

PERFORMANSI PENGENDALIAN KECEPATAN MOTOR INDUKSI PADA MODUL PEMBELAJARAN BERBASIS PLC-PID

Indra, Abraham Akbar K, Taufik Hidayat, Sila Wardono, Murie Dwiyaniti¹

*Politeknik Negeri Jakarta, Jurusan Teknik Elektro, Program Studi Teknik Otomasi Listrik Industri,
Jl prof Dr. GA Siwabessy, Kampus Baru UI Depok 16425*

E-mail : ¹murie.dwiyaniti@elektro.pnj.ac.id

ABSTRACT

Learning about controlling the rotational speed of a three-phase induction motor is one of the compulsory lessons in the Department of Electrical Engineering. Many industries use this induction motor because the construction is simple and easy to maintain. However, to control an induction motor at a constant rotating speed is quite difficult because it requires frequency adjustment. In this research, a PLC-PID-SCADA based induction motor speed module was made to make it easier for students to understand the process of controlling the rotational speed of an induction motor. The PID controller on the PLC provides a signal to the variable speed drive (VSD) to convert voltage data into frequency. The performance or characteristics of this system can be monitored via SCADA. Students can freely adjust the input or Setpoint in this learning module and change the PID parameters to get the best performance. The best performance is generated using the control parameters $K_p = 20$ and $T_i = 1$, namely no overshoot, 2.2 seconds settling time, and 0.1 second delay time.

Keywords: PID, PLC, SCADA, VSD

ABSTRAK

Pembelajaran tentang pengendalian kecepatan putar motor induksi tiga fasa merupakan salah satu mata pelajaran wajib di Jurusan Teknik Elektro. Banyak industri yang menggunakan motor induksi ini karena konstruksinya yang sederhana dan perawatannya mudah. Namun untuk mengendalikan motor induksi pada kecepatan putar konstan cukup sulit karena membutuhkan pengaturan frekuensi. Pada penelitian ini dibuat modul kecepatan motor induksi berbasis PLC-PID-SCADA untuk memudahkan mahasiswa dalam memahami proses pengendalian kecepatan putar motor induksi. Kontroler PID pada PLC memberikan sinyal ke penggerak kecepatan variabel (VSD) untuk mengubah data tegangan menjadi frekuensi. Kinerja atau karakteristik sistem ini dapat dipantau melalui SCADA. Mahasiswa dapat dengan leluasa mengatur input atau Setpoint dalam modul pembelajaran ini dan mengubah parameter PID untuk mendapatkan performa terbaik. Performansi terbaik dihasilkan dengan menggunakan parameter pengendali $K_p = 20$ dan $T_i = 1$, yaitu tidak ada *overshoot*, settling time 2,2 detik, dan waktu tunda 0,1 detik.

Kata kunci: PID, PLC, SCADA, VSD

1. PENDAHULUAN

Pada masa ini, perkembangan teknologi semakin pesat yang mendorong perusahaan dan juga industri memproduksi produk buatan mereka dengan lebih cepat. Salah satu yang dapat mempercepat produksi adalah penggunaan motor-motor listrik sebagai penggerak mesin.

Sebagai penggerak, motor listrik perlu diatur kecepatannya sesuai dengan kebutuhan. Selain

itu, untuk mengetahui kinerjanya, motor listrik perlu dilengkapi dengan system monitoring. Sehingga selain motor listrik diperlukan komponen lain yaitu SCADA, VSD, dan juga PLC.

PLC yang merupakan singkatan dari *Programmable logic controller* memiliki peran yang sangat penting pada pengendali kecepatan motor. PLC berfungsi untuk memproses perintah/input yang diberikan oleh pengguna

Performansi Pengendalian Kecepatan Motor Induksi

secara digital sehingga pengoperasian motor dapat dilakukan dengan mudah dan lebih fleksibel. Selain itu, didalam PLC terdapat Kontrol PID yang dapat membuat respons kecepatan motor tercapai dengan cepat dan tepat.

SCADA merupakan *software* yang digunakan untuk memonitor respon motor terhadap perubahan kecepatan dan melihat beberapa parameter kelistrikan seperti frekuensi, tegangan, arus, kecepatan putar, dan lain-lain.

Sedangkan VSD merupakan pengatur frekuensi motor yang menghubungkan motor dengan PLC [1].

Semua komponen yang ada dihubungkan sedemikian rupa sehingga membentuk modul pengendali kecepatan putar motor induksi.

Pada penelitian ini, kami merancang dan membuat modul pengendali kecepatan putar motor induksi yang digunakan untuk media pembelajaran praktik mahasiswa semester 4 di Program Studi Teknik Otomasi Listrik Industri.

Didalam artikel ini, kami menyajikan performansi pengendalian kecepatan motor induksi yang terdapat dalam modul pembelajaran dengan memvariasikan pengendali Proporsional dan Integral. Besarnya nilai pengendali proporsional dan integral yang dihasilkan dalam penelitian ini akan menjadi acuan mahasiswa dalam pembelajaran.

Beberapa modul pembelajaran tentang motor induksi telah dibuat oleh peneliti antara lain modul inverter pengendali motor [2]-[5], pengaturan kecepatan motor induksi berbasis PLC [6],[7], pengendalian motor induksi berbasis mikrokontroler [8], desain pengendalian motor induksi dengan Matlab [9],[10]. Dari berbagai referensi, belum ada yang membuat modul pembelajaran yang terintegrasi dengan system monitoring. Sehingga penambahan system monitoring berbasis SCADA yang diterapkan pada modul pembelajaran merupakan inovasi dari penelitian ini. Melalui SCADA, performansi pengendalian kecepatan putar motor induksi dapat terpantau dengan baik.

2. METODE PENELITIAN

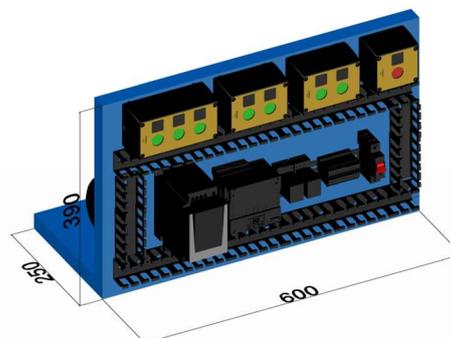
Metode penelitian yang digunakan adalah rancang bangun dan eksperimen. Tahapan penelitian yang dilakukan adalah:

1. Merancang system, memilih spesifikasi peralatan, menggambar tata letak komponen.
2. Membuat deskripsi kerja dan program PLC dan SCADA, serta konfigurasi inverter
3. Membuat modul sesuai dengan perencanaan
4. Melakukan uji coba modul untuk mengetahui performansinya.

Pengujian yang dilakukan adalah memvariasikan parameter pengendali P, dan

PI. Respon system dan data-data dapat dilihat pada PC melalui *software* SCADA.

Rancangan modul pembelajaran system pengendalian motor induksi terlihat pada Gambar 1.



Gambar 1. Modul pembelajaran system pengendalian motor induksi

Modul pembelajaran ini memiliki beberapa tombol tekan yang digunakan sebagai input seperti tombol *start* dan *stop*, *reset*, *preset speed*, dan juga mengaktifkan beban. PLC digunakan kontrol PID yang berfungsi sebagai pengontrol yang membuat kecepatan putar motor stabil.

PLC dihubungkan dengan VSD dengan menggunakan modbus serial, karena PLC tidak dapat langsung dihubungkan dengan motor AC 3 fasa, VSD berperan sebagai aktuator yang menghubungkan langsung dengan motor AC 3 fasa.

Pada modul pembelajaran ini juga terdapat sebuah sensor pembaca kecepatan putar yaitu *rotary encoder*. Fungsinya untuk membaca kecepatan putar motor AC dan *feedback* untuk inputan ke kontrol PID.

Alat ini juga dilengkapi dengan sistem SCADA yang berfungsi sebagai sistem monitoring. Selain itu, SCADA dapat difungsikan sebagai pengatur *setpoint* kecepatan yang diinginkan. Alat ini dibuat dengan tujuan untuk mengatur kecepatan putar motor AC secara stabil sesuai dengan setpoint meskipun diberikan beban, beban yang digunakan pada alat ini adalah sebuah generator.

a. Pengoperasian sistem melalui SCADA

Sistem ini dapat dioperasikan melalui SCADA. Pada tampilan SCADA terdapat kolom input yang berfungsi untuk memasukkan *setpoint* yang diinginkan. Sebelum memasukkan *setpoint*, terlebih dahulu tekan tombol tekan *Start/Stop* untuk memulai sistem, setelah itu tekan tombol *Ready to running* pada SCADA, kemudian masukan *setpoint* ke kolom input. *Setpoint* yang telah ditentukan akan diteruskan ke PLC untuk diolah oleh kontrol PID. Hasil olah dari kontrol

Performansi Pengendalian Kecepatan Motor Induksi

PID adalah *output* PID yang akan diteruskan ke VSD melalui *function block communication*. VSD akan menerima data dari PLC melalui sambungan serial. VSD akan menaikkan frekuensi sampai kecepatan putar motor sesuai dengan *setpoint*. Untuk membaca kecepatan putar motor sudah sesuai dengan setpoint maka digunakan sensor *rotary encoder*. Hasil pembacaan dari sensor ini akan digunakan sebagai *feedback* sehingga harus dihubungkan dengan PLC melalui terminal input PLC. Untuk mengaktifkan beban yang dibangkitkan oleh generator yang digerakkan oleh motor, tekan tombol Beban 1 atau Beban 2 pada SCADA. Setelah motor berhasil dijalankan, terdapat grafik untuk melihat respons sistem.

b. Pengoperasian sistem melalui tombol tekan

Pada sistem ini jika tidak menggunakan SCADA maka terdapat fitur yang tidak bisa digunakan seperti monitoring SCADA yang sangat berguna untuk mengetahui keadaan sistem dan respons sistem. Pengoperasian sistem menggunakan tombol tekan masih dapat dilakukan tetapi hanya dapat memberikan dua buah preset setpoint yaitu 500 rpm dan 1000 rpm. Ketika tombol preset setpoint ditekan maka tombol tekan tersebut akan menjadi input PLC yang akan menjalankan program pada PLC. Output PID akan tetap dihasilkan dan akan dikirimkan melalui *function block communication*. VSD menerima data dan akan menaikkan frekuensi sampai setpoint tercapai. Untuk membaca kecepatan putar motor sudah sesuai dengan setpoint maka digunakan sensor *rotary encoder*. Hasil pembacaan dari sensor ini akan digunakan sebagai *feedback* sehingga harus dihubungkan dengan PLC melalui terminal input PLC. Untuk mengaktifkan beban yang dibangkitkan oleh generator yang digerakkan oleh motor, tekan tombol tekan Beban 1 atau Beban 2.

3. HASIL DAN PEMBAHASAN

a. Pengujian Parameter Propotional Integral (PI)

Pengujian ini dilakukan untuk mencari nilai parameter PI yaitu Kp dan juga Ti dengan metode *trial and error*. Parameter Kp dan juga Ti akan mempengaruhi respons dari kecepatan putar motor terhadap *setpoint*. Respon kecepatan putar motor dapat dilihat pada grafik *trend* SCADA. Berdasarkan hasil data pengujian yang dilakukan, akan ditetapkan parameter PI berupa Kp dan juga Ti.

Untuk mendapatkan data respon kecepatan putar motor dengan berbagai parameter Kp dan

juga Ti, terdapat langkah – langkah yang harus dilakukan, yaitu:

1. Hubungkan PC/Laptop dengan PLC dengan menggunakan kabel Ethernet.
2. Pada *software EcoStruxure Machine Expert – Basic*, buka menu PID lalu atur parameter Kp dan juga Ti.
3. *Download program* yang telah di-*compile* ke PLC.
4. Tekan tombol *Start* lalu tekan tombol *Ready to Running* agar *input* dapat dimasukkan.
5. Masukkan *speed* 100 rpm.
6. Lihat dan *capture* hasil grafik pengujian pada *trend* SCADA.
7. Ulangi langkah 5 dengan memasukkan speed 500 rpm dan 1000 rpm.
8. Ulangi langkah 2 dengan mengganti nilai Kp dan juga Ti setelah langkah 7 telah selesai dilakukan.

Setelah pengujian dilakukan maka didapatkan data berupa grafik, dari grafik tersebut dapat dilihat berapa waktu respon kecepatan putar motor mencapai *setpoint*, berapa *overshot* yang terjadi, dan juga berapa waktu yang dibutuhkan sistem untuk mencapai keadaan mantapnya. Variasi dan hasil parameter PI terlihat pada Tabel 1.

Tabel 1 Parameter PI

Parameter	Kec (rpm)	Os (%)	Rise time (s)	Settling time (s)	Delay Time (s)
KP=1, TI=1	100	0.0%	0	41	1.2
	500	0.0%	0	50	0.7
	1000	0.0%	0	43.8	0.8
KP=5, TI=1	100	0.0%	0	7.8	0.6
	500	0.0%	0	9.2	0.4
	1000	0.0%	0	9.2	0.4
KP=10, TI=1	100	0.0%	0	3.8	1
	500	0.0%	0	4.6	0.7
	1000	0.0%	0	4.6	0.8
KP=15, TI=1	100	0.0%	0	2.4	0.4
	500	0.0%	0	2.4	0.4
	1000	0.0%	0	2.7	0.1
KP=20, TI=1	100	2.5%	2.5	2.4	0.4
	500	0.0%	0	2.2	0.6
	1000	0.0%	0	2	0.8
KP=25, TI=1	100	11.0 %	11	1.9	0.6
	500	16.0 %	80	2	0.4
	1000	2.5%	25	1.7	0.1
	100	2.5%	2.5	4.1	0.4

Performansi Pengendalian Kecepatan Motor Induksi

Parameter	Kec (rpm)	Os (%)	Rise time (s)	Settling time (s)	Delay Time (s)
Kp=30, Ti=1	500	32.0 %	160	1.9	0.4
	1000	15.5 %	155	2.2	0.1
Kp=20, Ti=2	100	0.0%	0	3.8	0.4
	500	0.0%	0	4.8	0.4
	1000	0.0%	0	5.2	0.4
	100	0.0%	0	2.5	0.6
Kp=30, Ti=2	500	0.0%	0	3.5	0.8
	1000	0.0%	0	3.6	0.8
Kp=40, Ti=2	100	8.0%	8	2.6	0.4
	500	1.6%	8	2.4	0.4
	1000	0.0%	0	2.3	0.4
	100	0.0%	0	2.2	0.6
Kp=50, Ti=2	500	0.0%	0	1.5	0.6
	1000	17.2 %	172	2.7	0.4
Kp=30, Ti=5	100	0.0%	0	7.5	0.4
	500	0.0%	0	8.9	0.6
	1000	0.0%	0	8.9	0.4
Kp=50, Ti=5	100	0.0%	0	4.5	0.8
	500	0.0%	0	5.9	0.8
	1000	0.0%	0	6.9	0.6
Kp=70, Ti=5	100	5.0%	5	3.5	0.6
	500	0.0%	0	4.8	0.4
	1000	0.0%	0	5.1	0.6
Kp=90, Ti=5	100	15.0 %	15	2.4	0.4
	500	10.0 %	50	3.4	0.4
	1000	33.0 %	330	2.7	0.2

Berdasarkan Tabel 1 respon kecepatan putar motor yang paling cepat mencapai *setpoint* adalah pada parameter Kp=70, Ti=5 dengan waktu 0.7 detik pada *setpoint* 100 rpm. Respon ditunjukkan pada Gambar 2. Tetapi pada kecepatan 500 rpm dan 1000 rpm, waktu yang diperlukan sistem untuk mencapai *setpoint* dapat dikatakan lambat, yaitu 4.8 detik dan 5.1 detik seperti Gambar 3 dan Gambar 4. Karena pada parameter Kp=70 dan Ti=5 tidak memiliki waktu respon yang konsisten pada waktu menuju *setpoint* maka dilanjutkan dengan menguji parameter PI dengan nilai yang berbeda. Respon tercepat selanjutnya ada pada Kp=30 dengan Ti=1, yaitu dengan waktu respons mencapai *setpoint* selama 1.2 detik pada *setpoint* 100 rpm seperti Gambar 5, 0.9 detik pada *setpoint* 500 rpm, dan juga 0.8 detik pada 1000 rpm. Sebenarnya parameter PI dengan nilai ini memiliki waktu respons yang konsisten yaitu tidak terlampaui jauh, tetapi pada parameter ini

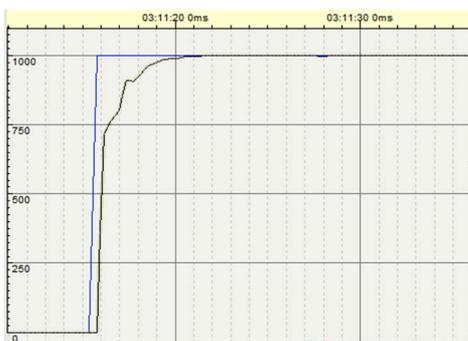
terdapat *overshot* yang nilainya cukup tinggi yaitu pada *setpoint* 500 rpm dengan *overshot* sebesar 32 persen atau 160 rpm seperti Gambar 6 dan juga ada pada *setpoint* 1000 rpm dengan *overshot* sebesar 15.5 persen atau 155 rpm seperti Gambar 7.



Gambar 2 Grafik Kp=70, Ti=5 pada 100 rpm



Gambar 3 Grafik Kp=70, Ti=5 pada 500 rpm

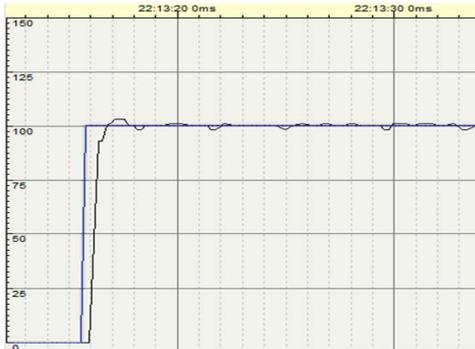


Gambar 4 Grafik Kp=70, Ti=5 pada 1000 rpm

Seperti yang ada pada Tabel 1, jika respons sistem yang diinginkan memiliki waktu respons yang cepat maka respons sistem akan mengalami *overshot*, tentu hal tersebut merupakan hal yang tidak diinginkan. Maka dari itu untuk mencari respon yang minim bahkan tidak memiliki *overshoot* maka hal yang harus dilakukan adalah mengorbankan sedikit waktu mencapai *setpoint*. Pada parameter Kp=20 dengan Ti=1, waktu respons untuk mencapai *setpoint* masih tergolong cukup cepat yaitu sebesar 1.3 detik

Performansi Pengendalian Kecepatan Motor Induksi

pada *setpoint* 100 rpm dengan sedikit sekali *overshot* yaitu sebesar 2.5 persen atau 2.5 rpm dan waktu respons untuk mencapai keadaan mantapnya hanya selama 2.4 detik seperti yang ditunjukkan Gambar 8. sedangkan pada *setpoint* 500 rpm dan 1000 rpm waktu respons yang dibutuhkan untuk mencapai *setpoint* hanya selama 2.2 detik dan 2 detik. Respons kecepatan putar motor pada parameter dengan nilai ini tergolong cukup stabil dan bagus seperti yang ditunjukkan pada Gambar 9 dan Gambar 10.



Gambar 5 Grafik Kp=30, Ti=1 pada 100 rpm



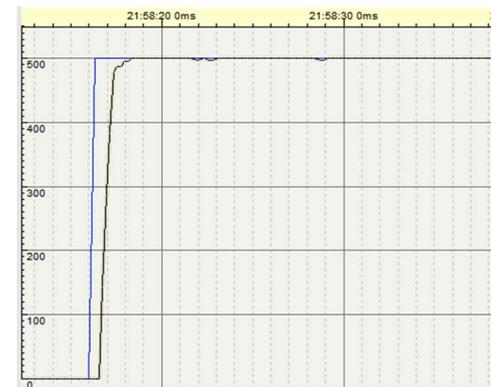
Gambar 6 Grafik Kp=30, Ti=1 pada 500 rpm



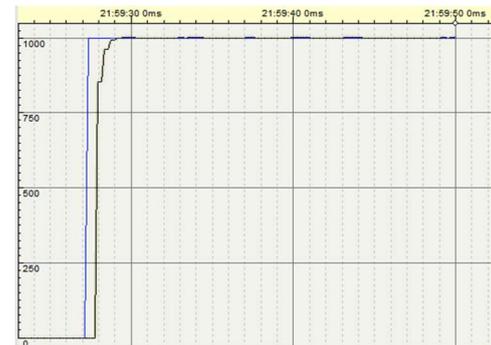
Gambar 7 Grafik Kp=30, Ti=1 pada 1000 rpm



Gambar 8 Grafik Kp=20, Ti=1 pada 100 rpm



Gambar 9 Grafik Kp=20, Ti=1 pada 500 rpm



Gambar 10 Grafik Kp=20, Ti=1 pada 1000 rpm

Berdasarkan Tabel 1 diketahui terdapat waktu tunda. Waktu tunda adalah waktu sistem untuk merespons sebuah *input* berupa *setpoint*. Berdasarkan data, waktu tunda yang berhasil didapatkan memiliki hasil yang tidak konsisten. Tetapi waktu tunda akan lebih lambat ketika nilai Kp rendah dan juga jika nilai Ti tinggi. Waktu tunda terjadi juga bisa disebabkan karena adanya *delay* atau keterlambatan sistem dalam mengirim data.

b. Pengujian Respons Sistem Saat Dibebani

Pengujian ini dilakukan untuk mengetahui perbandingan *output* PID pada saat tanpa beban dengan *output* PID dengan beban dua buah lampu. Pengujian dilakukan dengan melihat grafik pada SCADA dan juga memperhatikan

Performansi Pengendalian Kecepatan Motor Induksi

nilai output PID pada variasi *speed* dengan parameter PID yang telah ditentukan sebelumnya.

Untuk mendapatkan data respon kecepatan putar motor dengan berbagai parameter Kp dan juga Ti, terdapat langkah – langkah yang harus dilakukan, yaitu:

1. Hubungkan PC/Laptop dengan PLC dengan menggunakan kabel Ethernet.
2. Pada software EcoStruxure Machine Expert – Basic, buka menu PID lalu atur parameter Kp = 20 dan juga Ti = 1.
3. Download program yang telah di-compile ke PLC.
4. Tekan tombol Start lalu tekan tombol Ready to Running agar input dapat dimasukan.
5. Masukan speed mulai dari 100 rpm sampai dengan 1300 rpm dengan kelipatan 100 rpm.
6. Catat hasil data pengujian.

Setelah pengujian dilakukan maka didapatkan data berupa pulsa yang dihasilkan sensor, kecepatan *real* motor AC, *output* PID, frekuensi, grafik *trend* SCADA. Dari grafik *trend* SCADA dapat diketahui berapa *overshot* yang terjadi dan juga bagaimana keterangan dari grafik tersebut, apakah beresilasi atau stabil. Data tersebut dapat dilihat pada Tabel 2 untuk sistem tanpa beban sedangkan Tabel 3 merupakan sistem menggunakan beban berupa dua buah lampu.

Tabel 2 Pengujian Tanpa Beban

SetPoint (rpm)	Pulsasensor (pulsa)	Kecepatan (rpm)	Output PID	Frekuensi (x0.1 Hz)	Overshoot	Delay Time (Detik)
100	600	100	95	32	2.00%	0.4
200	1200	200	197	66	0.00%	0.1
300	1800	300	299	100	0.00%	0.4
400	2400	400	400	133	5.75%	0.2
500	3000	500	501	167	0.60%	0.4
600	3600	600	602	201	0.00%	0.8
700	4200	700	702	234	0.00%	0.4
800	4800	800	802	268	0.00%	0.3
900	5400	900	903	301	1.00%	0.3
1000	6000	1000	1004	334	0.00%	0.4
1100	6600	1100	1104	368	0.00%	0.2
1200	7200	1200	1204	401	0.00%	0.4
1300	7800	1300	1305	435	3.08%	0.4

Tabel 3 Pengujian Dengan Beban

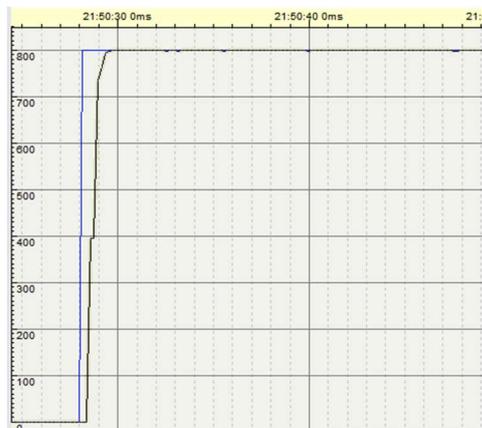
SetPoint (rpm)	Pulsasensor (pulsa)	Kecepatan (rpm)	Output PID	Frekuensi (x0.1 Hz)	Overshoot (rpm)	Delay Time (Detik)
100	600	100	98	33	0.00%	0
200	1200	200	201	67	0.00%	0.4
300	1800	300	303	101	1.33%	0.1
400	2400	400	405	135	1.25%	0.8
500	3000	500	506	169	2.40%	0.4
600	3600	600	606	202	0.00%	0.4
700	4200	700	707	236	1.14%	0.3
800	4800	800	808	269	1.25%	0.7
900	5400	900	909	303	0.00%	0.4
1000	6000	1000	1009	336	7.50%	0.1
1100	6600	1100	1110	370	0.00%	0
1200	7200	1200	1210	403	0.00%	0.1
1300	7800	1300	1311	437	9.46%	0.6

Berdasarkan data pada Tabel 2 dan Tabel 3 maka dapat dilihat perbandingan antara sistem dijalankan tanpa menggunakan beban dengan sistem dijalankan menggunakan beban berupa dua buah lampu pijar. Pada saat sistem dijalankan tanpa menggunakan beban, sistem untuk mencapai *setpoint* hanya membutuhkan lebih sedikit *output* PID dibandingkan dengan sistem dengan menggunakan beban. Sebagai contoh pada saat *setpoint* diatur 400 rpm, sistem yang tidak menggunakan beban hanya mengeluarkan *output* PID sebesar 400 sedangkan ketika diuji dengan menggunakan beban, *output* PID mengeluarkan *output* sebesar 405 meskipun sistem tanpa beban dan yang menggunakan beban sama – sama menggunakan parameter PI yang sama yaitu Kp=20 dan Ti=1. Hal ini terjadi karena beban yang digunakan memberatkan putaran motor yang dipasangkan dengan sebuah generator. Selain itu, alasan lainnya adalah PID *controller* yang digunakan pada sistem ini mendeteksi adanya penurunan pada kecepatan putar motor, oleh karena itu untuk mengembalikan kecepatan putar motor yang menurun tersebut, nilai dari *output* PID akan naik sampai PID *controller* membaca kecepatan putar motor sama dengan *setpoint* yang diberikan.

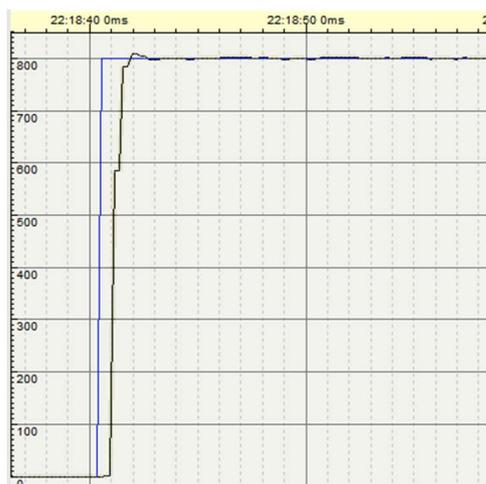
Jika kita bandingkan kedua tabel di atas, pada keterangan dapat lihat bahwa sistem yang tidak menggunakan beban akan berjalan lebih stabil dibandingkan sistem yang menggunakan beban. Sebagai contoh, bisa dilihat grafik pada Gambar 11 yang merupakan contoh grafik sistem tanpa

Performansi Pengendalian Kecepatan Motor Induksi

menggunakan beban pada kecepatan 800 rpm dan juga Gambar 12 yang merupakan contoh grafik sistem yang menggunakan beban pada kecepatan 800 rpm. Hal ini dikarenakan adanya beban yang sedikit mengganggu respons kecepatan dari sistem.

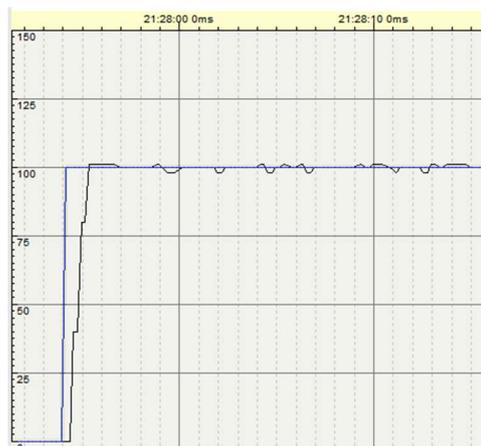


Gambar 11 Grafik kecepatan 800 rpm tanpa menggunakan beban

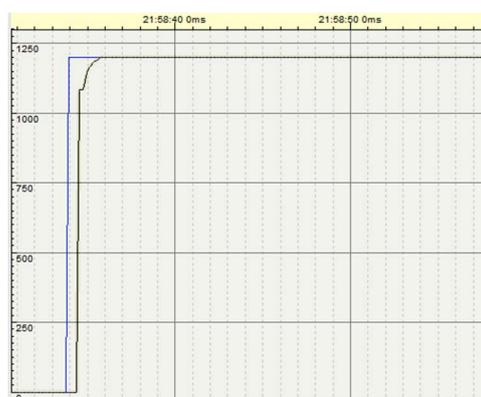


Gambar 12 Grafik kecepatan 800 rpm dengan menggunakan beban

Pada keterangan yang ada pada Tabel 2 dan 3 juga menunjukkan bahwa ketika kedua jenis sistem dijalankan dengan menggunakan kecepatan yang rendah, maka akan mudah mengalami osilasi dibandingkan ketika sistem dijalankan dengan kecepatan yang lebih tinggi. Seperti pada contoh Gambar 13 dan 14 yang merupakan grafik sistem tanpa beban dengan kecepatan 100 rpm dan 1200 rpm dan juga Gambar 15 dan Gambar 16 yang merupakan grafik sistem yang menggunakan beban. Hal ini terjadi karena *output* PID cukup sulit menyesuaikan *setpoint* yang rendah, sensor pembaca kecepatan putar motor juga sangat berperan, karena sensor tersebut merupakan *feedback* untuk PID controller.



Gambar 13 Grafik kecepatan 100 rpm tanpa menggunakan beban

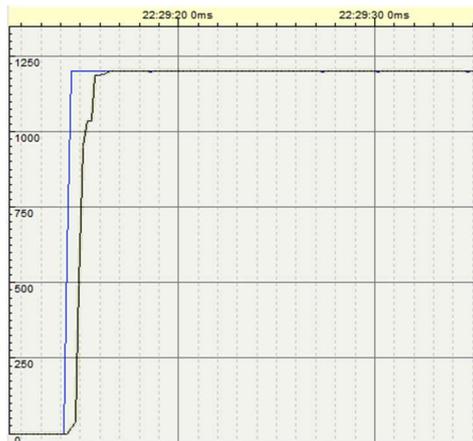


Gambar 14 Grafik kecepatan 1200 rpm tanpa menggunakan beban



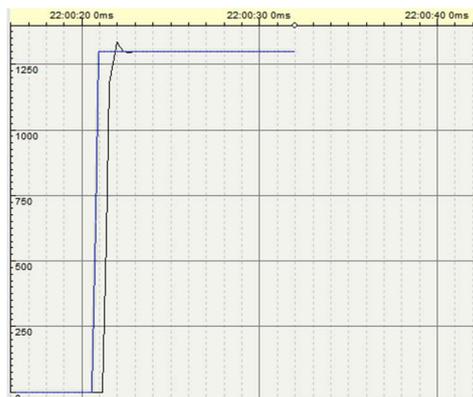
Gambar 15 Grafik kecepatan 100 rpm dengan menggunakan beban

Performansi Pengendalian Kecepatan Motor Induksi

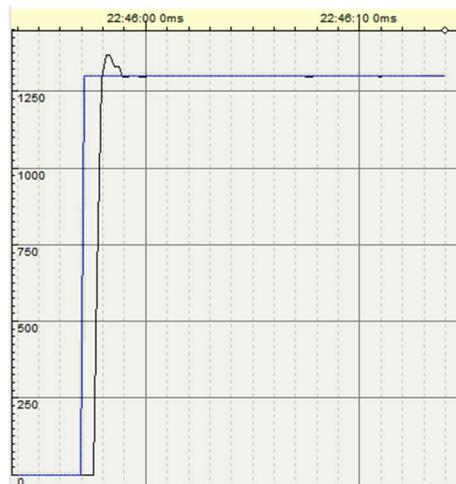


Gambar 16 Grafik kecepatan 1200 rpm dengan menggunakan beban

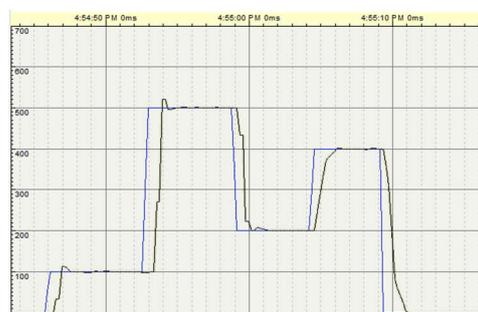
Pengujian ini juga dapat membuktikan bahwa dengan menggunakan beban, sistem akan sedikit terganggu dikarenakan memberatkan putaran motor. Dapat dilihat pada Tabel 2 dan Tabel 3, dengan kecepatan 1300 rpm, *overshot* yang terjadi akan lebih tinggi jika sistem menggunakan beban, *overshot* yang terjadi sebesar 123 jika sistem menggunakan beban, sedangkan sistem yang tidak menggunakan beban hanya sebesar 40 *overshot* yang terjadi. Gambar 17 merupakan contoh grafik *overshot* pada sistem tanpa menggunakan beban sedangkan Gambar 18 merupakan contoh grafik *overshot* pada sistem dengan menggunakan beban. Pada Gambar 19 merupakan grafik percobaan tanpa beban dan Gambar 20 merupakan grafik percobaan berbeban.



Gambar 17 Grafik kecepatan 1300 rpm tanpa menggunakan beban



Gambar 18 Grafik kecepatan 1300 rpm dengan menggunakan beban



Gambar 19 Grafik perubahan kecepatan tanpa beban



Gambar 20 Grafik perubahan kecepatan berbeban

4. KESIMPULAN

1. Semakin besar nilai parameter K_p maka akan semakin cepat respons dari sistem, tetapi jika dibesarkan akan membuat sistem tidak stabil.
2. Semakin besar nilai parameter T_i maka akan semakin lambat respons dari sistem, tetapi jika dibesarkan akan mengurangi overshoot dan sistem lebih stabil.
3. Parameter PID yang tepat digunakan pada sistem ini adalah $K_p=20$ dan $T_i=1$ dan tidak menggunakan parameter T_d .

Performansi Pengendalian Kecepatan Motor Induksi

4. Penggunaan beban pada sistem ini akan memberatkan kinerja dari sistem, ditunjukkan dengan meningkatnya nilai output PID untuk mengimbangi setpoint.
5. Semakin tinggi setpoint yang diberikan maka akan semakin stabil sistem yang diuji, tetapi semakin rendah setpoint yang diberikan maka sistem akan semakin beresilasi.

- [10] Harfina, D. M., & Zaini, Z. 2020. Sistem Pengendali Motor Induksi pada Belt Conveyor dengan Teknik Vektor Kontrol pada VFD menggunakan MATLAB/Simulink. *JTEV (Jurnal Teknik Elektro dan Vokasional)*, 6(1), 13-24.

DAFTAR PUSTAKA

- [1] Saidur, R., Mekhilef, S., Ali, M. B., Safari, A., & Mohammed, H. A. 2012. Applications of variable speed drive (VSD) in electrical motors energy savings. *Renewable and sustainable energy reviews*, 16(1), 543-550
- [2] H. Haryanto, "Pembuatan Modul Inverter sebagai Kendali Kecepatan Putaran Motor Induksi," *Rekayasa*, vol. 4, no. 1, pp. 9–20, 2011.
- [3] R. Saputra, "Rancang Bangun Alat Pengatur Kecepatan Motor Induksi Satu Fasa Melalui Pengaturan Frekuensi Menggunakan," *Skripsi*, 2014.
- [4] E. S. Nasution and A. Hasibuan, "Pengaturan Kecepatan Motor Induksi 3 Fasa Dengan Merubah Frekuensi Menggunakan Inverter ALTIVAR 12P," *J. Sist. Inf.*, vol. 2, no. 1, 2018.
- [5] Z. Zulfikar, N. Evalina, A. Azis, and Y. T. Nugraha, "Analisis Perubahan Kecepatan Motor Induksi 3 Fasa Dengan Menggunakan Inverter 3G3MX2," in *Seminar Nasional Teknik (SEMNASSTEK) UISU*, 2019, vol. 2, no. 1, pp. 174–177.
- [6] N. Evalina, A. Azis, and Z. Zulfikar, "Pengaturan Kecepatan Putaran Motor Induksi 3 Fasa Menggunakan Programmable logic controller," *JET (Journal Electr. Technol.)*, vol. 3, no. 2, pp. 73–80, 2018.
- [7] Badruzzaman, Y. 2015. Sistem Monitoring Kendali Motor Induksi Tiga Fasadengan Variable Speed Drive Berbasis PLC Dan SCADA. *Orbith: Majalah Ilmiah Pengembangan Rekayasa dan Sosial*, 11(2).
- [8] R. Hidayat, D. Notosudjono, and D. Suhendi, "Pengaturan kecepatan putaran motor induksi 1 fasa berbasis mikrokontroler ATMega 8535," *E-journal Progr. Stud. Tek. Elektro Univ. Pakuan Bogor. Diakses pada tanggal*, vol. 26, 2014.
- [9] S. Rifdian and H. Hartono, "Desain dan Simulasi Motor Induksi 3 Fasa dengan Menggunakan Matlab," *J. Penelit.*, vol. 4, no. 2, pp. 70–77, 2019.