

Makalah Seminar Tugas Akhir

HYBRID KENDALI PI dan FUZZY METODE SUGENO UNTUK PENGATURAN LEVEL CAIRAN BERBASIS MIKROKONTROLER ATMEGA 8535

Hani Purwanti^[1], Wahyudi, S.T, M.T^[2], Iwan Setiawan, S.T, M.T^[2]

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstrak

Suatu sistem kendali proses yang baik harus mempunyai ketahanan terhadap disturbance dan mempunyai respon yang cepat dan akurat. Sering terjadi permasalahan dalam sistem kendali PI, bila parameter PI dibuat sangat sensitif, maka respon sistem terhadap disturbance menghasilkan overshoot/undershoot yang besar sehingga kemungkinan dapat terjadi osilasi semakin tinggi. Bila dibuat kurang sensitif memang akan menghasilkan overshoot/undershoot kecil, tetapi akibatnya akan memperpanjang recovery time. Untuk mengatasi hal ini, diterapkan sistem kendali hybrid PI dan Fuzzy dengan menggunakan mikrokontroler ATmega 8535 sebagai kontrolernya.

Dengan sistem ini kendali hybrid ini diharapkan dapat bekerja untuk meminimalkan overshoot/undershoot yang terjadi dan juga meminimalkan recovery time dari respon system. Logika fuzzy yang didesain mempunyai 2 input yaitu error dan delta error.

Kata kunci : PI, Fuzzy Logic, mikrokontroler ATmega 8535

I. PENDAHULUAN

1.1 Latar Belakang

Dewasa ini teknologi industri terus berkembang dalam berbagai aspek salah satunya dibidang teknologi sistem kendali. Sangat dibutuhkan sistem kendali yang baik untuk dapat menunjang berjalannya proses industri dengan maksud meningkatkan efisiensi dalam proses produksi. Sebagai contoh, otomatisasi dalam bidang industri yaitu proses pengaturan level cairan, pengisian atau pembuangan cairan dalam tangki.

Kendala yang muncul ketika level cairan dalam tangki penampung tidak diketahui, sehingga sering terjadi keadaan tangki yang meluap atau kosong dikarenakan kurangnya pengawasan terhadap tangki penampung. Pada proses industri kimia yang dinamakan proses pemisahan cairan (destilasi), dibutuhkan pengaturan level cairan didalam destilator karena sering terjadi perubahan (deviasi) laju aliran masukan yang disebabkan perubahan tekanan aliran inlet atau juga dikarenakan adanya gaya gesek pada pipa saluran yang mengakibatkan perubahan debit masukan membuat level cairan dalam tangki menjadi tidak stabil.

Pada industri-industri banyak digunakan kendali konvensional seperti PID dan PI namun kendalanya adalah sistem kendali secara konvensional ini dianggap kurang memuaskan, sebab jika kendali konvensional diatur sensitif

maka overshoot/undershoot yang dihasilkan akan semakin peka, sehingga osilasi yang ditimbulkan akan lebih tinggi, sedangkan bila kendali diatur kurang peka maka terjadinya overshoot/undershoot dapat diperkecil, tetapi waktu yang dibutuhkan akan semakin lama, di lain pihak juga berkembang suatu teknologi yang tidak lagi memakai cara konvensional dalam mendapatkan suatu hasil yang diinginkan yaitu dengan menerapkan suatu sistem kemampuan manusia untuk mengendalikan sesuatu. Kemampuan manusia digambarkan dalam bentuk aturan-aturan Jika – maka (If – Then Rules), sehingga proses pengendalian akan mengikuti pendekatan secara linguistik, sistem ini disebut dengan sistem kendali logika fuzzy, yang mana sistem kendali logika fuzzy ini tidak memiliki ketergantungan pada variabel–variabel proses kendali. Sistem ini dikembangkan dalam bidang teknik kontrol, terutama untuk sistem nonlinier dan dinamis.

Pada tugas akhir ini dirancang suatu sistem pengendalian level cairan dengan memanfaatkan mikrokontroler ATmega 8535 sebagai kontroler, dan hybrid kendali PI dan logika Fuzzy untuk mendapatkan respon sistem yang baik yaitu rise time yang cepat dan overshoot yang minimal.

1.2 Tujuan

Tujuan dari Tugas Akhir ini adalah hybrid kendali PI dan logika Fuzzy berbasis mikrokontroler ATmega 8535 untuk mengatur ketinggian cairan pada plant.

1.3 Pembatasan Masalah

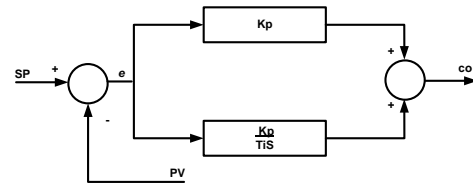
Pembatasan masalah pada tugas akhir ini sebagai berikut :

1. *Plant* yang diuji adalah penampungan cairan dengan bentuk selinder.
2. Konstanta kendali PI diperoleh dengan metode *Ziegler Nichols*.
3. Kendali Fuzzy menggunakan metode Sugeno dengan fungsi keanggotaan masukan himpunan Fuzzy adalah segitiga dan trapesium, metode pengambilan keputusan menggunakan fungsi implikasi MIN dan metode rerata terbobot (*weighted average*).
4. Pengujian dilakukan pada perubahan referensi ketinggian dan pemberian gangguan yaitu gangguan sesaat dan gangguan kontinyu. Gangguan sesaat diberikan dengan cara menutup aliran *outlet* selama 10 detik, hal ini dilakukan untuk mengurangi laju alir pada sisi *outlet* sedangkan gangguan kontinyu diberikan dengan cara memperbesar laju aliran *outlet* dengan menambah laju aliran keluaran plant .
5. Sensor PING digunakan untuk mengetahui ketinggian cairan.
6. Mikrokontroler ATmega 8535 digunakan sebagai kontroler, sedangkan digunakan PC (*Personal Computer*) sebagai monitoring.
7. Bahasa pemrograman yang digunakan pada mikrokontroler adalah dengan menggunakan bahasa C dan delphi 6.0 digunakan dalam proses monitoring. Pada Tugas Akhir ini program monitoring tidak dibahas.

II. DASAR TEORI

2.1 Kendali PI (*Proporsional-Integral*) dengan menggunakan Metode ke-1 Ziegler Nichols

Gabungan aksi kontrol proporsional dan aksi kontrol integral membentuk aksi kontrol proporsional plus integral (*controller* PI). Gabungan aksi ini mempunyai keunggulan dibandingkan dengan masing-masing kontrol yang menyusunnya. Keunggulan utamanya adalah diperolehnya keuntungan dari masing-masing aksi kontrol dan kekurangan dari aksi kontrol yang satu dapat diatasi dengan aksi kontrol yang lain. Dengan kata lain elemen-elemen *controller* P dan I secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem dan menghilangkan *offset*. Untuk dapat mengimplementasikan sistem kendali PI pada komputer, PI harus diubah ke dalam persamaan diskrit. Gambar 1 adalah blok sistem aksi kontrol PI.



Gambar 1 Blok Sistem Aksi Kontrol Proporsional-Integral.

Pengolahan parameter-parameter PI menjadi konstanta-konstanta pengendalian secara diskrit sesuai dengan penjelasan perhitungan berikut. Berdasarkan Persamaan 1 dan dengan menggunakan Transformasi *Laplace*, didapatkan persamaan PI dalam kawasan *s* sebagai berikut.

$$co(s) = K_p E \left[1 + \frac{K_p}{T_i s} \right] \quad (1)$$

Persamaan 2.1 diubah kembali ke kawasan waktu, sehingga menjadi :

$$\frac{dco(t)}{dt} = K_p \frac{de(t)}{dt} + \frac{K_p}{T_i} e(t) \quad (2)$$

Persamaan 2 diubah ke dalam bentuk diskrit, digunakan Persamaan *backward difference*, di mana :

$$\frac{dy(t)}{dt} = \frac{y(k) - y(k-1)}{T} \quad \text{dan}$$

$$\frac{d^2 y(t)}{dt^2} = \frac{y(k) - 2y(k-1) + y(k-2)}{T^2}$$

sehingga Persamaan (2) menjadi:

$$\frac{co(k) - co(k-1)}{T} = K_p \left[\frac{e(k) - e(k-1)}{T} \right] + \frac{K_p}{T_i} e(k) \quad (3)$$

Persamaan 3 menunjukkan persamaan pengendali PI dalam bentuk diskrit, dari Persamaan 3 dapat diketahui bahwa pengendali PI menggunakan konstanta-konstanta pengendalian sebagai berikut :

$$co(k) = co(k-1) + K_2 e(k) - K_3 e(k-1) \quad (4)$$

dengan

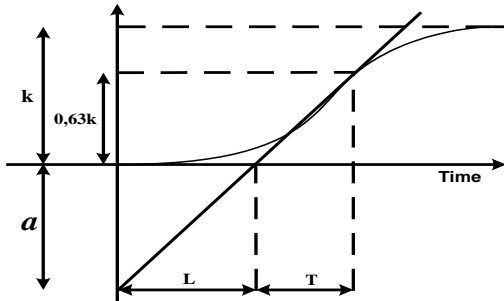
$$K_2 = K_p + K_i \Rightarrow K_i = \frac{K_p T}{T_i} \quad (5)$$

$$K_3 = K_p \quad (6)$$

Metode ke-1 Ziegler-Nichols

Metode ke-1 Ziegler Nichols didasarkan pada respon *plant* terhadap masukan tangga (*step*) dalam kalang terbuka. *Plant* yang tidak mempunyai

integrator, akan menghasilkan kurva tanggapan terhadap masukan tangga, seperti kurva yang berbentuk huruf S pada Gambar 2. Jika ditarik garis singgung pada titik peralihan kurva S dan melanjutkannya sampai memotong garis kondisi tunak, maka diperoleh parameter waktu mati (*dead time*), L , dan konstanta waktu, T .



Gambar 2 Kurva tanggapan berbentuk S.

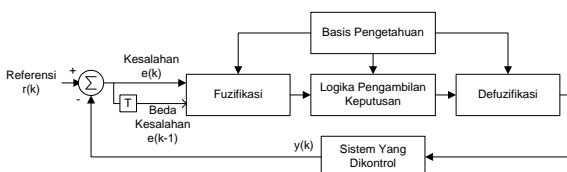
Parameter-parameter yang didapat dari kurva reaksi digunakan untuk menentukan parameter-parameter pengendali PID berdasarkan tetapan-tetapan empiris Zielger-Nichols. Rumus-rumus untuk mencari parameter pengendali menggunakan metode kurva reaksi ditabelkan pada Tabel 1.

Tabel 1 Rumus parameter PID dengan penalaan Ziegler-Nichols metode ke-1.

Pengendali	K_p	T_i	T_d
P	$1/a$	-	-
PI	$0,9/a$	$3L$	-
PID	$1,2/a$	$2L$	$L/2$

2.2 Logika Fuzzy^[11]

Fuzzy berarti samar, kabur atau tidak jelas. Fuzzy adalah istilah yang dipakai oleh Lotfi A Zadeh pada bulan Juli 1964 untuk menyatakan kelompok / himpunan yang dapat dibedakan dengan himpunan lain berdasarkan derajat keanggotaan dengan batasan yang tidak begitu jelas (samar), tidak seperti himpunan klasik yang membedakan keanggotaan himpunan menjadi dua, himpunan anggota atau bukan anggota. Gambar 3 menunjukkan mekanisme kendali logika Fuzy.

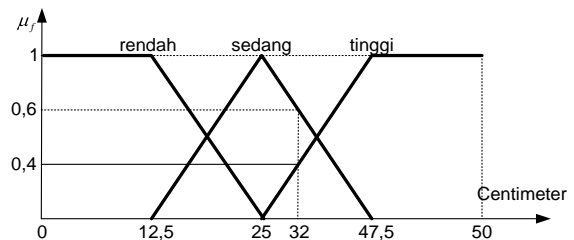


Gambar 3 Mekanisme kendali logika Fuzzy kalang tertutup.

Kendali logika Fuzzy dilakukan dalam tiga tahap, yaitu fuzzifikasi, evaluasi aturan (basis pengetahuan) dan mekanisme pengambilan keputusan dengan metode rerata terbobot.

Unit Fuzzifikasi (*Fuzzification Unit*)

Unit fuzzifikasi pada dasarnya berfungsi untuk mengubah masukan yang berupa data *crisp* menjadi nilai linguistik pada semesta pembicaraan tertentu. Proses fuzzifikasi dapat diperlihatkan dengan contoh berikut ini, jika tinggi muka cairan merupakan suatu variabel linguistik, dengan nilai linguistik $T(\text{tinggi}) = \{\text{rendah, sedang, tinggi}\}$, dimana semesta pembicaraannya terletak antara level 0 cm sampai dengan level 50 cm, dan distribusi fungsi keanggotaan segitiga diperlihatkan pada Gambar 4.



Gambar 4 Proses Fuzzifikasi.

Jika masukan adalah tinggi muka cairan sebesar 32 cm, maka fuzzifikasi menghasilkan nilai linguistik rendah dengan derajat keanggotaan 0, sedang dengan derajat keanggotaan 0,6, dan tinggi dengan derajat keanggotaan 0,4. Syarat jangkauan masukan untuk fuzzifikasi adalah berada dalam semesta pembicaraan Fuzzy.

Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi pengetahuan sistem kendali sebagai pedoman evaluasi keadaan sistem untuk mendapatkan keluaran kendali sesuai yang diinginkan perancang. Basis pengetahuan terdiri dari basis data dan basis aturan Fuzzy.

1. Basis Data

Basis data merupakan komponen untuk mendefinisikan himpunan Fuzzy dari masukan dan keluaran.

2. Basis Aturan Fuzzy

Basis Aturan Fuzzy merupakan kumpulan pernyataan aturan 'IF-THEN' yang didasarkan kepada pengetahuan pakar.

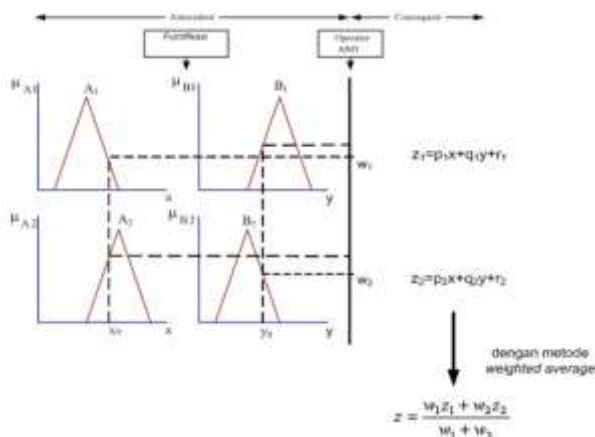
Metode yang digunakan dalam penentuan basis aturan Fuzzy adalah menggunakan metode *trial and error*. Logika pengambilan keputusan disusun dengan cara menuliskan aturan yang menghubungkan antara masukan dan keluaran sistem Fuzzy. Aturan ini diekspresikan dalam

kalimat: 'jika <masukan> maka <keluaran>'. Metode ini mempunyai bentuk aturan seperti persamaan 7

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = k \quad (7)$$

Mekanisme Pengambilan Keputusan (Fuzzy Reasoning)

Logika pengambilan keputusan disusun dengan cara menuliskan aturan yang menghubungkan antara masukan dan keluaran sistem Fuzzy. Aturan ini diekspresikan dalam kalimat: 'jika <masukan> maka <keluaran>'. Pengambilan keputusan dari beberapa variabel sistem dilakukan sesuai dengan proses berpikir seorang operator ketika sedang mengendalikan suatu sistem. Proses eksekusi aturan ini adalah saat nilai masukan berupa himpunan Fuzzy dibaca oleh aturan 'jika-maka', maka tiap kaidah aturan yang mempunyai nilai kebenaran akan dieksekusi sehingga menghasilkan keluaran Fuzzy. Proses pengambilan keputusan menggunakan metode Sugeno, maka akan didapatkan nilai tegas dengan perhitungan rerata terbobot. Metode Sugeno dengan rerata terbobot dapat dilihat pada Gambar 5.



Gambar 5 Proses pengambilan keputusan metode Sugeno.

2.3 Model Kontrol Proses Level Fluida^[3]

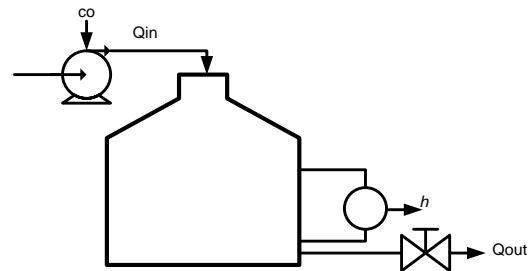
Model kontrol diperoleh dari suatu proses level fluida dapat dilakukan dengan menerapkan hukum kesetimbangan massa yaitu: "Laju akumulasi massa = Laju massa yang masuk - Laju massa yang keluar". Gambar 6 merupakan model kontrol proses fluida.

$$\frac{d}{dt} \rho A h(t) = \rho q_{in} - \rho q_{out} \quad (9)$$

Dimana :

- q_{in} = Perubahan laju aliran fluida input sekitar nilai nominalnya (m^3/det).
- q_{out} = Perubahan laju aliran fluida output sekitar nilai nominalnya (m^3/det).

- ρ = densitas (kg/m^3).
- A = Luas penampang tanki (m^2).
- H = Perubahan ketinggian fluida dalam tanki (m).



Gambar 6 Model kontrol proses level fluida.

Penurunan rumus :

$$q_{out} = K_{out} \cdot h(t) \quad (10)$$

K_{out} = konstanta proporsionalitas yang harganya tergantung pada bukaan valve beban.

Diasumsikan aliran input linear, maka perubahan debit fluida input akan proporsional terhadap besar perubahan sinyal kontrol (co) penggerak .

$$q_{in} = K_{in} \cdot co \quad (11)$$

Substitusikan persamaan 10 dan persamaan 11 pada persamaan 9 dengan melakukan sedikit penyederhanaan, maka diperoleh persamaan diferensial linear.

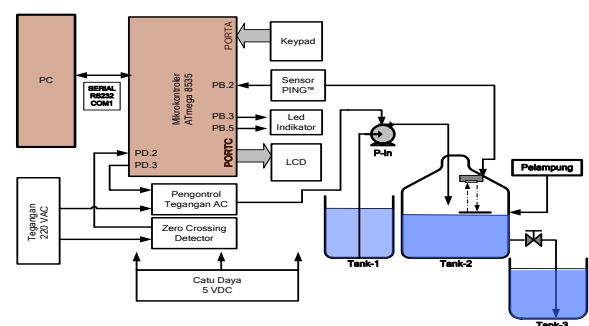
$$\frac{A dh(t)}{K_{out} dt} = -h(t) + \frac{K_{in}}{K_{out}} co(t) \quad (12)$$

Atau dapat ditulis dalam bentuk umum .

$$T \frac{dh(t)}{dt} = -h(t) + K \cdot co(t) \longrightarrow \frac{h(s)}{co(s)} = \frac{K}{Ts + 1}$$

2.4 Sistem Plant Pengaturan Level Cairan

Blok perangkat keras sistem pengaturan level cairan pada Tugas Akhir ini dapat dilihat pada Gambar 7.



Gambar 7 Perangkat keras plant pengendali muka cairan.

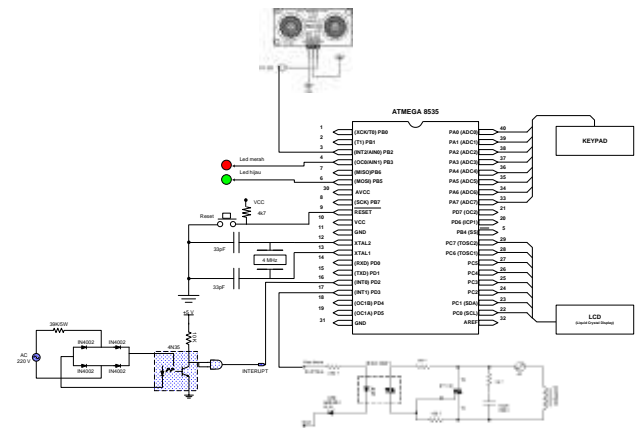
Penjelasan dari masing-masing blok sistem pengendali muka cairan pada Gambar 7 adalah sebagai berikut :

1. Sensor PING™ merupakan sensor yang akan mendeteksi ketinggian sensor terhadap pelampung yang direpresentasikan sebagai level cairan. Keluaran sensor ini berupa tegangan digital, yaitu memiliki logika 1 saat *transmitter* memancarkan gelombang ultrasonik dan memiliki logika 0 saat *receiver* menerima gelombang ultrasonik yang telah dipancarkan.
2. *Keypad* berfungsi sebagai media masukan untuk mengatur *set point* level cairan dan sebagai tombol untuk menjalankan proses pengendalian.
3. LCD (*Liquid Crystal Display*) dan *driver* LCD berfungsi sebagai media tampilan selama proses pengendalian berlangsung.
4. *Zero crossing detector* adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus tegangan jala-jala AC 220 V ketika melalui persimpangan titik nol.
5. Rangkaian pengendali tegangan AC adalah rangkaian yang berfungsi untuk mengendalikan tegangan beban dengan memberi waktu tunda pemicuan *triac*. Komponen utama pengatur tegangan AC berupa *triac* dan optoisolator *MOC3021*.
6. Pompa berfungsi sebagai *aktuator*.
7. Catu daya berfungsi sebagai suplai sistem keseluruhan.
8. Mikrokontroler AVR ATmega 8535 yang berfungsi sebagai pusat pengendalian pada sistem pengatur level cairan ini dapat diprogram dengan menggunakan bahasa C *embedded*.
9. Personal komputer sebagai unit monitoring yang akan menampilkan data ketinggian aktual air berupa grafik respon sistem. Grafik ini selanjutnya dapat digunakan untuk keperluan analisis.

Sistem mikrokontroler digunakan sebagai unit kendali utama yang didalamnya berisi program untuk mengendalikan proses pengaturan pompa (*pump*), termasuk pengesetan parameter, pengaturan tampilan LCD, pengaturan data serial dan input *clock* sensor PING, kendali PI dan penalaan dengan logika Fuzzy serta mengatur aliran data komunikasi serial melalui RS232 dengan komputer untuk proses *monitoring*. Secara umum, alokasi penggunaan *port* pada rangkaian ATmega 8535 dilihat pada Tabel 2 dan Gambar 8.

Tabel 2 Penggunaan *port* pada Atmega8535.

PORT	ATMEGA8535	Fungsi
Port A	PORTA.0 – PORTA.7	Keypad
Port B	PORTB.2 PORTB.5	Sensor PING™ Led Hijau (Indikator Auto)
Port C	PORTC.0 – PORTC.7	LCD
Port D	PORTD.2 PORTD.3	Intrupsi <i>Zero Crossing Detector</i> Output ke Pompa (MOC3021)



Gambar 8 Alokasi *port* pada sistem minimum mikrokontroler ATmega 8535.

PortA.0 - PortA.7 mikrokontroler digunakan sebagai masukan *keypad* 4x4. *PortB.2* dihubungkan dengan data serial sensor PING, data ini yang nantinya digunakan sebagai pentransferan data level dari sensor ke mikrokontroler. *PortC* dihubungkan untuk tampilan LCD, *mode* antarmuka LCD yang digunakan adalah *mode* 4 bit. *PortD.2* adalah input dari rangkaian *Zero Crossing Detector* yang akan mengaktifkan interupsi *eksternal0*. *PortD.3* pada mikrokontroler berfungsi sebagai PWM untuk mengatur besarnya pulsa yang masuk ke *driver* MOC3021, yang kemudian digunakan sebagai sinyal pemicuan yang akan mengendalikan tegangan input pompa (*pump*).

III. PERANCANGAN

Pemrograman mikrokontroler ATmega 8535 dapat dilakukan dengan menggunakan bahasa *assembly* dan C. Perancangan perangkat lunak Tugas Akhir ini digunakan bahasa C dengan kompilator CodevisionAVR versi 1.24. Pemilihan bahasa C dikarenakan kemudahan, kesederhanaan, serta fleksibilitas pemrograman karena selain perintah-perintah dalam bahasa C dapat pula disisipkan bahasa *assembly* yang disebut dengan *inline assembly*.

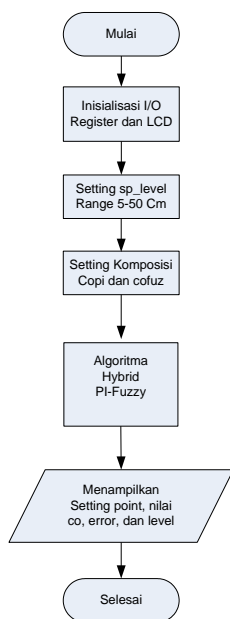
Penggunaan kompilator CodevisionAVR berdasarkan pertimbangan bahwa CodevisionAVR merupakan *cross-compiler* C

yang memiliki pustaka khusus (*library*) di samping pustaka C standar untuk beberapa aplikasi yang memudahkan pemrograman mikrokontroler, mudah dipelajari, memiliki terminal komunikasi antara mikrokontroler dengan PC, serta memiliki kemampuan untuk membuat program inisialisasi mikrokontroler melalui pembangkit program otomatis (*Automatic Program Generator*). Program monitoring untuk menampilkan respon sistem ke komputer digunakan program borland delphi 6.0. Secara umum perancangan perangkat lunak pada mikrokontroler ATmega 8535 terdiri atas :

- Perancangan Program Utama
- Program *Zero Crossing* dan sinyal pemicuan *Triac*.
- Program Kendali PI.
- Program Kendali Fuzzy.
- Program Hybrid Kendali PI - Fuzzy.

3.1 Perancangan Program Utama

Perancangan perangkat lunak program utama ini secara garis besar bertujuan untuk mengatur kerja sistem seperti inisialisasi *register* I/O dan variabel, *scanning keypad*, memilih kontroler, *setting* parameter, pembacaan hasil sensor, proses pengaturan sinyal kontrol proses penghitungan sinyal *error* dan *delta error*. Program utama berperan sebagai jantung perangkat lunak yang akan mengatur keseluruhan operasi yang melibatkan fungsi-fungsi pendukung. Fungsi-fungsi pendukung akan melakukan kerja khusus sesuai kebutuhan dari program utama. Diagram alir program utama dapat dilihat pada Gambar 9.



Gambar 9 Diagram alir program utama.

Diagram alir program utama pada Gambar 9 dapat dilihat bahwa program dimulai dengan melakukan inisialisasi I/O *port* dan inisialisasi *variable* pada mikrokontroler. Setelah inisialisasi, program akan ke *setting* *Sp_Level*. Masukkan nilai *setpoint* yang diinginkan dengan range 5-40 cm. Tombol *enter* pada *keypad* digunakan untuk mengeksekusi program yang akan membaca nilai referensi yang telah dimasukkan oleh operator dan nilai level yang terdeteksi oleh sensor, kemudian menghitung *error* dan *delta error* sebagai masukan logika Fuzzy, Nilai Konstanta *Kp* dan *Ki* dari kendali PI diperoleh dengan metode *Ziegler Nichols*. Penjumlahan sinyal kontrol (*co*) antara kendali PI dan kendali Fuzzy dengan variasi 5 komposisi yaitu *copi* = 10 % dan *cofuz* = 90 %, *copi* = 40 % dan *cofuz* = 60 %, *copi* = 50 % dan *cofuz* = 50 %, *copi* = 60 % dan *cofuz* = 40 %, *copi* = 90 % dan *cofuz* = 10 %.

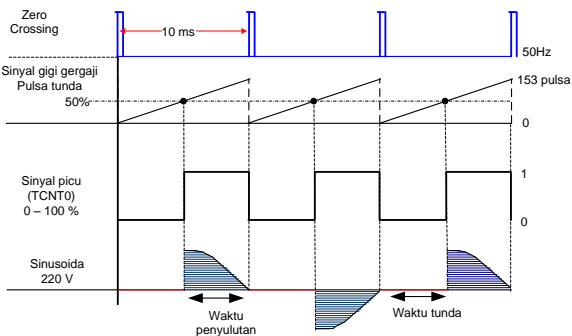
3.1.1 Program *Zero Crossing* dan Sinyal Pemicuan *Triac*

Pemicuan dilakukan setiap setengah siklus gelombang jala-jala arus AC dan dihitung dari titik persimpangan nol (*zero crossing detector*), untuk frekuensi jala-jala satu periode = 50 Hz maka untuk waktu setengah periode adalah $T = 10$ ms. Pemrograman dengan menggunakan fasilitas *delay* (ms) tidak efektif karena hanya memiliki 10 kemungkinan saja, sehingga diperlukan *timer/counter* sebagai pengganti *delay* (ms) ini. Mikrokontroler ATmega 8535 yang digunakan mempunyai *clock* maksimal sebesar 4 MHz, sehingga dalam penggunaan alokasi interupsi-interupsi pengendalian dapat di ubah sesuai dengan kebutuhan yang sudah disediakan oleh *compiler* di gunakan frekuensi interupsi sebesar 15625 Hz. Artinya pada 1 detik terdapat 15625 pulsa, sehingga untuk 10 ms terdapat 156 pulsa, dan cukup digunakan *timer/counter* 0 (*timer* 8 bit) yang dapat menjangkau hingga nilai 255 pulsa.

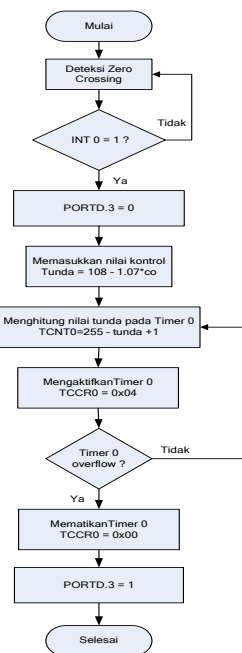
Program untuk pemicuan *triac* ini terdiri dari 2 buah rutin yaitu rutin *interupsi eksternal0* dan *interupsi timer0*. *Interupsi eksternal 0* bersumber dari masukan *zero crossing detector* yaitu dengan memberikan *logic low* pada *PortD.3*. Sinyal kontrol akan memicu *TCNT0*, dan mengaktifkan *timer 0*. Nilai yang dimasukkan ke *TCNT0* adalah $(255 - (Tunda+1))$, dimana *Tunda* adalah $(156-1.55*co)$ untuk *range* tegangan keluaran 0 - 220 V. Namun pada tugas akhir ini digunakan perhitungan tundaan yaitu $(108-1.07*co)$ dikarenakan karakteristik pompa yang digunakan *range* tegangannya adalah 140 - 220 V. Perhitungan tundaan diperoleh dengan cara melakukan percobaan. *Range* nilai kendali sinyal

kontrol adalah antara 0 – 100. Pada pengisian register TCCR0 dioperasikan OR dengan data 0x04 sebagai pertanda digunakannya frekuensi interupsi timer0 sebesar 15625 Hz.

Interupsi timer0 digunakan sebagai penunda pemicuan kendali tegangan AC beban, dimana pada PortD.3 dalam keadaan low kemudian diberi tunda sesuai dengan nilai TCNT0 yang diberikan ke timer0, setelah itu timer0 akan mengalami overflow, lalu timer0 dimatikan dimana TCCR0 di AND kan dengan data 0x00 dan mengaktifkan PortD.3 diberi logika high, kemudian kembali lagi ke interupsi external0 untuk mendeteksi Int0 apakah = 1 (sebagai pertanda jala-jala tegangan AC telah berada dipersimpangan titik nol). Bila int0 belum = 1, maka PortD.3 akan tetap high, setelah Int0 = 1 program akan dilanjutkan, kemudian PortD.3 diberi logika low, dan begitu seterusnya dilakukan secara berulang-ulang. Gambar 10 memperlihatkan urutan pemicuan tegangan I fase. Gambar 11 menunjukkan blok diagram proses pemicuan kendali tegangan AC.



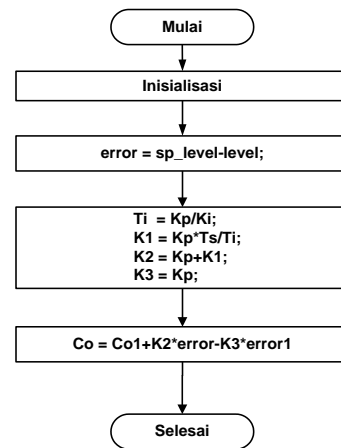
Gambar 10 Sinyal pemicuan tegangan I fase.



Gambar 11 Blok diagram sistem pemicuan terkendali.

3.1.2 Program Kendali PI

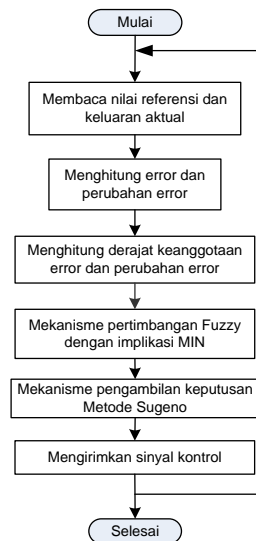
Pada program kendali PI konstanta Kp dan Ki diperoleh dengan metode ziegler Nichols. Perhitungan dengan algoritma PI dimulai dengan menghitung error yaitu selisih antara setpoint level dengan level sebenarnya. Error digunakan sebagai masukan pada kendali PI. Diagram alir proses pengontrolan dengan menggunakan algoritma kendali PI digital ditunjukkan dalam Gambar 12.



Gambar 12 Diagram alir algoritma PI.

3.1.3 Program Kendali Fuzzy

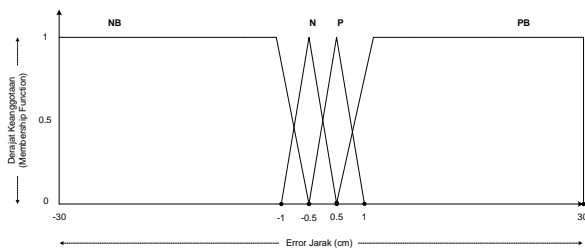
Program kendali Fuzzy terdiri atas Fuzzifikasi, evaluasi aturan, mekanisme pengambilan keputusan metode Sugeno. Keluaran pada proses mekanisme pengambilan keputusan metode Sugeno merupakan hasil dari proses kendali Fuzzy secara keseluruhan untuk mendapatkan nilai tegasnya yaitu sinyal kontrol. Blok diagram program kendali Fuzzy diperlihatkan pada Gambar 13.



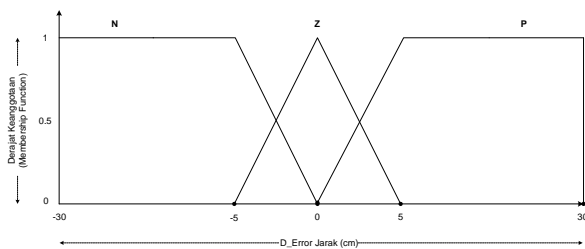
Gambar 13 Diagram blok utama sistem kendali Fuzzy.

Fuzzifikasi

Fuzzifikasi adalah proses pemetaan *input crisp* ke dalam himpunan-himpunan Fuzzy dalam bentuk fungsi keanggotaan. Tujuan dari Fuzzifikasi adalah untuk mendapatkan derajat keanggotaan dari hasil pemetaan *input crisp* ke dalam fungsi keanggotaan yang bersesuaian. Derajat keanggotaan bernilai antara 0 dan 1. Tahapan awal proses Fuzzifikasi adalah menentukan parameter-parameter fungsi keanggotaan pada setiap himpunan Fuzzy masukan. Pada pemrograman Fuzzifikasi ini digunakan parameter fungsi keanggotaan masukan berupa *error* dan *d_error*. Gambar 14 dan Gambar 15 merupakan parameter fungsi keanggotaan *error level* dan *d_error level*.



Gambar 14 Parameter fungsi keanggotaan *error level*.



Gambar 15 Parameter fungsi keanggotaan *d_error level*.

Evaluasi aturan

Tahapan kedua dalam *Fuzzy Inference System* adalah evaluasi aturan. Evaluasi aturan adalah proses mengevaluasi derajat keanggotaan tiap-tiap fungsi keanggotaan himpunan Fuzzy masukan ke dalam basis aturan yang telah ditetapkan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran Fuzzy. Himpunan Fuzzy keluaran digunakan dalam perancangan adalah *singleton*. Sebelum melakukan evaluasi aturan terlebih dahulu ditetapkan basis aturan. Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin. Secara lengkap, jumlah kombinasi yang mungkin dari dua himpunan Fuzzy masukan dengan masing-masing empat dan tiga fungsi keanggotaan sehingga jumlah aturannya

adalah dua belas aturan. Basis aturan yang dibuat berdasarkan tingkah laku *plant* yang diinginkan.

Pada bentuk yang lebih sederhana, dua belas aturan kendali Fuzzy dapat dilihat pada Tabel 3.

Tabel 3 Basis Aturan kendali Fuzzy.

D_Error / Error	N	Z	P
NB	KS	KS	KS
N	KS	KS	KS
P	K	K	K
PB	BS	BS	BS

Metode pengambilan keputusan yang digunakan dalam pemrograman ini adalah implikasi Min. Setelah semua aturan Fuzzy dieksekusi, dilakukan proses agregasi dengan mengambil nilai maksimal dari masing-masing fungsi keanggotaan variabel keluaran. Realisasi proses evaluasi aturan dalam bentuk program adalah sebagai berikut.

$$\begin{aligned} \omega_1 &= \min(\text{error_PB}, \text{derror_N}); \\ \omega_2 &= \min(\text{error_PB}, \text{derror_Z}); \\ \omega_3 &= \min(\text{error_PB}, \text{derror_P}); \end{aligned}$$

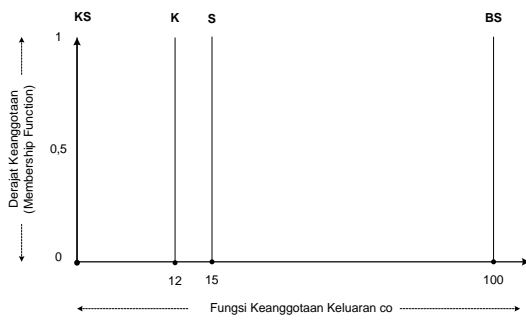
$$\begin{aligned} \omega_4 &= \min(\text{error_P}, \text{derror_N}); \\ \omega_5 &= \min(\text{error_P}, \text{derror_Z}); \\ \omega_6 &= \min(\text{error_P}, \text{derror_P}); \end{aligned}$$

$$\begin{aligned} \omega_7 &= \min(\text{error_N}, \text{derror_N}); \\ \omega_8 &= \min(\text{error_N}, \text{derror_Z}); \\ \omega_9 &= \min(\text{error_N}, \text{derror_P}); \end{aligned}$$

$$\begin{aligned} \omega_{10} &= \min(\text{error_NB}, \text{derror_N}); \\ \omega_{11} &= \min(\text{error_NB}, \text{derror_Z}); \\ \omega_{12} &= \min(\text{error_NB}, \text{derror_P}); \end{aligned}$$

Mekanisme Pengambilan Keputusan Metode Sugeno

Tahap terakhir dari inferensi Fuzzy adalah Mekanisme pengambilan keputusan metode Sugeno. Pada tahap ini adalah kebalikan dari proses Fuzzifikasi, yaitu mengubah himpunan Fuzzy keluaran menjadi keluaran tegas (*crisp*). Pengubahan ini diperlukan karena konstanta kendali PI hanya mengenal nilai tegas sebagai variabel sinyal kontrol. Perancangan ini, menggunakan sebuah himpunan Fuzzy keluaran dengan fungsi keanggotaannya berupa *singleton*, seperti tampak pada Gambar 16.



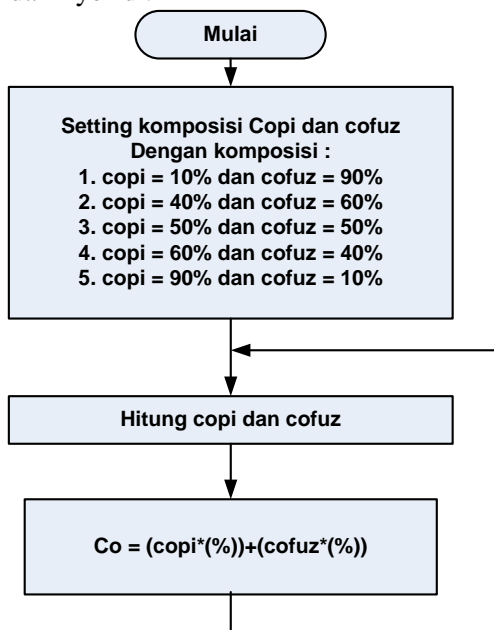
Gambar 16 Fungsi keanggotaan keluaran sinyal kontrol co.

Hasil keluaran *crisp output* akan merupakan sinyal kontrol, besarnya nilai sinyal kontrol akan mengaktifkan *register* TCNT0 pada kendali tegangan AC yang berbanding lurus dengan tegangan input pompa. Realisasi proses pengambilan keputusan metode sugeno menjadi bentuk *crisp output* dalam bentuk program adalah sebagai berikut :

$$\begin{aligned} \text{Cofuz} = & (\omega_1 * 100) + (\omega_2 * 100) + (\omega_3 * 100) + \\ & (\omega_4 * 12) + (\omega_5 * 12) + (\omega_6 * 12) + (\omega_7 * 0) + (\omega_8 * 0) + (\omega_9 * 0) + (\omega_{10} * 0) \\ & + (\omega_{11} * 0) + (\omega_{12} * 0) / (\omega_1 + \omega_2 + \omega_3 \\ & + \omega_4 + \omega_5 + \omega_6 + \omega_7 + \omega_8 + \omega_9 + \omega_{10} + \omega_{11} \\ & + \omega_{12}); \end{aligned}$$

3.1.4 Program Hybrid Kendali PI-Fuzzy

Pada sistem Hybrid kendali PI-Fuzzy sinyal kontrol yang dihasilkan merupakan penjumlahan sinyal kontrol kendali PI dan kendali Fuzzy. Gambar 17 merupakan diagram alir kendali hybrid :



Gambar 17 diagram alir kendali hybrid.

Pemrograman hybrid PI dan Fuzzy dimulai dengan pemberian komposisi sinyal kontrol dengan range 0-100 %. Pada tugas akhir ini di variasikan 5 komposisi sinyal kontrol yaitu copi = 10% dan cofuz = 90%, copi = 40% dan cofuz = 60%, copi = 50% dan cofuz = 50%, copi = 60% dan cofuz = 40%, copi = 90% dan cofuz = 10%. *Error* dan *delta error* merupakan input dari kendali hybrid, keluaran algoritma kendali hybrid menghasilkan sinyal kontrol yang merupakan nilai tundaan TCNT0 untuk meregulasi tegangan AC yang merupakan tegangan input pompa.

```
error=sp_level-level;
fuzzy();
Ti=Kp/Ki;
K1=Kp*Ts/Ti;
K2=Kp+K1;
K3=Kp;
copi=col+K2*error-K3*error1;
error1=error;
col=copi;
if (copi>100)
    copi=100;
    col=copi;
if ((copi<0) || (error<0))
    copi=0;
    col=copi;
co=((copi*komposisiPI)+(cofuz*komposisi fuz));
```

IV. PENGUJIAN DAN ANALISA

Pengujian pengendalian level ketinggian cairan dilakukan pada *plant* model tangki berbentuk tabung dengan ukuran jari-jari = 14 cm dan tinggi mendapatkan = 50 cm. Pengujian terdiri dari pengujian kalang terbuka yaitu untuk nilai L, T, dan a, pengujian perangkat lunak yaitu pengujian respon sistem dengan PC.

4.1 Pengujian Kalang Terbuka

Pengujian kalang terbuka pada pengendalian level untuk mengetahui tanggapan keluaran level ketinggian cairan pada *plant*. Tanggapan keluaran pada pengujian kalang terbuka akan memberi tahu kelayakan Ziegler-Nichols dalam mencari parameter proporsional dan integral $0,1 \leq L/T \leq 1$. Pada pengujian kalang terbuka, mikrokontroler memberi masukan berupa sinyal kendali Co ke pengontrol tegangan AC sebesar 100 %. Sinyal kendali tersebut akan mengendalikan level ketinggian cairan pada *plant* model. Tanggapan keluaran level ketinggian cairan pada *plant* ditunjukkan pada Gambar 18.



Gambar 18 Tanggapan keluaran sistem pengendalian level untuk kalang terbuka.

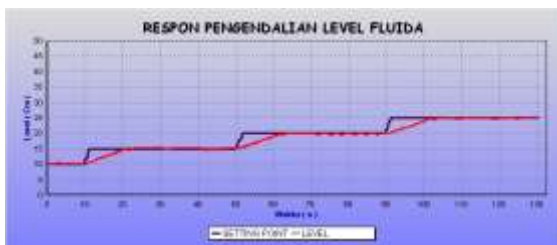
Tanggapan keluaran menunjukkan *plant* level memiliki waktu mati (*dead time*) L sebesar 5 detik dan konstanta waktu T sebesar 49 detik setelah pompa mendapatkan sinyal kontrol. Perbandingan besar waktu tunda terhadap konstanta waktu (L/T) menghasilkan nilai $5/49 = 0,102$. Karakteristik ini memenuhi persyaratan Ziegler-Nichols, sehingga dapat diperoleh nilai $K_P = 0,9/0,1 = 9$ dan $T_i = 3 \times 5 = 15$; $K_i = 9/15 = 0,6$.

4.2 Pengujian Respon Sistem dengan PC dengan melakukan Pengujian Program Hybrid Kendali PI dan Kendali Fuzzy

Pengujian ini dilakukan untuk mengetahui unjuk kerja pengendali pada sistem terhadap nilai referensi yang diberikan. Beberapa respon sistem yang diambil antara lain respon sistem tanpa gangguan dan respon sistem dengan gangguan.

Respon Sistem Hybrid Tanpa Gangguan

Kendali hybrid diuji pada 6 variasi komposisi sinyal kendali hybrid PI dan kendali Fuzzy yaitu : $\text{copi} = 10\%$ dan $\text{cofuz} = 90\%$; $\text{copi} = 40\%$ dan $\text{cofuz} = 60\%$; $\text{copi} = 50\%$ dan $\text{cofuz} = 50\%$; $\text{copi} = 60\%$ dan $\text{cofuz} = 40\%$; $\text{copi} = 90\%$ dan $\text{cofuz} = 10\%$ dengan memberikan nilai referensi level yang berbeda yaitu 15 cm, 20 cm, dan 25 cm. Respon sistem tanpa gangguan dapat dilihat pada Gambar 19, Gambar 20, Gambar 21, Gambar 22, dan Gambar 23 .



Gambar 19 Tanggapan sistem kendali level tanpa gangguan untuk $\text{copi} = 10\%$; $\text{cofuz} = 90\%$ dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 20 Tanggapan sistem kendali level tanpa gangguan untuk $\text{copi} = 40\%$; $\text{cofuz} = 60\%$ dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 21 Tanggapan sistem kendali level tanpa gangguan untuk $\text{copi} = 50\%$; $\text{cofuz} = 50\%$ dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 22 Tanggapan sistem kendali level tanpa gangguan untuk $\text{copi} = 60\%$; $\text{cofuz} = 40\%$ dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 23 Tanggapan sistem kendali level tanpa gangguan untuk $\text{copi} = 90\%$; $\text{cofuz} = 10\%$ dengan referensi 15 Cm, 20 cm, dan 25 cm.

Pada pengujian tanpa gangguan diperoleh grafik respon kendali hybrid yang terbaik yaitu dengan komposisi sinyal kendali $\text{copi} = 10\%$ dan $\text{cofuz} = 90\%$ pada referensi 15 cm respon sistem stabil terjadi kesalahan keadaan

tunak sebesar 0,5 cm, waktu naik (t_r) 10 detik. Pada referensi 20 Cm respon sistem stabil terjadi kesalahan keadaan tunak sebesar 0,5 cm, waktu naik (t_r) 11 detik. Pada referensi 25 Cm respon sistem stabil terjadi kesalahan keadaan tunak sebesar 0,5 cm, waktu naik (t_r) 12 detik.

Respon Sistem Hybrid dengan Gangguan sesaat

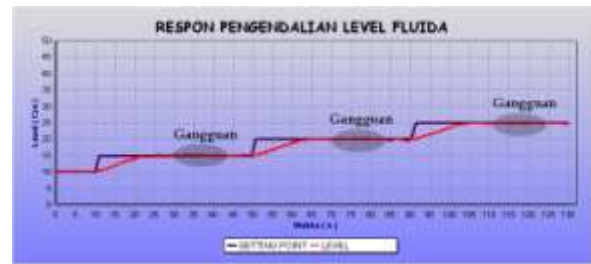
Kendali hybrid diuji pada 6 variasi komposisi sinyal kendali hybrid PI dan kendali Fuzzy yaitu : copi = 10% dan cofuz = 90% ; copi = 40% dan cofuz = 60% ; copi = 50% dan cofuz = 50% ; copi = 60% dan cofuz = 40% ; copi = 90% dan cofuz = 10% dengan memberikan Gangguan sesaat pada referensi level yang berbeda yaitu 15 cm, 20 cm, dan 25 cm. Gangguan sesaat yaitu dengan memperkecil laju alir outlet selama 10 detik. Respon sistem gangguan sesaat dapat dilihat pada Gambar 24, Gambar 25, Gambar 26, Gambar 27, dan Gambar 28.



Gambar 24 Tanggapan sistem kendali level gangguan sesaat untuk copi = 10%; cofuz = 90% dengan referensi 15 Cm, 20 cm, 25 cm.



Gambar 25 Tanggapan sistem kendali level dengan gangguan sesaat untuk copi = 40%; cofuz = 60% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 26 Tanggapan sistem kendali level dengan gangguan sesaat untuk copi = 50%; cofuz = 50% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 27 Tanggapan sistem kendali level dengan gangguan sesaat untuk copi = 60%; cofuz = 40% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 28 Tanggapan sistem kendali level dengan gangguan sesaat untuk copi = 90%; cofuz = 10% dengan referensi 15 Cm, 20 cm, dan 25 cm.

Pada pengujian gangguan sesaat dengan grafik respon sistem kendali hybrid diperoleh hasil pengamatan untuk ke lima variasi sinyal kendali mengalami waktu *recovery* yang baik ketika mengalami gangguan sesaat. Hal ini disebabkan perubahan aliran keluaran sesaat tidak mempengaruhi kinerja kendali hybrid yang memberikan sinyal kontrol pada kendali tegangan AC.

Respon Sistem Hybrid dengan Gangguan Kontinyu

Kendali hybrid diuji pada 6 variasi komposisi sinyal kendali hybrid PI dan kendali Fuzzy yaitu : copi = 10% dan cofuz =

90% ; copi = 40% dan cofuz = 60% ; copi = 50% dan cofuz = 50% ; copi = 60% dan cofuz = 40% ; copi = 90% dan cofuz = 10% dengan memberikan gangguan kontinyu pada referensi level yang berbeda yaitu 15 cm, 20 cm, dan 25 cm. Gangguan kontinyu yaitu dengan memperbesar laju alir outlet. Respon sistem gangguan sesaat dapat dilihat pada Gambar 29, Gambar 30, Gambar 31, Gambar 32, dan Gambar 33.



Gambar 29 Tanggapan sistem kendali level gangguan kontinyu untuk copi = 10%; cofuz = 90% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 30 Tanggapan sistem kendali level dengan gangguan kontinyu untuk copi = 40%; cofuz = 60% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 31 Tanggapan sistem kendali level dengan gangguan kontinyu untuk copi = 50%; cofuz = 50% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 32 Tanggapan sistem kendali level dengan gangguan kontinyu untuk copi = 60%; cofuz = 40% dengan referensi 15 Cm, 20 cm, dan 25 cm.



Gambar 33 Tanggapan sistem kendali level dengan gangguan kontinyu untuk copi = 90%; cofuz = 10% dengan referensi 15 Cm, 20 cm, dan 25 cm.

Pada pengujian dengan gangguan kontinyu diperoleh grafik respon kendali hybrid yang terbaik yaitu dengan komposisi sinyal kontrol copi = 10% dan cofuz = 90% dengan referensi 15 cm respon sistem stabil terjadi kesalahan keadaan tunak sebesar 0,5 cm, waktu naik (t_r) 28 detik. Pada referensi 20 Cm respon sistem stabil terjadi kesalahan keadaan tunak sebesar 0,5 cm, waktu naik (t_r) 46 detik. Pada referensi 25 Cm respon sistem stabil terjadi kesalahan keadaan tunak sebesar 0,5 cm, waktu naik (t_r) 76 detik.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan didapatkan hal-hal penting sebagai berikut:

1. Pada pengujian hybrid kendali PI dan kendali Fuzzy tanpa gangguan diperoleh respon sistem yang terbaik yaitu pada komposisi copi = 10% dan cofuz = 90% dengan waktu naik (t_r) = 10 detik dan *error offset* = 0,5 cm untuk referensi 15 cm, (t_r) = 11 detik dan *error offset* = 0,5 cm untuk referensi 20 cm, (t_r) = 11 detik dan *error offset* = 0,5 cm untuk referensi

- 25 cm dan kesalahan keadaan tunak tidak melebihi dari $0.05 \times$ referensi.
2. Pada pengujian dengan gangguan sesaat, kendali hybrid tidak mengalami gangguan kinerja yang signifikan. Lima komposisi yang diberikan mampu bekerja dengan baik terhadap gangguan.
 3. Pada pengujian hybrid kendali PI dan kendali Fuzzy dengan gangguan kontinu diperoleh respon sistem yang terbaik yaitu pada komposisi copi = 10% dan cofuz = 90% dengan waktu naik (t_r) = 28 detik dan *error offset* = 0,5 cm untuk referensi 15 cm, (t_r) = 44 detik dan *error offset* = 0,5 cm untuk referensi 20 cm, (t_r) = 76 detik dan *error offset* = 0,5 cm untuk referensi 25 cm dan kesalahan keadaan tunak tidak melebihi dari $0.05 \times$ referensi.
 4. Nilai sinyal kendali co adalah penjumlahan sinyal kendali PI (copi) dan sinyal kendali Fuzzy (cofuz).

5.2 Saran

Pada pengembangan sistem lebih lanjut ada beberapa saran yang dapat dilakukan yaitu sebagai berikut:

1. Dapat dilakukan pengendalian level cairan dengan metode *gain scheduling*, adaptif STR (*Self Tuning Regulator*), dan jaringan saraf tiruan.

DAFTAR PUSTAKA

- (1) Astrom, John and Bjorn Wittenmark, *Adaptive Control Second Edition*, Addison-Wesley Publishing Company Inc, 1995.
- (2) Benjamin C. Kuo, *Automatic Control System Sixth Edition*, Prentice-Hall, New Jersey, 1991.
- (3) Brosilow, Coleman and Babu Joseph, *Techniques of Model-Based Control*, Prentice Hall International Series, New Jersey, 2001.
- (4) Budiharto Widodo, *Panduan Praktikum Mikrokontroler AVR ATmega16*, PT Elex Media Komputindo, Jakarta, 2008.
- (5) Gunterus, Frans, *Falsafah Dasar: Sistem Pengendalian Proses*, PT Elex Media Komputindo, Jakarta, 1997.
- (6) Heryanto, M. Ary dan Wisnu Adi P, *Pemrograman Bahasa C Untuk Mikrokontroler AT MEGA 8535*, Penerbit Andi, Yogyakarta, 2008.
- (7) Indra, Bakhtiar K, *Tugas Akhir: Pengendali Pempa dan Monitoring Ketinggian Air Jarak*

- Jauh Menggunakan Komputer dan Mikrokontroler AT89S51*, Diploma III Instrumentasi dan Elektronika, Universitas Diponegoro, Semarang.
- (8) Jamshidi, Mohammad, *Application of Fuzzy Logic*, Prentice-Hall International Inc, New Jersey, 1980.
 - (9) Kurniawan, David, *Kontrol Mobile Robot Penjejak Objek Bergerak Berbasis Logika Fuzzy*, Teknik Elektro, Universitas Diponegoro, Semarang.
 - (10) Malvino. "Prinsip – Prinsip Elektronika". Jakarta : Erlangga, 1996.
 - (11) Mashuri, Ahmad, *Perancangan Sistem Pengendalian Suhu Dan Akuisisi Data Tingkat Kelembaban Pada Mesin Pengereng Kertas Berbasis Kendali Logika Fuzzy*, Teknik Elektro, Universitas Diponegoro, Semarang.
 - (12) Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 1*, diterjemahkan oleh Edi Leksono, Erlangga, Jakarta, 1994.
 - (13) Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 2*, diterjemahkan oleh Edi Leksono, Erlangga, Jakarta, 1994.
 - (14) Syahadi, Mohamad, *Skripsi : Aplikasi Kontrol Proporsional Integral Berbasis Mikrokontroler Atmega 8535 Untuk Pengaturan Suhu Pada Alat Pengereng Kertas*, Teknik Elektro, Universitas Diponegoro, Semarang.
 - (15) Wardhana L, *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*, Penerbit Andi, Yogyakarta, 2006.
 - (16) -----, *ATmega8535 Data Sheet*, <http://www.atmel.com>.

Hani Purwanti (L2F 306 029)



Lahir di Semarang, 8 Januari 1976. Saat ini sedang melanjutkan studi pendidikan strata I di Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro Konsentrasi Kontrol.

Mengetahui dan mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Wahyudi, ST, MT
NIP.

196906121994031001
Tanggal: _____

Iwan Setiawan, ST, MT
NIP.

197009262000121001
Tanggal: _____