JURNAL SAINS TERAPAN VOL. 6 NO.1 2020 e-*ISSN* 2477-5525 *p-ISSN* 2406-8810

Received : March 2020	Accepted: April 2020	Published : April 2020

Metode Kontrol Skalar dengan Penala Parameter PID Otomatis Menggunakan Algoritma PSO Sebagai Pengendali Kecepatan Motor Induksi Tiga Fasa Berbasis LabView

R. Akbar Nur Apriyanto[1], Era Purwanto[2], Hary Oktavianto[3], Gigih Prabowo[4], Hanif Hasyier Fakhruddin[5], Handri Toar[6]

[1, 2, 3, 4, 5] *Politeknik Elektronika Negeri Surabaya*

[6] *Politeknik Negeri Batam*

akbarnurapriyanto@gmail.com

Abstract

This paper presents the PID parameter tuning automatically for three-phase induction motors with the V/F control method or scalar control. PID control is one type of simple control, its computation is light, easy to implement and known to be tough in dealing with disturbances. But PID has a weakness that is the difficulty of determining the optimal PID parameters, especially when applied to nonlinear systems such as speed control on a scalar control-based induction motor. Output tuning results automatically in the form of the best PID parameter data collection. This study was validated through simulations using the LabView application by testing dynamic speeds and dynamic loads. When testing using the automatic tuning algorithm PID parameters with parameters that have been optimized the Particle Swarm Optimization (PSO) algorithm, the dynamic speed performance characteristic results are better seen from the transient time in the form of average dead time and rise time of less than 1ms. The results of the Global Best Fitness from the PID automatic tuning simulation using the LabView-based PSO algorithm in the form of Kp, Ki, and Kd values can be used as input for setting the speed of the induction motor in real time.

Keywords : autotuning, speed control, induction motor, Particle Swarm Optimization , LabVIEW®

Abstrak

Pada tulisan ini menyajikan penala parameter PID secara otomatis untuk motor induksi tiga fase dengan metode kendali V/F atau kendali skalar. Kendali PID merupakan salah satu tipe kendali sederhana, komputasinya ringan, mudah diimplementasi dan dikenal tangguh menghadapi gangguan. Tetapi PID memiliki kelemahan yaitu sulitnya menentukan parameter PID yang optimal, apalagi bila diterapkan pada sistem *non-linear* seperti pengendalian kecepatan pada motor induksi berbasis kendali skalar. Luaran hasil penalaan secara otomatis berupa kumpulan data parameter PID terbaik. Penelitian ini divalidasi melalui simulasi menggunakan aplikasi *LabView* dengan pengujian kecepatan dinamik dan beban dinamik. Ketika pengujian menggunakan parameter PID algoritma penala otomatis dengan parameter yang telah dioptimalkan algoritma *Particle Swarm Optimization* (PSO), didapatkan hasil karakteristik performa kecepatan dinamik yang lebih baik dilihat dari waktu transien berupa rata-rata *dead time* dan *rise time* kurang dari 1ms. Hasil *Global Best Fitness* dari simulasi penalaan otomatis PID menggunakan algoritma PSO berbasis *LabView* yang berupa nilai Kp, Ki, dan Kd dapat dijadikan input untuk pengaturan kecepatan motor induksi secara *real time*.

Kata kunci : penala otomatis, pengatur kecepatan, motor induksi, Particle Swarm Optimization, LabVIEW®

1. Pendahuluan

Motor induksi terbagi menjadi dua menurut jumlah fasenya yaitu motor induksi satu fase dan motor induksi tiga fase. Motor induksi telah lama menjadi bagian utama dalam penggerak industri, jika sebelumnya kendaraan listrik menggunakan motor DC sebagai penggerak, kini penggerak tersebut mulai beralih ke motor induksi. Motor induksi tiga fase banyak digunakan di bidang industri karena harganya relatif murah untuk bidang industri dan mempunyai keandalan yang tinggi. Dari sisi emisi, motor ini ramah lingkungan karena tidak menghasilkan gas

buang. Keunggulan lainnya adalah motor induksi mudah dirawat, tidak menggunakan sikat penghantar listrik menuju rotor motor sehingga aman dari percikan bunga api, tidak menggunakan magnet permanen yang merupakan logam yang mulai langka [1]. Faktor tersebut yang menjadikan motor induksi mulai dikembangkan oleh para peneliti bidang kendaraan listrik [2].

Dalam prakteknya, ketika supir mengemudikan kendaraan listrik terutama perjalanan dalam kota, motor penggerak kendaraan listrik tersebut jarang berputar dengan kecepatan yang sama disetiap waktu, kecepatannya lebih sering berubah. Untuk itu Kendaraan listrik memerlukan motor induksi yang dapat berputar responsif dengan tingkat presisi yang baik. Faktor tersebut bertolak belakang penggunaan motor induksi terdahulu, seperti kerja kompresor, pompa dan kipas yang selalu berputar dengan kecepatan yang sama.

Dan para pengguna motor induksi terdahulu tidak mempermasalahkan jika kecepatan yang dihasilkan oleh motornya tidak presisi. Pengendalian kecepatan motor induksi dibagi ke dalam dua fokus analisis. Ada yang menganalisis motor induksi sebagai kendali skalar dan yang lain menganalisanya sebagai kendali vektor. Saat mengacu kondisi *transient* pada kendali skalar, maka kendali skalar dalam keadaan *steady state* [3]. Kendali skalar atau biasa disebut juga kendali tegangan/frekuensi (V/F) lebih menekankan memaksa motor memiliki hubungan yang konstan antara tegangan dan frekuensi [4]. Misalnya mengubah tegangan akan mengatur fluksi, dan juga akan mengubah frekuensi atau slip akan mengatur torsi untuk tetap konstan. Kendali skalar sangat mudah diimplementasi karena kendali ini memiliki struktur yang sederhana [5]. Selain itu keuntungan penggunaan kendali skalar adalah cepat diprogram, tidak sulit dan dapat dijalankan menggunakan sistem *open loop* secara ekonomis lebih murah. Kendali skalar secara umum digunakan sebagai pengatur kecepatan berbiaya rendah yang dengan performa yang cukup.

Agar kendali skalar selalu beroperasi pada performa puncaknya, maka kendali skalar perlu kendali tambahan yaitu kendali optimasi. Salah satu kendali optimasi yang sederhana adalah kendali PID. Kendali PID bertugas mengoptimalkan fungsi kendali skalar. Salah satu kelemahan dari kendali PID adalah sulitnya menangani sistem *non-linear*, karena performanya akan semakin menurun sementara motor induksi merupakan sebuah sistem *non-linear* [6]. Kelemahan tersebut dapat diatasi dengan menala ulang (*tune up*) setiap ada perubahan kondisi sehingga performa pengendali kecepatan motor induksi tiga fase berbasis kendali skalar tetap terjaga optimal [6]. Salah satu metode yang digunakan untuk menala ulang parameter PID adalah algoritma penala otomatis.

Penelitian ini bertujuan untuk mengembangkan penelitian tentang LabView di Indonesia dan penelitian ini melakukan pengamatan terhadap respon kecepatan motor induksi tiga fase dengan penalaan parameter PID yang diperoleh dari algoritma *Particle Swarm Optimization* (PSO).

2. Metoda Penelitian

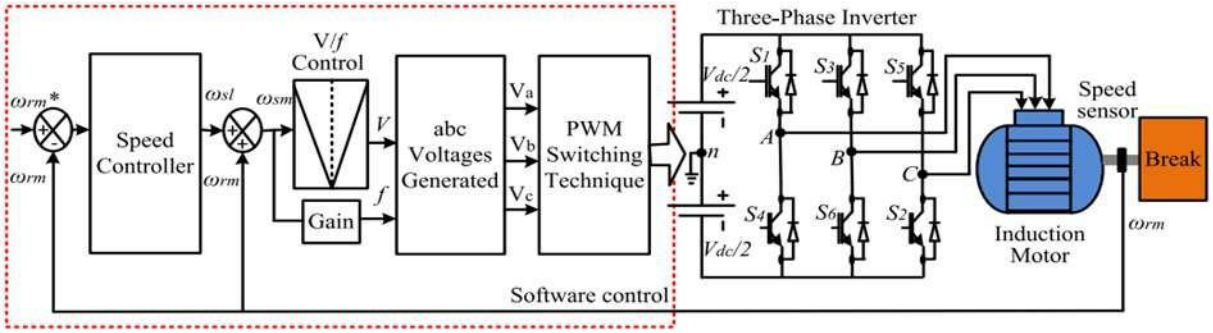
2.1 Kendali Skalar

Kendali skalar pada motor induksi pertama kali dikenalkan pada tahun 1960 oleh Bimal K. Bose. Kendali skalar dibagi menjadi sistem kendali *open loop* dan *close loop*. Pada *open loop* tidak dibutuhkan respon kecepatan yang akurat karena sistem ini tanpa sensor umpan balik. Contoh penerapan *open loop* seperti pada ventilasi, AC, kipas, dan blower [7]. Sedangkan untuk *closed loop*, dibutuhkan sensor kecepatan untuk mengubah tegangan dan frekuensi. Prinsip dasar kendali ini adalah menjaga perbandingan tegangan dengan frekuensi (V/f) selalu konstan, sehingga mempertahankan medan magnet di celah udara (*air gap*) motor induksi[8].

$$\Psi m \cong \frac{Vp}{f} \cong Kv$$

Slip kecepatan (ω_{s1}) adalah variable dari

e-*ISSN* 2477-5525
p-ISSN 2406-8810

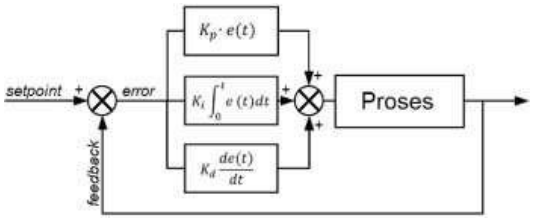


Gambar 1. Diagram blok metode kendali V/F

kecepatan yang terhitung pada karakteristik dasar dari motor induksi. Kecepatan slip motor ditambah dengan kecepatan rotor (ω_{rm}) untuk mendapat (ω_{sm}). Kecepatan sinkron tersebut kemudian dikonversi menjadi frekuensi sinkron untuk mendapat tegangan maksimal dalam pengendali V/f. Batasan utama dari kendali ini adalah performa yang kurang baik pada kecepatan rendah[8], [9]. Blok diagram dari metode kendali V/f dan slip kecepatan (ω_{s1}) ditunjukkan pada Gambar 1.

2.2 Kendali PID

Pada suatu sistem kendali konvensional terdiri dari beberapa pengendali, yaitu kendali proportional, kendali *integral* dan kendali *derivative*. Pada setiap kendali memiliki kelebihan-kelebihan tertentu, dimana kendali *proportional* memiliki kelebihan *rise time* yang cepat, kendali *integral* memiliki kelebihan untuk memperkecil error, dan kendali derivative untuk memperkecil *error* atau meredam *overshoot/undershoot*. Maka dari itu agar dapat memperoleh hasil keluaran dengan *rise time* yang cepat dan *error* yang kecil, ketiga kendali ini dapat digabungkan menjadi kendali PID.



Gambar 2. Diagram Blok kendali PID

PID merupakan kendali sederhana yang terdiri dari *Proportional, Integral, Derivative* [10] dan saling memiliki pengaruh satu terhadap yang lain. Gambar 3 menyatakan bahwa jika ketiga parameter tersebut dikombinasikan secara optimal akan menghasilkan respon luaran yang baik. *Proportional* sangat responsif terhadap sinyal error dan berfungsi penting pada keadaan transien. Integral berperan menjaga kestabilan sinyal pada keadaan *steadystate*. Sementara *derivative* berfungsi untuk memperhalus tugas *proportional*. PID sangat cocok untuk sistem yang linear dan kualitasnya mulai berkurang ketika menghadapi sistem *non-linear* [6]. Bentuk umum dari PID adalah persamaan berikut:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de(t)}{dt} \right) \quad (2)$$

atau dalam bentuk persamaan (3 dan 4) dapat ditulis sebagai berikut:

$$u(t) = K_p\, e(t) + \frac{K_p}{T_i} \int e(t)dt + K_p\, T_d \frac{de(t)}{dt} \quad (3)$$

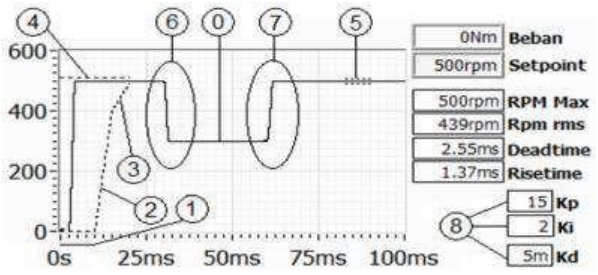
$$u(t) = K_p\, e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (4)$$

dengan $K_i = \frac{K_p}{T_i}$ dan$K_d = K_p\, T_d$.

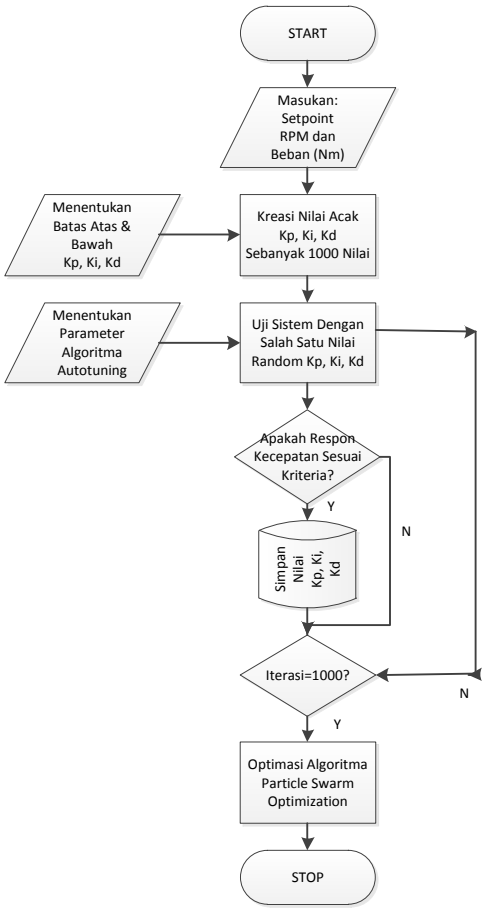
2.3 Algoritma Penala Otomatis

Proses mendapatkan parameter PID tidak serta merta berhasil didapatkan dalam sekali percobaan, proses butuh banyak tenaga yang terkuras dan perlu waktu lama, terutama jika

menggunakan metode tradisional seperti *trial&error*. Untuk itu diperlukan suatu metode agar perolehan parameter tersebut berjalan otomatis. Ziegler dan Nichols adalah peneliti pertama yang mengusulkan metode penalaan PID berdasarkan karakteristik pada keadaan transien [11]. Penggunaan metode ini mensyaratkan sistem harus mengalami waktu transien terlebih dahulu, akan tetapi tidak semua sistem dapat memenuhi syarat tersebut, salah satunya adalah pada kendali skalar pada motor induksi. Sehingga sejak saat itu pengembangan penalaan PID mulai mendapatkan perhatian lebih. Misalkan pada [10] mengembangkan algoritma genetika untuk memperoleh parameter PID pada pengendalian motor induksi hingga 100 generasi yang diuji untuk tiga beban yang berbeda, didapatkan hasil yang optimal. Pada [12] dengan menggunakan modifikasi *Particle Swarm Optimization* (PSO) yang menggunakan koefisien penyempitan waktu yang beragam dapat meningkatkan hasil yang akurat serta dapat menghemat waktu penalaan parameter PID yang berupa nilai konstanta Kp, Ki dan Kd. Algoritma penala otomatis akan mencari parameter PID untuk mengendalikan kecepatan motor induksi berbasis kendali skalar berdasarkan dua masukan yaitu setpoint kecepatan dan respon beban yang dikeluarkan untuk menghasilkan respon kecepatan yang ditentukan pada Tabel 1 dan Gambar 3 menjelaskan bagian-bagian dalam grafik respon kecepatan yang akan digunakan. Flowchart mengenai algoritma penala otomatis ditunjukkan oleh Gambar 4.



Gambar 3. Keterangan pada grafik respon kecepatan.



Gambar 4. Flowchart skema penala otomatis.

PSO merupakan teknik komputasi yang terbaru, dikembangkan oleh Ebehart dan Kennedy pada tahun 1995. PSO terinspirasi dari cara burung berkelompok. Metode ini menjanjikan implementasi yang tidak rumit tetapi memiliki keandalan yang sangat baik [13]. Posisi setiap individu dapat disebut sebagai kandidat solusi (*candidate solution*) untuk setiap permasalahan optimasi. Setiap individu digolongkan kedalam fungsi fitness yang dirancang menyesuaikan dengan masalah. Partikel dari PSO menempati 2 tempat, tempat pertama adalah titik terbaik dalam suatu kelompok atau iterasi *(local best*), sedangkan tempat kedua adalah titik terbaik dari semua iterasi (*global best*). Algoritma PSO bergantung pada 2 faktor, kecepatan (*velocity*) dan posisi dari partikel-partikel. Faktor-faktor tersebut dapat diperbarui menggunakan persamaan (5)(6) [14].

$$V_k^r(t+1) = wV_k^r(t) + c_1 r_1\left(P_k^r(t) - X_k^r(t)\right) + c_2 r_2\left(P_k^r(t) - X_k^r(t)\right) \quad (5)$$

$$X_k^r(t+1) = X_k^r(t) + V_k^r(t+1) \quad (6)$$

Dimana CC adalah laju perubahan sosial untuk masing-masing ; RR merupakan *random* antara 0-1; V adalah faktor kecepatan dari individu I pada iterasi d; t adalah iterasi saat ini; w adalah bobot inersia; dan X adalah faktor posisi. PSO memiliki keunggulan dalam mencapai *konvergen* dan kemampuan untuk memecahkan masalah optimasi yang kompleks dalam berbagai macam domain. Batasan dari PSO adalah mudah terjebak pada lokal optimal, dan jika ada kesalahan dalam memilih parameter, hasilnya akan tidak baik [12].

Tabel 1. Karakteristik respon kecepatan.

No.	*Deskripsi*	*Nilai*
	Time simulation	*=150ms*
	Discrete time (d_t)	*=5e-6s*
1.	*Dead time (t_d)*	*<=10ms*
2.	*Rise time*	*Dead time +10ms*
3.	*Middle*	*<=15ms, <=Setpoint-20%*
4.	*Overshoot*	*<=Setpoint±2%*
5.	*Steadystate (s_t)*	*<=Setpoint±2%*
6.	*Transisi kecepatan turun*	-
7.	*Transisi kecepatan naik*	-
8.	*Parameter PID*	-

Investigasi performa penala otomatis dilakukan dalam simulasi menggunakan spesifikasi komputer yang tertera pada Tabel 2.

Tabel 2. Spesifikasi perangkat computer.

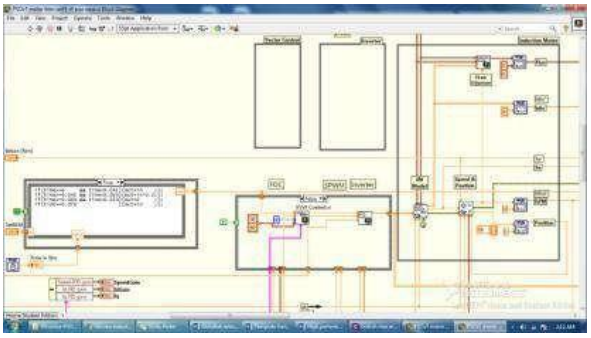
Deskripsi	Spesifikasi
Sistem operasi	Windows 7 64bit
Prosesor	Intel Core i5-2304 4CPU@3.0GHz
Memori	8GB RAM
Perangkat lunak	LabVIEW 2016 versi pelajar
Perangkat keras	MyRIO-1900
Waktu diskrit	5E-6s
Waktu simulasi	100ms dan 600ms

3. Hasil Penelitian

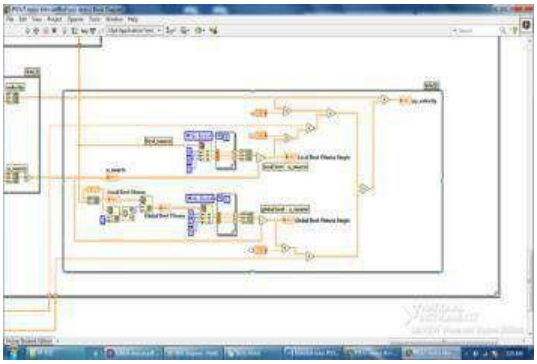
3.1 Desain Simulasi Pada LabView

Perancangan kontrol skalar (V/F) sebagai penggerak motor induksi 3 fase diawali dengan membuat simulasi menggunakan simulink *LabView*. Dari hasil simulasi penala parameter PID otomatis akan didapatkan nilai Kp, Ki, dan Kd. Gambar 5 merupakan diagram blok dari simulasi pada *LabView*.

Gambar 5. Blok diagram simulasi pada LabView.



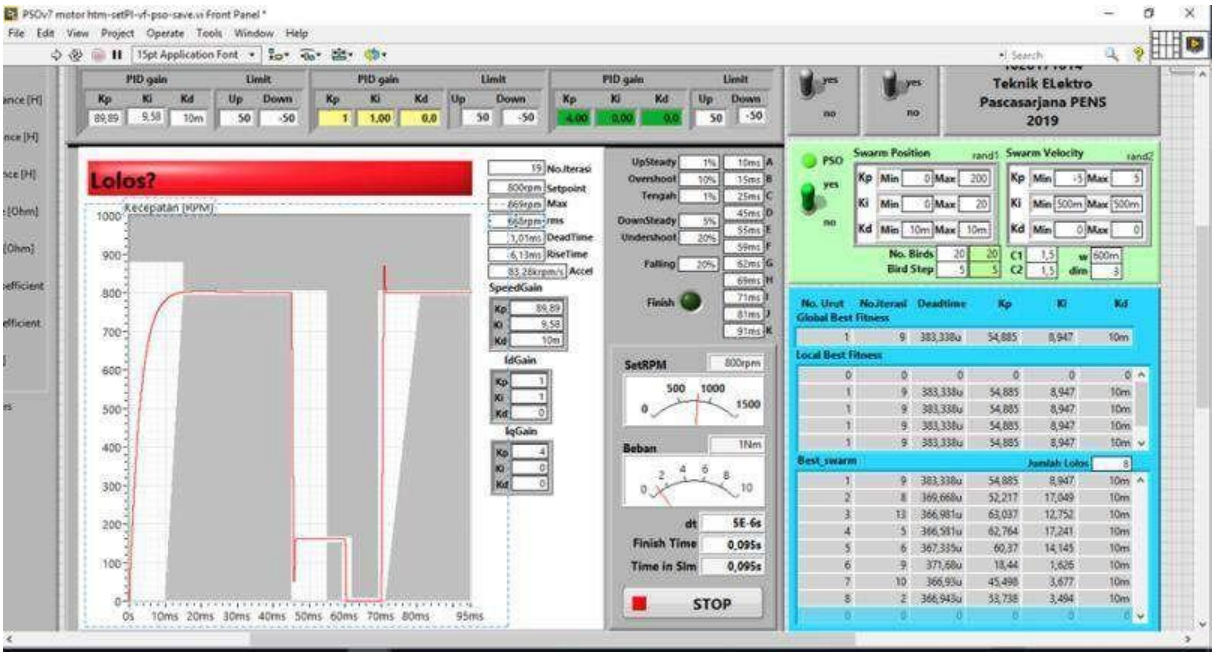
LabView mode simulasi memungkinkan penggunanya untuk memodelkan motor AC sebagai sebuah diagram. Parameter-parameter yang dimasukkan pada permodelan motor induksi (*IM Model*) adalah parameter asli motor 3 fase yang akan digunakan. Sedangkan pada Gambar 6 menunjukkan blok diagram dari penala otomatis menggunakan PSO pada *LabView*.



Gambar 6. Blok diagram PSO pada LabView.

Penggunaan *LabView* pada makalah ini dengan metode penalaan otomatis PID menggunakan PSO memiliki tampilan antar muka pada Gambar 7.

e-*ISSN* 2477-5525
p-ISSN 2406-8810



Gambar 7. Tampilan antar muka simulasi.

Keterangan Gambar 7 yaitu Panel bewarna hijau merupakan pengaturan awal PSO, meliputi rentang nilai (*swarm position*) Kp, Ki, dan Kd, rentang perubahan nilai (*swarm velocity)* Kp, Ki, dan Kd, jumlah burung dalam 1 kelompok dinamakan *number birds*, jumlah kelompok dalam 1 populasi dinamakan *bird step*.

3.2 Peforma PID Penala Otomatis

Penentuan parameter PID menggunakan algoritma penala otomatis ini memerlukan waktu yang cukup lama mengingat sederhananya algoritma yang digunakan tetapi tetap ringan dalam kalkulasinya sehingga proses ini dapat disimulasikan dengan baik menggunakan komputer maupun pada Digital Signal Processing (DSP) berupa *LabVIEW* MyRIO. Keduanya menghasilkan hasil respon yang serupa. Perbedaannya hanya pada waktu komputasi, perbedaan tersebut ditunjukkan pada Tabel 3.

Tabel 3. Waktu komputasi

Perangkat keras	Konsumsi waktu
Komputer	±10menit
LabVIEW MyRIO	±10jam 32menit

Algoritma penala otomatis menggunakan batasan parameter PID yang telah ditentukan pada Tabel 4.

Tabel 4. Batasan nilai parameter PID

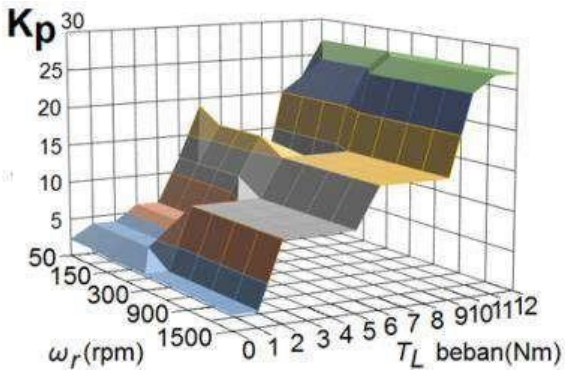
Parameter	Batasan nilai
Parameter P	0-100
Parameter I	0-100
Parameter D	0-0.01

Penelitian ini melakukan beberapa skenario simulasi. Skenario pertama, dan kedua digunakan untuk mengumpulkan hasil dari algoritma penala parameter PID otomatis.

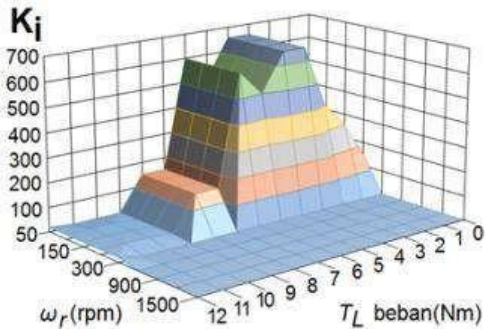
3.2.1Skenario Skenario ke-1: Mendapatkan koleksi parameter penala otomatis

Simulasi algoritma penala otomatis dijalankan mulai dari kecepatan 50rpm hingga 1500rpm dengan kelipatan setiap 100rpm dengan pertambahan beban per 1Nm mulai dari 0Nm hingga mencapai beban puncak yaitu 12Nm, kemudian hasilnya disimpan sesuai dengan format pada Tabel 6. lalu hasil tersebut disajikan dalam bentuk grafik tiga dimensi pada Gambar 8, 9 dan 10.

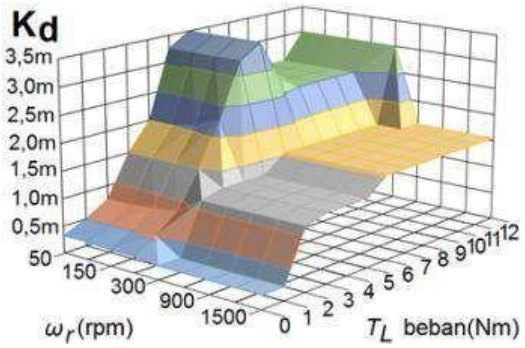
e-*ISSN* 2477-5525
p-ISSN 2406-8810



Gambar 8. Pemetaan parameter Kp.



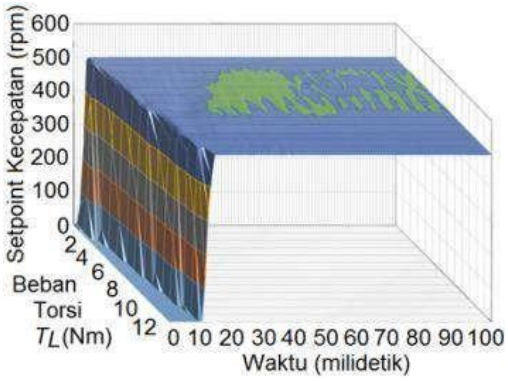
Gambar 9. Pemetaan parameter Ki.



Gambar 10. Pemetaan parameter Kd.

3.2.2Skenario ke-2: Respon penggunaan algoritma penala otomatis pada masa transien di semua beban

Skenario ketiga ini melakukan simulasi untuk mendapatkan respon kecepatan menggunakan algoritma penala otomatis pada setpoint 500rpm mulai dari beban 0 hingga 12Nm dengan kelipatan setiap 1Nm kemudian ditampilkan pada Gambar 11.

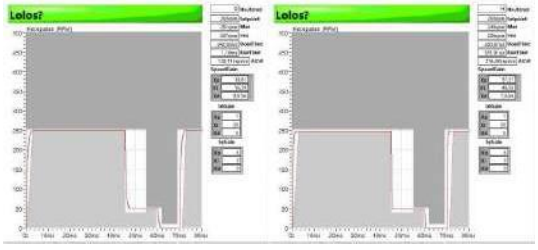


Gambar 11. Respon kecepata pada penala otomatis.

Gambar 11 membuktikan bahwa algoritma penala parameter PID otomatis dapat menjaga kualitas performa kontroler PID dalam mengatur kecepatan motor induksi tiga fase berbasis kontrol skalar. Penggunaan algoritma ini membuat respon kecepatan menjadi seragam pada beban berapapun.

3.3 Hasil Pengujian LabView

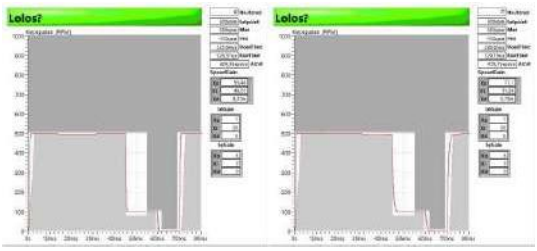
Hasil simulasi algoritma penala otomatis pada pengujian dengan kecepatan dinamis 250rpm, 500rpm, 750rpm, 1000rpm dengan beban berat yaitu 1Nm dan 5Nm ditujukan untuk mengetahui respon kecepatan rendah, menengah, tinggi.



Gambar 12. Respon kecepatan pada algoritma penala otomatis setpoint 250rpm dan beban torsi 1Nm dan 5Nm.

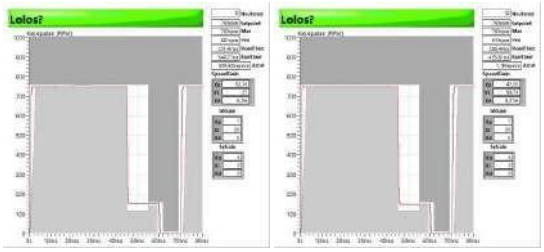
Hasil simulasi algoritma penala parameter PID otomatis antara Gambar 12 menampilkan nilai dead time dan rise time. Pada *setpoint* 250rpm dengan beban torsi 1Nm didapatkan nilai *dead time* sebesar 0,242 ms (milisekon) dan nilai *rise time* 1,16 ms (milisekon). Pada setpoint 250rpm dengan beban torsi 5Nm didapatkan nilai *dead time* sebesar 0,260 ms (milisekon) dan nilai *rise time* 0,581 ms (milisekon).

e-*ISSN* 2477-5525
p-ISSN 2406-8810



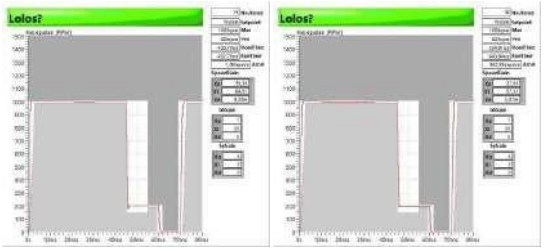
Gambar 13. Respon kecepatan pada algoritma penala otomatis setpoint 500rpm dan beban torsi 1Nm dan 5Nm.

Hasil simulasi algoritma penala parameter PID otomatis antara Gambar 13 menampilkan nilai *dead time* dan *rise time*. Pada *setpoint* 500rpm dengan beban torsi 1Nm didapatkan nilai *dead time* sebesar 0,298 ms (milisekon) dan nilai *rise time* 0,329 ms (milisekon). Pada setpoint 500rpm dengan beban torsi 5Nm didapatkan nilai *dead time* sebesar 0,323 ms (milisekon) dan nilai *rise time* 0,528 ms (milisekon).



Gambar 14. Respon kecepatan pada algoritma penala otomatis setpoint 750rpm dan beban torsi 1Nm dan 5Nm.

Hasil simulasi algoritma penala parameter PID otomatis antara Gambar 14 menampilkan nilai dead time dan rise time. *Pada setpoint* 750rpm dengan beban torsi 1Nm didapatkan nilai *dead time* sebesar 0,331 ms (milisekon) dan nilai *rise time* 0,544 ms (milisekon). Pada setpoint 750rpm dengan beban torsi 5Nm didapatkan nilai *dead time* sebesar 0,396 ms (milisekon) dan nilai *rise time* 0,435 ms (milisekon).



Gambar 15. Respon kecepatan pada algoritma penala otomatis setpoint 1000rpm dan beban torsi 1Nm dan 5Nm.

Hasil simulasi algoritma penala parameter PID otomatis antara Gambar 15 menampilkan nilai dead time dan rise time. Pada *setpoint* 1000rpm dengan beban torsi 1Nm didapatkan nilai *dead time* sebesar 0,430 ms (milisekon) dan nilai *rise time* 0,429 ms (milisekon). Pada setpoint 1000rpm dengan beban torsi 5Nm didapatkan nilai *dead time* sebesar 0,529 ms (milisekon) dan nilai *rise time* 0,640 ms (milisekon).

Dari hasil simulasi untuk *setpoint* 250rpm, 500rpm, 750rpm, 1000rpm dengan beban berat yaitu 1Nm dan 5Nm didapatkan rata-rata *dead time* dan *rise time*. Rata-rata dead time dengan beban 1 Nm sebesar 0,325 ms (milisekon), sedangkan untuk beban 5Nm sebesar 0,377 ms (milisekon). Rata-rata rise time dengan bebas 1Nm sebesar 0,631 ms (milisekon), sedangkan untuk beban 5 Nm sebesar 0,546 ms (milisekon).

Dalam penelitian ini penulis mendapatkan hasil Kp, Ki, dan Kd dari simulasi *LabView* sebagai nilai pengaturan PID dengan algoritma penala otomatis secara *real time* (impelementasi), nilai parameter Kp, Ki, Kd ditunjukkan pada Tabel 5.

Tabel 5. Hasil *global best fitness*

Global Best Fitness					
Rpm	Beban Torsi	*Dead Time*	Kp	Ki	Kd
250	1Nm	0,242ms	19,808	16,288	9,609m
	5Nm	0,260ms	37,273	48,583	7,519m
500	1Nm	0,298ms	11,104	11,69	5,777m
	5Nm	0,323ms	55,457	48,007	9,205m
750	1Nm	0,347ms	16,075	80,713	3,694m
	5Nm	0,396ms	42,055	59,738	8,312m
1000	1Nm	0,454ms	29,322	31,287	5,889m
	5Nm	0,524ms	27,583	57,322	5,813m

4. Kesimpulan

Penelitian ini melakukan pengamatan terhadap respon kecepatan motor induksi tiga fase dengan penalaan parameter PID yang diperoleh dari algoritma *Particle Swarm Optimization* (PSO). Kendali skalar dengan menggunakan metode V/F pada pengendali kecepatan motor induksi telah disimulasikan menggunakan *LabView* dengan kendali parameter PID otomatis menggunakan algoritma *Particle Swarm Optimization.* Desain penala parameter PID algoritma PSO dapat menjaga kualitas performa dari tanggapan kecepatan motor induksi tiga fase. Penggunaan algoritma ini membuat respon kecepatan menjadi seragam pada beban berapapun. Tolak ukur karakteristik performa kecepatan dinamik yang lebih baik dapat dilihat dari masa transien berupa rata-rata *dead time* dan *rise time* kurang dari 1ms. Hasil dari simulasi LabView yang berupa nilai Kp, Ki, dan Kd dapat dijadikan input untuk pengaturan kecepatan motor induksi secara *real time.*

5. Saran

Diperlukan adanya penelitian lebih lanjut analisis *real time* dengan menggunakan nilai *global best fitness* hasil simulasi kendali parameter PID otomatis tersebut, untuk mengetahui hasil simulasi diatas benar-benar nilai yang dibutuhkan saat *real time*.

6. Daftar Pustaka

[1] B. Praharsena, E. Purwanto, A. Jaya, M. R. Rusli, H. Toar, and R. Wk, "Stator Flux Estimator Using Feed-Forward Neural Network for Evaluating Hysteresis Loss Curve in Three Phase Induction Motor," *Emit. Int. J. Eng. Technol.*, vol. 6, no. 1, pp. 168–184, 2018.

[2] D. Dorrell, L. Parsa, and I. Boldea, "Automotive electric motors, generators, and actuator drive systems with reduced or no permanent magnets and innovative design concepts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5693–5695, 2014.

[3] M. A. Martinez-Hernandez, J. M. Gutierrez-Villalobos, S. M. Malagon-Soldara, F. Mendoza-Mondragon, and J. Rodriguez-Resendiz, "A speed performance comparative of field oriented control and scalar control for induction motors," *2016 IEEE Conf. Mechatronics, Adapt. Intell. Syst. MAIS 2016*, 2016.

[4] A. Mu, "A New Induction Motor V / f Control Method Capable of High-Performance Regulation at Low Speeds," vol. 34, no. 4, pp. 813–821, 1998.

[5] A. W. Aaditya, D. C. Happyanto, and B. Sumantri, "Application of Sliding Mode Control in Indirect Field Oriented Control (IFOC) for Model Based Controller," *Emit. Int. J. Eng. Technol.*, vol. 5, no. 2, pp. 255–269, 2018.

[6] R. H. Subrata *et al.*, "Perancangan Pengendali Pid," vol. 14, pp. 1–16, 2017.

[7] R. Saidur, S. Mekhilef, M. B. Ali, A. Safari, and H. A. Mohammed, "Applications of variable speed drive (VSD) in electrical motors energy savings," *Renew. Sustain. Energy Rev.*, vol. 16, no. 1, pp. 543–550, 2012.

[8] C. M. F. S. Reza, M. D. Islam, and S. Mekhilef, "A review of reliable and energy efficient direct torque controlled induction motor drives," *Renew. Sustain. Energy Rev.*, vol. 37, pp. 919–932, 2014.

[9] M. Suetake, I. N. Da Silva, and A. Goedtel, "Embedded DSP-based compact fuzzy system and its application for induction-motor V/f speed control," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 750–760, 2011.

[10] P. Era *et al.*, "Pengembangan Metoda Self Tuning Parameter PID Controller Dengan Menggunakan Genetic Algorithm Pada Pengaturan Motor Induksi Sebagai Penggerak Mobil Listrik," *Emitter*, vol. 1, no. 1, 2009.

[11] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 115, no. 2B, pp. 220–222, 1993.

[12] . A., S. Mu, S. Nakashima, and K. Tanaka, "PID Controller Design of Nonlinear System using a New Modified Particle Swarm Optimization with Time-Varying Constriction Coefficient," *Emit. Int. J. Eng. Technol.*, vol. 2, no. 2, pp. 80–90, 2014.

[13] L. Ramli, Y. M. Sam, Z. Mohamed, M. Khairi Aripin, M. Fahezal Ismail, and L. Ramli, "Composite nonlinear feedback control with multi-objective particle swarm optimization for active front steering system," *J. Teknol.*, vol. 72, no. 2, pp. 13–20, 2015.

[14] F. Salem, M. A. Awadallah, and E. H. E. Bayoumi, "Model Predictive Control for Deadbeat Performance of Induction Motor Drives," vol. 14, no. October, pp. 303–311, 2015.

e-*ISSN* 2477-5525
p-ISSN 2406-8810