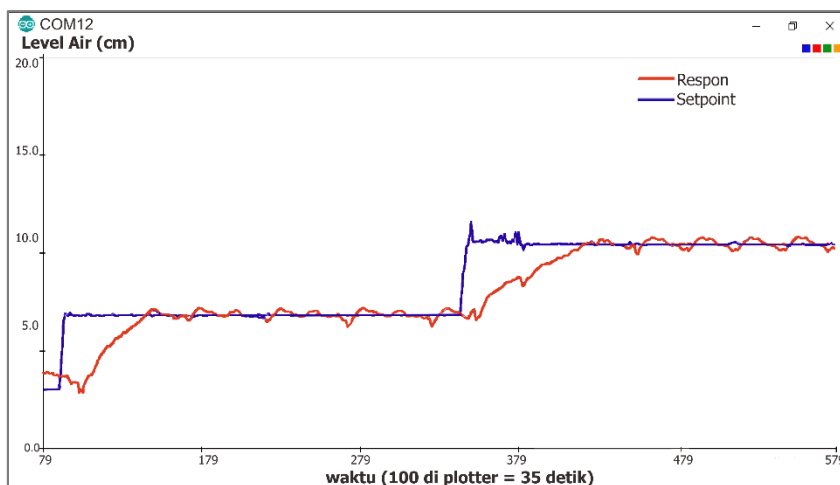
Pengujian pertama yang dilakukan yaitu dengan menggunakan parameter hasil penalaan *Ziegler Nichols* Tipe 1 (ZN1) yaitu $K_p = 14,28$, $T_i = 0,07632$, dan $T_d = 0,01908$ sedangkan *Ziegler Nichols* Tipe 2 (ZN2) yaitu $K_p = 72$, $T_i = 0,08586$, dan $T_d = 0,02146$ pada masing-masing kendali. Parameter – parameter kendali ini disubstitusi kedalam metode PID diskritisasi langsung dan metode *Backward Difference* kemudian dianalisis respon sistemnya.

Hasil respon untuk kendali PID diskritisasi langsung ZN1 ditampilkan seperti pada Gambar 7 sedangkan respon PID *Backward Difference* ZN1 ditampilkan pada Gambar 8. Gambar 9 menunjukkan respon dengan kendali PID diskritisasi langsung ZN2 dan Gambar 10 menunjukkan respon sistem saat menggunakan kendali PID *Backward Difference* ZN2.

Berdasarkan grafik respon yang telah didapat, menunjukkan bahwa respon sistem lebih baik saat diberikan kendali PID dengan metode *Backward Difference* dibandingkan dengan metode diskritisasi langsung. Tampak bahwa pada Gambar 7 dan Gambar 9 ketika *plant* menggunakan metode PID diskritisasi langsung, respon tidak dapat mengikuti *setpoint* yang diberikan bahkan tidak mencapai *steady state*.



Gambar 7. Pengujian ZN1 dengan Diskritisasi Langsung

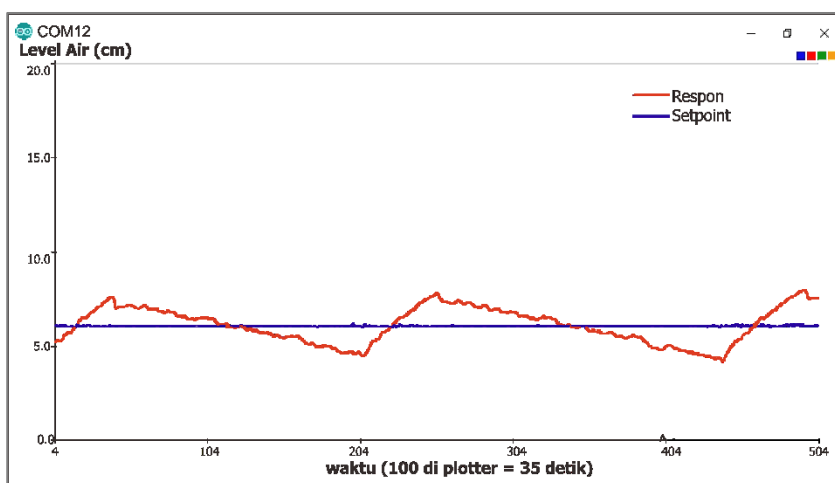


Gambar 8. Pengujian ZN1 dengan *Backward Difference*

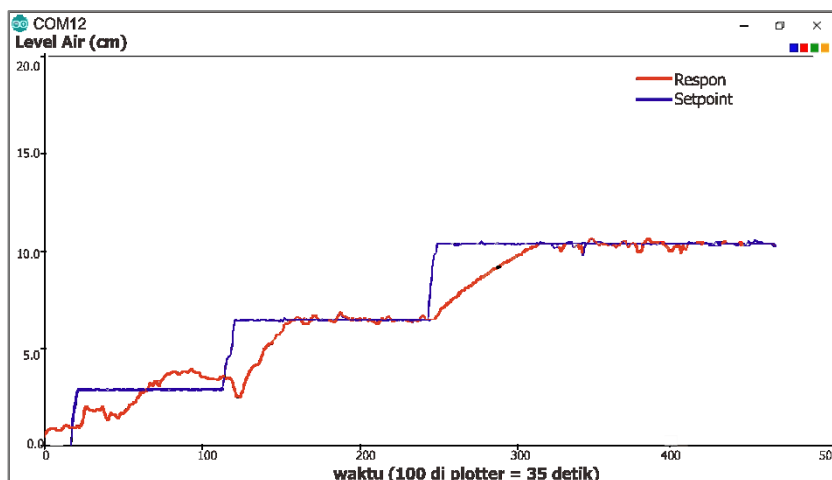
Karakteristik respon seperti parameter *rise time* (t_r), *error steadystate* (e_{ss}), maksimum *overshoot* ($\%M_p$), dan *settling time* (t_s) untuk Gambar 8 ditunjukkan pada Tabel 3.

Tabel 3. Karakteristik Respon Saat Pengujian ZN1 dengan *Backward Difference*

Parameter	Nilai
<i>Rise Time</i> (t_r)	27,47 s (0-5 cm)
<i>Settling Time</i> (t_s)	51,25 s
<i>%overshoot</i> ($\%M_p$)	0%
<i>Error steady state</i> (e_{ss})	0 cm



Gambar 9. Pengujian ZN2 dengan Diskritisasi Langsung



Gambar 10. Pengujian ZN2 dengan *Backward Difference*

Tabel 4. Karakteristik Respon Saat Pengujian ZN2 dengan *Backward Difference*

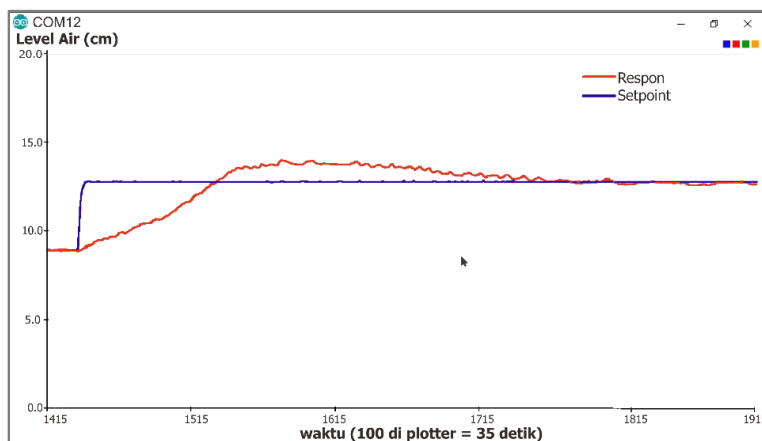
Parameter	Nilai
<i>Rise Time</i> (t_r)	23,87 s (0-5 cm)
<i>Settling Time</i> (t_s)	56.17 s
% <i>overshoot</i> (% M_p)	0%
<i>Error steady state</i> (e_{ss})	0 cm

Karakteristik respon seperti parameter *rise time* (t_r), *error steadystate* (e_{ss}), maksimum *overshoot* (% M_p), dan *settling time* (t_s) untuk Gambar 10 ditunjukkan pada Tabel 4. Hasil pengujian menunjukkan bahwa respon *plant* saat menggunakan metode *Backward Difference* dengan *initial parameter* dari ZN1 maupun ZN2 lebih baik dibanding saat menggunakan metode diskritisasi langsung.

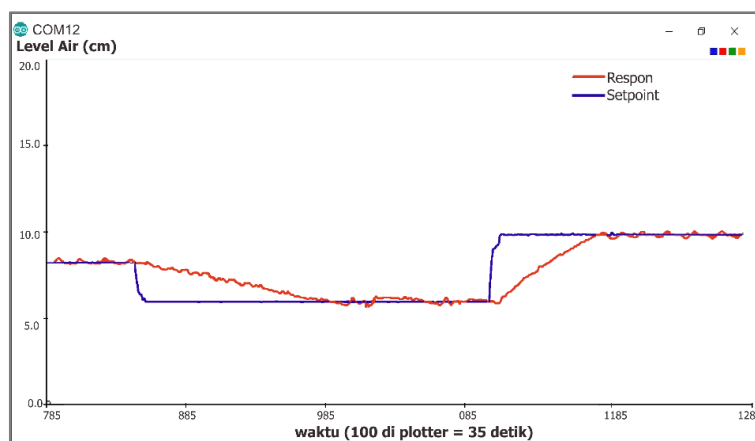
3.2 Uji Perubahan *Setpoint*

Dikarenakan respon sistem saat menggunakan kendali PID diskritisasi langsung baik ZN1 maupun ZN2 (Gambar 7 dan 9) tidak dapat mengikuti *setpoint* yang diberikan, maka dilakukan *tuning* parameter PID secara manual agar mendapatkan hasil respon yang baik. Adapun hasil *tuning* manual didapat dengan parameter K_p , T_i , dan T_d berturut turut adalah 50, 80 dan 0,001. Setelah mendapatkan respon sistem yang diinginkan yaitu dapat *steady* mengikuti *setpoint* yang diberikan, dilakukan pengujian kedua yaitu pengujian terhadap perubahan *setpoint*. Pengujian ini dilakukan untuk mengetahui efek perubahan (diferensial) pada sistem. Hasilnya dapat dilihat pada Gambar 11 untuk respon saat menggunakan PID diskritisasi langsung dan Gambar 12 untuk respon saat menggunakan *Backward Difference*.

Berdasarkan Gambar 11 dan Gambar 12, dilihat bahwa kedua metode kendali menghasilkan respon yang dapat mengikuti terhadap setiap perubahan *setpoint*. Metode PID dengan *Backward Difference* lebih reaktif terhadap perubahan dibandingkan dengan PID dengan diskritisasi langsung. Hal ini dikarenakan PID *Backward Difference* memiliki efek differensial dua kali dari diskritisasi langsung yang dapat dilihat dari persamaan kendali keduanya yaitu Persamaan (6) dan Persamaan (10).



Gambar 11. Pengujian Terhadap Perubahan *Setpoint* pada Sistem dengan PID Diskritisasi Langsung



Gambar 12. Pengujian Terhadap Perubahan *Setpoint* dengan *Backward Difference*

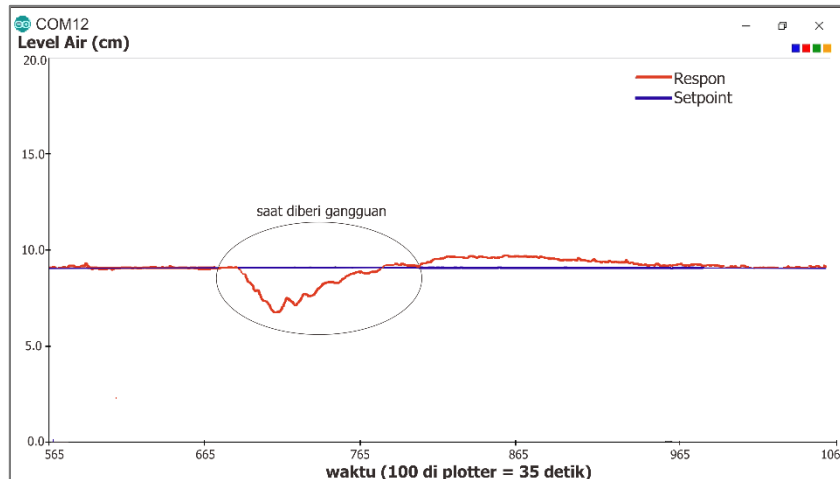
Tabel 5. Analisis Respon Terhadap Perubahan *Setpoint*

Parameter	Diskritisasi Langsung	<i>Backward Difference</i>
<i>Rise time</i> (t_r)	33,91s (0-4cm)	33,66s (0-4cm)
<i>%overshoot</i> (%Mp)	14,05%	0,9%
<i>Settling time</i> (t_s)	151,4s	90,39s
<i>Error Steady State</i> (ess)	0 cm	0,1 cm

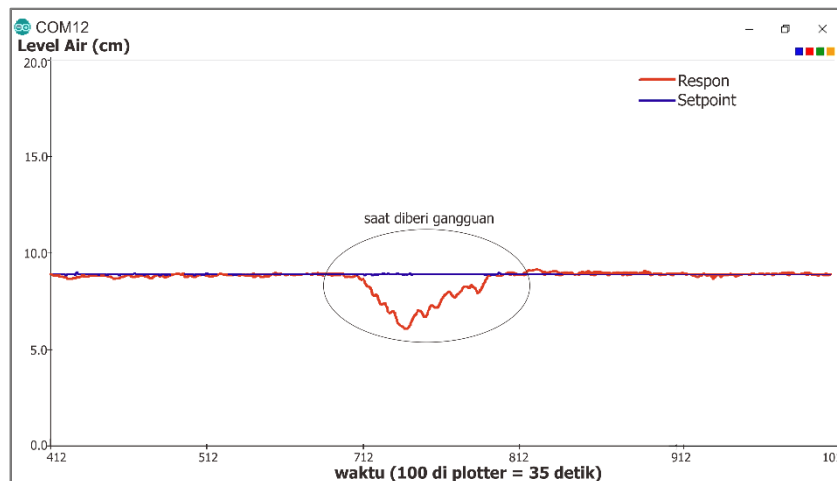
Tabel 5 menunjukkan hasil analisis parameter respon, terlihat bahwa respon dengan metode *Backward Difference* memiliki parameter yang lebih baik dibanding diskritisasi langsung. Salah satu parameter yang terlihat perbedaannya yaitu maksimum *overshoot* (%Mp) dari diskritisasi langsung lebih besar yaitu 14,05% sedangkan *Backward Difference* 0,9%.

3.3 Pengujian dengan Gangguan

Pengujian ini dilakukan untuk mengetahui tingkat keandalan masing-masing pengendali dalam mempertahankan respon agar sesuai dengan yang diinginkan. Uji gangguan pada *plant level* air dilakukan dengan membuka keran pembuangan air $\frac{3}{4}$ bagian dari yang sebelumnya hanya terbuka $\frac{1}{4}$ bagian. Perbandingan hasil respon dapat dilihat pada Gambar 13 dan Gambar 14.



Gambar 13. Pengujian Terhadap Gangguan dengan Diskritisasi Langsung



Gambar 14. Pengujian Terhadap Gangguan dengan *Backward Difference*

Tabel 6. Analisis Respon Terhadap Gangguan

Parameter	Diskritisasi Langsung	<i>Backward Difference</i>
<i>Rise time</i> (t_r)	23,87 s (0 -4cm)	21,95 (0-4cm)
<i>%overshoot</i> (%Mp)	4,88%	1,9%
<i>Settling time</i> (t_s)	123,3 s	84,46 s
<i>Error steadystate</i> (ess)	0cm	0,1cm

Pada analisis respon didapatkan hasil bahwa pengendali PID dengan *Backward Difference* menghasilkan respon yang lebih baik dibandingkan dengan respon diskritisasi langsung dalam menangani gangguan yang diberikan. Pada *plant* dengan kendali PID diskritisasi langsung, saat diberi gangguan, terdapat *overshoot* sehingga respon memerlukan waktu yang lama untuk mencapai *steady state*. Sebaliknya dengan kendali PID *Backward Difference* respon dapat langsung mengikuti *setpoint* sesaat setelah gangguan diberikan. Perbandingan nilai parameter *rise time* (T_r), *error steady state* (ess), maksimum *overhoot* (%Mp) dan *settling time* (T_s) untuk respon dengan kendali PID *Backward Difference* dan diskritisasi langsung saat diberi gangguan dapat dilihat pada Tabel 6.

4. PENUTUP

Dalam penelitian ini, sintesis kendali PID *digital* menggunakan dua metode yaitu diskritisasi langsung dan *Backward Difference* telah berhasil diterapkan pada sistem *plant level* air. Parameter kendali PID diperoleh menggunakan pendekatan grafik respon sistem yaitu metode Ziegler Nichols tipe 1 (ZN1) dan tipe 2 (ZN2). Hasil pengujian menunjukkan bahwa kendali PID diskrit menggunakan *Backward Difference* menghasilkan respon sistem yang lebih baik dibandingkan saat kendali PID didiskritisasi menggunakan metode diskritisasi langsung (*direct discretization*). Hasil pengujian menunjukkan kendali PID diskrit menggunakan *Backward Difference* menghasilkan respon sistem yang lebih baik dibandingkan metode PID diskritisasi langsung dengan nilai K_p , T_i , dan T_d yang digunakan adalah 50, 80 dan 0,001 menghasilkan respon dengan nilai rise time, settling time dan overshoot berturut-turut sebesar 33,66s, 90,39s dan 0,9%. Untuk pengembangan penelitian selanjutnya dapat dilakukan analisis pengaruh diskritisasi menggunakan metode lain, seperti transformasi Z (*backward rectangular*, *forward rectangular*, dan *trapezoidal rule*).

DAFTAR RUJUKAN

- Almanda, S. P., Susanto, E. & Wibowo, A. S. (2016). Desain dan Implementasi Kontrol Ketinggian Air menggunakan Kontrol PID Adaptif. *Jurnal Teknik Elektro*, 3(3), 4059-4064.
- Angel, L., Viola, J. & Paez, a. M. (2019). Evaluation of The Windup Effect in a Practical PID Controller for The Speed Control of a DC-motor System. *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*. (pp. 1 - 6).
- Babu, A. R., Kibreab, S. & Mehari., S. (2020). Experimental Studies on Step Response of Water Level Control System with P, PI and PID Control Mechanisms. *International Research Journal of Engineering and Technology (IRJET)*, 7(10), 1504-1509.
- Chen, Q., Tan, Y., Li, J. & Mareels, I. (2017). Decentralized PID Control Design for Magnetic Levitation Systems Using Extremum Seeking. *IEEE Access*, 6, 3059-3067.
- Chia, K. S. (2018). Ziegler-Nichols Based Proportional-Integral-Derivative Controller for a Line Tracking Robot. *Indonesian Journal of Electrical Engineering and Computer Science*, 9(1), 221-226.
- Chopade, A. S., Khubalkar, Swapnil W., Junghare, A. S., Aware, M. V., Das, Shantanu. (2016). Design and Implementation of Digital Fractional Order PID Controller using Optimal Pole-Zero Approximation Method for Magnetic Levitation System. *IEEE/CAA Journal Of Automatica Sinica*, 3(3), 977-989.
- Coughi, S. & Ntogramatzidis, L. (2013). Direct and exact methods for the synthesis of discrete-time proportional-integral-derivative controllers. *IET Control Theory Applied*, 7(18), 2164-2171.

- Das, S., Halder, K. & Gupta, A. (2020). Delay Handling Method in Dominant Pole Placement based PID Controller Design. *IEEE Transactions on Industrial Informatics*, 16(2), pp. 980-991.
- El-sharif, I. A., Hareb, F. O. & Zerek, A. R. (2014). Design of Discrete-time PID Controller. *International Conference on Control, Engineering & Information Technology (CEIT'14)*, (pp. 110-115).
- Isdaryani, F., Feriyonika, F. & Ferdiansyah, R. (2020). Comparison of Ziegler-Nichols and Cohen Coon Tuning Method for Magnetic Levitation Control System. *Journal of Physics: Conference Series*, 1450, 012033.
- Kumar, A. & Pan, S. (2018). A PID Controller Design Method using Stability Margin with Transient Improvement Criteria. *4th International Conference on Electrical Energy Systems (ICEES)*, pp. 506-510.
- Narkvitul, N., Ukakimaparn, P. & Trisuwannawat, T. (2014). Closed-form Formulas for Continuous/discrete-time PID Controllers' Parameters. *14th International Conference on Control, Automation and Systems (ICCAS 2014)*, p. 1526 – 1530.
- Patel, V. V. (2020). Ziegler-Nichols Tuning Method. *Resonance*, 25, 1385–1397.
- Podržaj, P. (2018). Continuous VS Discrete PID Controller. *IEEE 9th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, (pp. 177–181).
- Verma, B. & Kumar, P. P. (2019). Robust Fine Tuning of Optimal PID Controller with Guaranteed Robustness. *IEEE Transactions on Industrial Electronics*, 67, 4911-4920.