SIMULASI KONTROL PID UNTUK MENGATUR PUTARAN MOTOR AC

M. Subchan Mauludin*, Rony Wijanarko, Nugroho Eko Budiyanto
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Wahid Hasyim
Jl. Menoreh Tengah X/22, Sampangan, Semarang 50236.
*Email: aan.subhan18@gmail.com

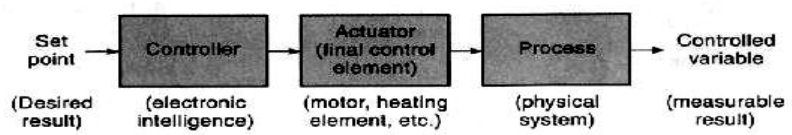
Abstrak

PID (Proportional Integral Derivative) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Didalam suatu sistem kontrol mengenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol derivative . Masing-masing aksi kontrol ini mempunyai keunggulan-keunggulan tertentu, aksi kontrol proporsional mempunyai keunggulan risetime yang cepat, aksi kontrol integral mempunyai keunggulan untuk memperkecil error , dan aksi kontrol derivative mempunyai keunggulan untuk memperkecil derror atau meredam overshot/undershot. Untuk itu agar dapat menghasilkan output dengan risetime yang tinggi dan error yang kecil kita dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID. Simulasi kontrol PID untuk mengatur putaran motor AC. Tujuan dari simulasi ini adalah untuk mendapatkan nilai Kp, Ki, dan Kd yang ideal sehingga di dapat putaran motor AC yang konstant.

Kata Kunci: *Risetime, error, overshoot, PID, motor AC*

1. PENDAHULUAN

Sistem kontrol adalah suatu sistem yang berfungsi untuk mengatur pengontrolan suatu *plant* dengan cara mengatur *input*-nya. Semua sistem kontrol memliliki kontroler dan *actuator*. *Input* pada kontroler biasanya disebut sebagai *set point*. *Actuator* pada alat elektromekanik mengambil sinyal dari kontroler dan diubah dalam bentuk mekanik. Diagram blok sistem kontrol secara umum terdapat pada gambar di bawah ini.



Gambar 1. Diagram Blok Sistem Kontrol
Sumber : Kilian, Christoper T. 2001. *Modern* Kontrol *Technology Components And System.* St. Paul: West Publishing Company

Sistem kontrol dapat digolongkan menjadi dua bagian, yaitu sistem kontrol *loop* terbuka (*open loop*) dan sistem kontrol tertutup (*closed loop*). Perbedaan sistem kontrol *loop* terbuka dan sistem kontrol *loop* tertutup adalah pada ada tidaknya *feedback* pada sistem.

Jika suatu sistem memiliki *feedback*, maka *output*-nya akan berpengaruh pada proses kontrol. Kebanyakan sistem kontrol menggunakan istilah *error* sebagai *feedback* bagi sistem. *Error* adalah perbedaan dari nilai SP (*Set Point*) dan PV *(Present Value)*.

Pada industri-industri yang membutuhkan suatu sistem kontrol dengan kecepatan tinggi dan keakuratan data output, maka pemakaian aksi kontrol PID mungkin masih dianggap kurang memuaskan. Sebab jika menggunakan aksi kendali PID didapatkan jika suatu kontroler di set sangat sensitif, maka *overshot/undershot* yang dihasilkan akan semakin peka, sehingga osilasi yang ditimbulkan akan lebih tinggi, sedangkan bila kontroler di set kurang peka maka terjadinya *overshot/undershot* dapat diperkecil, tetapi waktu yang dibutuhkan akan semakin lama, dan ini akan menjadikan suatu masalah dalam suatu proses industri.

Dasar Teori

Kontrol Proposional

Pada sistem kontrol *proporsional* ini hubungan antara *output* dan *error* adalah sebagai berikut :

$$M(t) = Kp \cdot e(t) \quad (1)$$

Fungsi alihnya dalam transfromasi *Laplace* adalah sebagai berikut :

$$\frac{M(s)}{E(s)} = Kp \quad (2)$$

Salah satu dari ketiga mode unit kontrol yang paling populer dan paling banyak dipakai adalah *unit* kontrol P. Seperti yang tercermin dari namanya, besar *output unit* kontrol P selalu sebanding dengan besarnya *input*. Bentuk dari *transfer function* sangat sederhana dan bentuk diagram bloknya juga sederhana. Seperti yang tergambar dalam gambar 2. *Unit* kontrol P adalah *unit* kontrol yang paling banyak dipakai, baik dalam bentuk pengendali tunggal maupun dengan kombinasi mode *integral* (I) dan *differensial* (D).

Gambar 2. Diagram Blok Pengendali Proporsional
Sumber : Gunterus, Frans. 1994. *Falsafah Dasar : Sistem Pengendalian Proses*. PT Elex Media Komputindo

$$O = Gc \cdot i \quad (3)$$

Gain (Gc) *unit* kontrol *Proporsional* bisa berupa bilangan pecahan. Besarnya tetap, *linier* pada semua daerah kerja dan tidak tergantung pada fungsi waktu. Sepintas istilah *gain* memberikan kesan bahwa ada penguatan dan pembesaran sinyal. Istilah *gain* biasanya jarang dipakai, biasanya menggunakan istilah *Proportional Band* (PB), dimana :

$$Gc = \frac{100\%}{PB} \quad (4)$$

Gain atau disebut dengan PB, dapat diatur besarnya sesaui dengan kebutuhan. Dari persamaan 4 diketahui bahwa gain (Gc) berbanding terbalik dengan PB, kalau gain semakin kecil maka PB semakin besar.

Kekurangan dari pengendalian *proporsional* adalah timbulnya *offset*, dimana besarnya *offset* tergantung pada besarnya *gain* elemen-elemen di dalam *loop* serta *Proportional Band* (PB). Hal ini disebabkan oleh sifat dasar pengendali *proporsional* yang membutuhkan *error* yang menghasilkan *output*.

Kontrol Pengendali *Integral*

Kalau diteliti lebih seksama, *offset* dapat terjadi di pengendali *proporsional* (karena selalu membutuhkan *error*). Jadi untuk menghilangkan *offset*, diperlukan sebuah pengendali yang lain (yang dapat menghasilkan *output* walaupun padanya tidak diberikan *input*). Dengan kata lain, diperlukan pengendali yang menghasilkan *output* lebih besar atau lebih kecil dari bilangan tetap (B) pada saat *input* (*error*) sama dengan nol. Pengendali yang memenuhi kreiteria ini adalah pengendali *integral*, disingkat I.

Sifat dasar pengendali *integral*, yang dapat mengeluarkan *output* pada saat *input* sama dengan nol, adalah sifat dari *unit integrator* yang prinsipnya sama dengan sifat elemen proses orde satu.

Transfer function dari *unit* kontrol *integral* adalah sebagai berikut

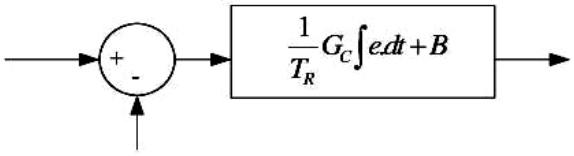
$$O = \frac{1}{Tr} Gc \int e dt + B \quad (5)$$

Dimana :

O = *output*
e = *error* (*input* dari *unit* kontrol)
Tr = *integral time*
B = Bilangan tetap (yang merupakan *bias* atau hasil dari *integral* sebelumnya)
Gc = *gain* dari *controller*

ISBN 978-602-99334-3-7

Besarnya *integral time* (Tr) dinyatakan dalam satuan *minute / repeat*. Artinya, sebuah pengendali *integral* dengan Gc = 1, dikatakan mempunyai *integral time* 2 *minute / repeate* apabila pengendali memerlukan waktu 2 menit untuk mencapai *output* sama dengan *input*. Jadi unsur yang diperhatikan dalam hal ini adalah unsur waktu (*time*). Sebaliknya jika nilai *minute / repeat* kecil, reaksi pengendali akan semakin cepat atau semakin sensitif. Kalau sebaliknya nilai *time / repeat* besar, pengendali akan semakin lambat, atau pengendali akan kurang sensitif. *Integral time* yang kecil pada dasarnya lebih menguntungkan. Walaupun, seperti PB yang kecil, *integral time* yang yang kecil akan membuat *loop* lebih mudah berisolasi. Itu berarti bahwa keadaan tidak stabil akan menjadi lebih mudah terjadi apabila *integral time* dibuat terlalu kecil.



Gambar 3. Diagram Blok Pengendali Integral

Sumber : Gunterus, Frans. 1994. *Falsafah Dasar : Sistem Pengendalian Proses*. PT Elex Media Komputindo

Kontrol Pengendali *Differential*

Karena lambatnya kontrol pengendali PI dalam pengendalian sistem. Upaya memperbaiki *respon* didapat dengan menggunakan *unit* kontrol *differential* atau *derivative*, disingkat D. *Output* pengendali D merupakan *differential* dari fungsi *input*. Sayangnya unsur D tidak dapat mengeluarkan *output* (bila tidak ada perubahan *input*). Karena sifat ini, pengendali D tidak pernah dipakai sendirian. *Unit* pengendali D selalu dipakai dalam kombinasinya dengan P dan I, menjadi pengendali PD atau PID. Selain itu, pengendali D tidak dapat dipakai untuk proses variable yang mengandung *noise*. Karena banyaknya kendala dalam pengendali D, populasi pengendali PID dan PD menjadi tidak sebanyak pengendali P atau PI, (Gunterus 1997, p, 8-3) menyajikan persamaan *transfer function* (O) lengkap dari pengendali *differential* sebagai berikut :

$$\mathbf{O} = Gc.T_D \frac{de}{dt} + B \qquad (6)$$

Dimana :

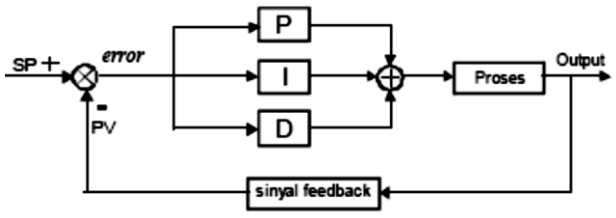
Gc = *gain*

E = *Error*

T_D = *derivative time*

B = *bias*

Unsur *derivatif* yang ada pada *unit* kontrol ini, jika diberi *input* yang naik secara perlahan – lahan dalam bentuk fungsi *ramp*, maka *output* berfungsi *step*. Besarnya *output* tersebut tergantung pada kecepatan naiknya *input* dan TD. Oleh karena itu pengendali ini juga disebut *rate controller*.

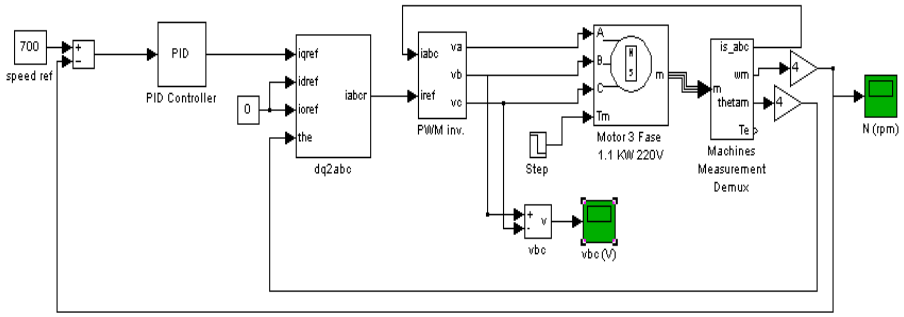


Gambar 4. Blok Diagram Kendali PID

2. METODOLOGI PENELITIAN

Menentukan Plant Model Motor AC

Kendali PID yang dirancang adalah sebagai berikut, yang terdiri dari *block parameter sum, PID controller, dq2abc, inverter, motor induksi dan machine measurement.*



Gambar. 5 ***Simulink control PID pada motor 3 fase***

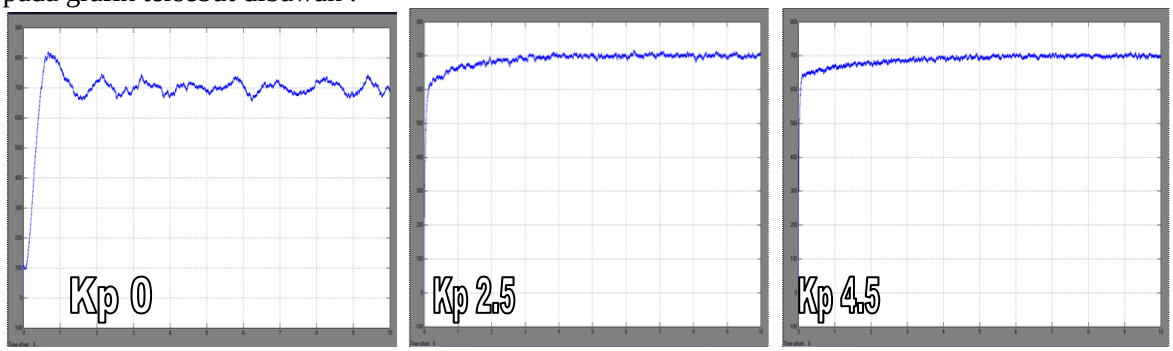
3. SIMULASI DAN PEMBAHASAN

Simulasi dilakukan untuk mengetahui kinerja sistem PID dengan parameter PID yang ditentukan yaitu dengan mengubah nilai-nilai Kp, Ki dan Kd. Dengan perubahan konstanta *proporsional*, *integral* dan *derivative* kita dapat menganalisa putaran motor yang dihasilkan.

Simulasi Dengan Perubahan Konstanta *Proporsional*

Konstanta *proporsional* 0, 2.5, dan 4.5

Pada simulasi dengan menggunakan konstanta *proporsional* sebesar 0, *konstanta* integral sebesar 2.5 dan konstanta *derivative* sebesar 0.1. dalam simulasi tersebut dapat kita lihat hasilnya pada grafik tersebut dibawah :

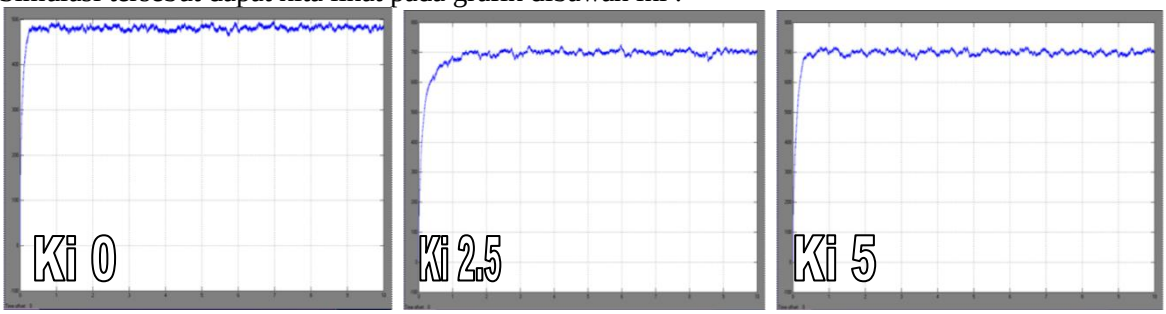


Gambar 6 *Grafik putaran motor dengan menggunakan Kp 0,2.5, 4.5 Ki 2.5 dan Kd 0.1*

Simulasi Dengan Perubahan Konstanta *Integral*

Konstanta *Integral* 0, 2.5, dan 5

Pada simulasi perubahan nilai konstanta *integral* kita mencoba dengan konstanta *proporsional* sebesar 1, sedangkan nilai konstanta *integral* kita lakukan *trial and error* menggunakan nilai konstanta dimulai dari 0 sampai dengan 5 dan konstanta *derivative* sebesar 0.1. Simulasi tersebut dapat kita lihat pada grafik dibawah ini :



Gambar 7. *Grafik putaran motor dengan menggunakan Kp.1, Ki 0, 2.5, 5 dan Kd 0,1*

ISBN 978-602-99334-3-7

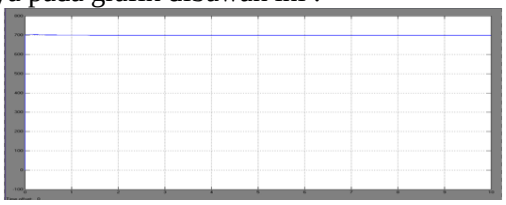
Pada gambar.7 mencoba dengan konstanta *integral* sebesar 5. didapat hasil pada grafik menunjukkan respon yang baik, putaran *out put* sangat cepat medekati putaran sebesar 700 rpm. Respon tersebut diimbangi dengan putaran yang kurang stabil.

Simulasi Dengan Perubahan Konstanta *Derivatif*

Pada simulasi dengan perubahan nilai konstanta *derivative* mencoba dengan konstanta *proporsional* sebesar 1, sedangkan nilai konstanta *integral sebesar* 2.5 sedangkan konstanta *derivative* dilakukan *trial and error* dengan range 0 sampai dengan 0.3.

Konstanta *Derivatif* 0

Pada simulasi ini dicoba dengan menggunakan konstanta *derivative* sebesar 0, Simulasi tersebut dapat kita lihat hasilnya pada grafik dibawah ini :

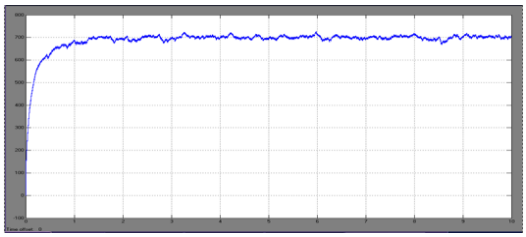


Gambar.8 *Grafik putaran motor dengan menggunakan Kp.1, Ki 2.5 dan Kd 0*

Pada gambar. dengan konstanta *derivatifl* sebesar 0. didapat hasil pada grafik menunjukkan respon yang sangat baik, putaran *out put* sangat cepat mencapai putaran yang diinginkan sebesar 700 rpm. Respon tersebut diimbangi dengan putaran yang sangat stabil.

Konstanta *Derivatif* 0.1

Simulasi dengan menggunakan konstanta *proporsional* sebesar 1, konstanta *integral* sebesar 5 dan konstanta *derivative* sebesar 0.1. dalam simulasi tersebut dapat kita lihat hasilnya pada grafik tersebut dibawah :

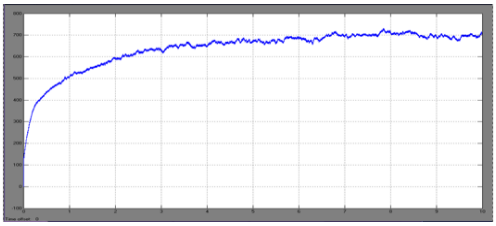


Gambar .9 *Grafik putaran motor dengan menggunakan Kp.1, Ki 2.5 dan Kd 0.1*

Pada gambar.12 mencoba dengan konstanta *derivatifl* sebesar 0.1 didapat hasil pada grafik menunjukkan respon yang baik, putaran *out put* yang naik dengan perlahan mencapai putaran yang diinginkan sebesar 700 rpm. Respon tersebut diimbangi dengan putaran yang stabil.

Konstanta *Derivatif* 0.2

Simulasi dengan menggunakan konstanta *proporsional* sebesar 1, konstanta *integral* sebesar 5 dan konstanta *derivative* sebesar 0.2. dalam simulasi tersebut dapat kita lihat hasilnya pada grafik tersebut dibawah :

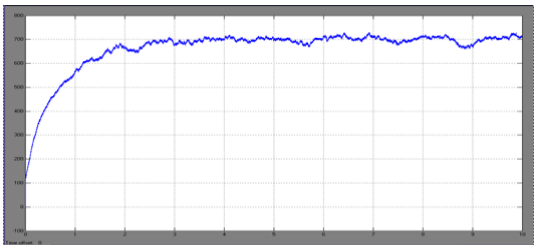


Gambar.10 *Grafik putaran motor dengan menggunakan Kp.1, Ki 2.5 dan Kd 0.2*

Pada gambar.10 dengan konstanta *derivatifl* sebesar 0.2 didapat hasil pada grafik menunjukkan respon yang baik, putaran *out put* yang naik dengan perlahan mencapai putaran yang diinginkan sebesar 700 rpm. Respon kurang bagus kerena responnya lambat.

Konstanta *Derivatif* 0.3

Simulasi dengan menggunakan konstanta *proporsional* sebesar 1, konstanta *integral* sebesar 5 dan konstanta *derivative* sebesar 0.3. dalam simulasi tersebut dapat kita lihat hasilnya pada grafik tersebut dibawah :



Gambar.11 *Grafik putaran motor dengan menggunakan Kp.1, Ki 2.5 dan Kd 0.3*

Pada gambar.11 mencoba dengan konstanta *derivatifl* sebesar 0.3 didapat hasil pada grafik menunjukkan respon yang baik, putaran *out put* yang naik dengan perlahan mencapai putaran yang diinginkan sebesar 700 rpm. Respon sangat lambat dibandingkan dengan memakai konstanta *derivative* sebesar 0.2.

4. KESIMPULAN

Berdasarkan hasil simulasi dan analisa yang dilakukan didapatkan simulasi dengan melakukan pengubahan konstanta *derivatif* dimulai dari 0 sampai dengan 0.3, pengubahan tersebut dengan interval 0.1. Sedangkan konstanta *proporsional* sebesar 1 dan konstanta *integral* sebesar 2.5 diperoleh grafik putaran yang sangat baik pada saat konstanta derivatifnya sama dengan 0, yaitu putaran dapat mencapai putaran yang diinginkan sangat cepat dan setelah mencapai putaran yang dinginkan yaitu sebesar 700 rpm putaran motor tersebut tetap konstan.

DAFTAR PUSTAKA

Bambang Sarjono , Haris Santosa , *Mesin Listrik II (Mesin AC)*, Politeknik Negeri Semarang.

Soelaiman , Mabuchi Magarisawa , 1984, *Mesin Tak Serempak Dalam Praktek*, PT. Pradnya Paramitha

Fitzgerald, 1990., *Mesin-Mesin Listrik* , ErlanggaJakarta,.

Dwi Haryanto, Thomas wahyu, 2001, *Analisis dan Desain system Kontrol dengan Matlab*, Andi ,Yogyakarta,

Guntereus, Frans, 1994, *Sistem Pengendali Proses*, PT. Elex Media Komputindo, Jakarta.

Kontroler PID-Ziegler-Nichols pada Sistem Kontrol posisi berbasis komputer IBM-PC, http/www. Elektro Indonesia.com/, Maret 1998

Mallesham, Gaddam, Rajani, Akula, *Automatic Tuning Of PID Controller Using Fuzzy Logic*, University College of Engineering, Osmania University, Hyderabad, India

Ogata, Katsuhiko, 1997, *Modern Control Engineering*. Prentice Hall Intrenational.

Putranto, Agus, dan kawan-kawan, 2008, *Teknik Otomasi Industri untuk SMK*, Departemen Pendidikan Nasional

Pitowarno, Endro, 2006, *Robotika : Desain, Kontrol, dan Kecerdasan Buatan,* Andi,Yogyakarta.

Sumathi, Sivanandam, and Deepa, 2007, *Introduction to FuzzyLogic using MATLAB*, Springer-Verlag Berlin Heidelberg

ISBN 978-602-99334-3-7