

Makalah Seminar Tugas Akhir

Tuning Parameter Proporsional – Integral dengan *FL* (Fuzzy Logic) untuk Pengaturan Level Air Berbasis Mikrokontroler Atmega 8535

Eduward Tigor H.L.^[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

Metode kontrol PI (Proporsional-Integral) merupakan sebuah metode kontrol yang banyak diterapkan di bidang industri sampai saat ini. Kontroler ini memiliki parameter-parameter pengontrol, yaitu konstanta proporsional (Kp) dan konstanta integral (Ki). Pada metode PI konvensional, kedua parameter tersebut diturunkan dari perhitungan matematis. Kesulitan penalaan parameter-parameter tersebut akan ditemui bila plant yang dikendalikan adalah sistem dengan orde tinggi. Maka dari itu, diperlukan suatu metode penala PI yang lebih baik dan dapat diterapkan dalam sistem orde tinggi.

Metode penalaan PI dengan logika fuzzy dapat dilakukan secara otomatis tanpa melakukan pemodelan sistem. Pada metode ini nilai Kp dan Ki ditentukan berdasarkan sejumlah aturan-aturan yang ditetapkan oleh Logika fuzzy. Input fuzzy berupa error dan delta error .

Plant pengendalian level ketinggian cairan menggunakan mikrokontroler ATmega8535 sebagai kontroler dan pompa sebagai aktuator. Pada pengujian didapatkan hasil penalaan parameter PI yang mampu menghasilkan tanggapan keluaran dengan rise time yang cepat, selisih tunak yang kecil, overshoot yang sangat kecil, dan ketika sistem diberi gangguan, tanggapan keluaran akan tetap terjaga dalam kestabilan.

Kata kunci : *PI (Proporsional-Integral), metode Penalaan PI dengan logika fuzzy, 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, sedangkan permasalahan dalam proses industri kimia yaitu proses pemisahan cairan (destilasi) yang membutuhkan pengaturan level cairan didalam destilator, sering terjadi perubahan (deviasi) laju aliran masukan yang disebabkan perubahan tekanan aliran inlet atau juga disebabkan oleh adanya gaya gesek pada pipa saluran maka akan mengakibatkan perubahan debit masukan yang 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 membutuhkan pengetahuan tentang parameter-parameter sistem terlebih dahulu, selain itu untuk sistem yang kompleks terdapat kendala dalam menentukan parameter-parameter yang sesuai agar mendapatkan respon keluaran yang stabil. Logika Fuzzy dapat digunakan untuk menala parameter-parameter kendali konvensional seperti kendali PI, logika Fuzzy menerapkan suatu sistem kemampuan manusia untuk mengendalikan sesuatu, yaitu dalam bentuk aturan-aturan Jika – Maka (*If – Then Rules*), sehingga proses pengendalian akan mengikuti pendekatan secara linguistik. Pada tugas akhir ini dirancang suatu sistem pengendalian level cairan dengan memanfaatkan mikrokontroler ATmega 8535 sebagai kontroler, dan logika Fuzzy sebagai metode penala konstanta kendali PI dengan tujuan mendapatkan respon sistem yang baik yaitu *rise time* yang cepat dan *overshoot* yang minimal.

1.2 Tujuan

Tujuan dari Tugas Akhir ini adalah penalaan konstanta kendali PI menggunakan logika Fuzzy berbasis mikrokontroler ATmega

¹ Mahasiswa Jurusan Teknik Elektro UNDIP

² Staf Pengajar Jurusan Teknik Elektro UNDIP

8535 untuk mengatur ketinggian cairan pada tangki penampung.

1.3 Pembatasan Masalah

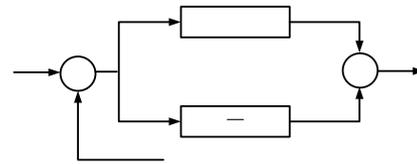
Pembatasan masalah pada tugas akhir ini sebagai berikut :

1. *Plant* yang diuji adalah penampungan cairan dengan bentuk selinder.
2. Aksi pengontrolan PI mengaktifkan kendali tegangan AC yang digunakan untuk menggerakkan pompa.
3. Penalaan dengan sistem pengendali Fuzzy metode Sugeno, fungsi keanggotaan masukan himpunan Fuzzy adalah segitiga dan trapesium, metode pengambilan keputusan menggunakan fungsi implikasi MIN dengan agregator MAX, dan defuzzifikasi menggunakan metode rata-rata terbobot (*weighted average*).
4. Pengujian dilakukan pada perubahan referensi ketinggian dan pemberian gangguan.
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. Dalam Tugas Akhir ini pemrograman monitoring tidak dibahas.

II. DASAR TEORI

2.1 Kendali PI (*Proporsional-Integral*)^[2]

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.

$$co(s) = K_p E(s) + \frac{K_p}{T_i s} E(s)$$

$$sco(s) = K_p sE(s) + \frac{K_p}{T_i} E(s) \quad (1)$$

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:

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

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

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

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

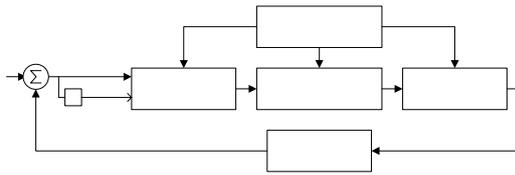
$$K_2 = (K_p + K_1) \Rightarrow K_1 = \frac{K_p T}{T_i} \quad (5)$$

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

2.2 Sistem Logika Fuzzy^[11]

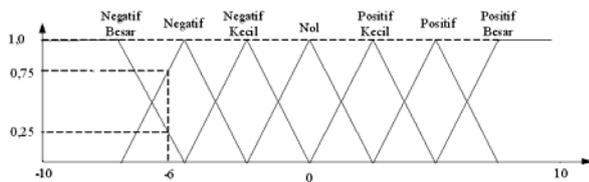
Fuzzy berarti samar, kabur atau tidak jelas. *Fuzzy* adalah istilah yang dipakai oleh Lotfi A Zadeh pada bulan Juli 1964 untuk menyatakan kelompok / himpunan yang dapat dibedakan dengan himpunan lain berdasarkan derajat keanggotaan dengan batasan yang tidak begitu jelas (samar), tidak seperti himpunan klasik yang

membedakan keanggotaan himpunan menjadi dua, himpunan anggota atau bukan anggota. Gambar 2 menunjukkan mekanisme kendali logika Fuzzy.



Gambar 2 Mekanisme kendali logika Fuzzy kalang tertutup.

Kendali logika Fuzzy dilakukan dalam tiga tahap, yaitu fuzzifikasi, evaluasi aturan dan defuzzifikasi. Komponen Fuzifikasi berfungsi untuk memetakan masukan data tegas ke dalam himpunan Fuzzy menjadi nilai kesalahy dari beberapa variabel linguistik masukan. Gambar 3 menunjukkan proses Fuzzyfikasi



Gambar 3 Proses Fuzzyfikasi.

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

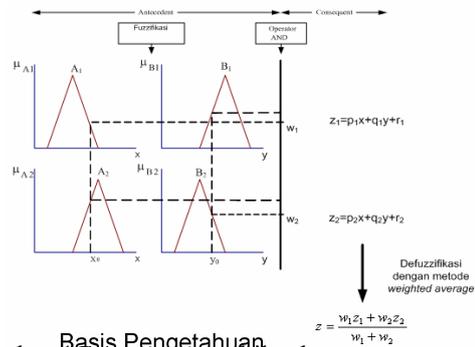
Basis Aturan Fuzzy merupakan kumpulan pernyataan aturan 'IF-THEN' yang didasarkan kepada pengetahuan pakar.

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

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

Defuzzifikasi dapat didefinisikan sebagai proses perubahan besaran Fuzzy yang disajikan

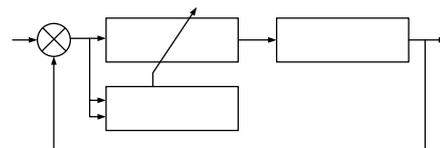
dalam bentuk himpunan-himpunan Fuzzy keluaran dengan fungsi keanggotaan untuk mendapatkan kembali bentuk tegasnya (*crisp*). Gambar 4 menunjukkan proses pengambilan keputusan metode sugeno.



Gambar 4 Proses pengambilan keputusan metode Sugeno.

2.3 Metode Pengambilan Keputusan Fuzzy Multiterm Controllers^[8]

Pada metode Fuzzy Multiterm Controllers, logika Fuzzy akan dipergunakan untuk menala atau mengadaptasi sebuah kontroler *multiterm* seperti kontroler PD, PI atau PID seperti yang terlihat pada Gambar 5.



Gambar 5 Kontroler PI yang diadaptasi oleh fuzzy sebagai tuning.

Kontroler PI dapat direpresentasikan oleh salah satu dari dua bentuk berikut yaitu

1. Bentuk Kontinyu :

$$co(t) = Kp.e(t) + Ki \int e(t)dt$$

2. Bentuk Diskrit :

$$co(k) = Kp.e(k) + KiTs \sum_{i=1}^n e(i)$$

Dimana Kp dan Ki masing-masing adalah konstanta Proporsional, dan konstanta Intergral, Nilai $Ti = Kp/Ki$, sering dikenal sebagai konstanta waktu integral, $\Delta e(k) = e(k)-e(k-1)$, Ts adalah periode sampling dan n adalah jumlah sampel data. Pendekatan dasar dalam mengadaptasi kontroler *multiterm* (kontroler PI) dengan menggunakan logika Fuzzy adalah dengan merencanakan *supervisory rules* dari Fuzzy tersebut dengan aturan (*rule*) persamaan 8 :

- IF $e(k)$ is Ai and $\Delta e(k)$ is Bi , THEN Kp is Ci and Ki is Di (8)

Ai , Bi , Ci , dan Di adalah variabel linguistik dari *rule ke-I* ($i = 1, 2, \dots, m$) yang dapat direpresentasikan dengan fungsi

keanggotaan tertentu.

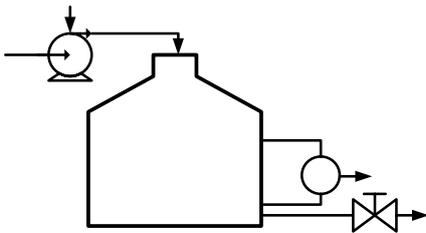
2.4 Model Kontrol Proses Level Fluida^[3]

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.

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

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.

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

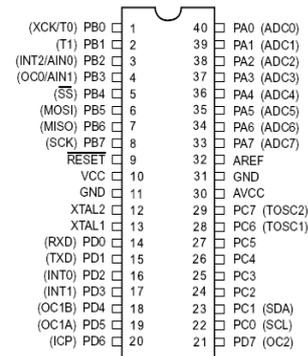
Atau dapat ditulis dalam bentuk umum .

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

2.5 Mikrokontroler AVR ATmega 8535^[9]

Mikrokontroler AVR ATmega 8535 telah didukung penuh dengan program dan sarana pengembangan seperti: kompilervkompilerv C,

simulator program, emulator dalam rangkaian, dan *kit* evaluasi. Konfigurasi pin terlihat pada Gambar 7.

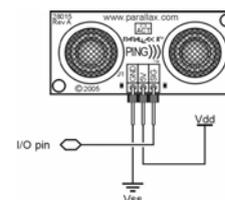


Gambar 7 Konfigurasi pin ATmega 8535.

Mikrokontroler AVR ATmega 8535 memiliki ruang pengalamatan memori data dan memori program terpisah. Memori program yang terletak dalam *Flash PEROM* tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. Selain itu, mikrokontroler ini memiliki *4KByteX16-bit Flash PEROM* dan memiliki 12-bit *Program Counter* (PC) sehingga mampu mengalami isi *Flash*.

2.6 Sensor Ultrasonik^[9]

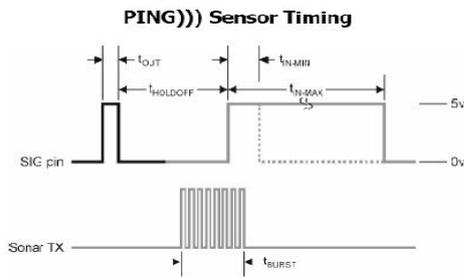
Sensor yang digunakan pada Tugas Akhir ini merupakan sebuah sensor ultrasonik buatan Parallax (*Sensor PING™ Ultrasonik Range Finder*). Bentuk visual sensor yang digunakan dalam Tugas Akhir ini ditunjukkan pada Gambar 8. Sensor PING™ ini dapat mengukur tinggi muka cairan.



Gambar 8 Sensor PING Ultrasonik Range Finder.

Berdasarkan *datasheet*, modul PING™ ini dapat digunakan untuk mengukur benda sejauh 3 cm sampai 300 cm. Pada sistem kendali ini, rentang level yang digunakan adalah 5 cm sampai 40 cm.

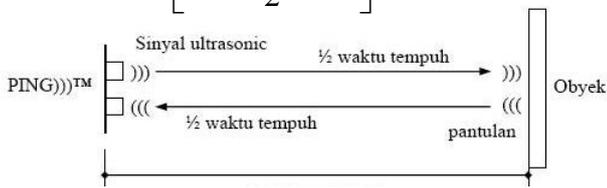
Sensor PING™ mendeteksi ketinggian pelampung dengan cara memancarkan gelombang ultrasonik (40 KHz) selama waktu pemancaran (200 μs) kemudian mendeteksi pantulannya. Gambar 9 menunjukkan *timing diagram* dari sensor PING™.



Gambar 9 Timing Diagram Sensor PING Ultrasonik Range Finder.

Waktu tempuh gelombang ultrasonik untuk ditunjukkan pada Gambar 10. Maka level yang diukur dapat dirumuskan sebagai berikut :

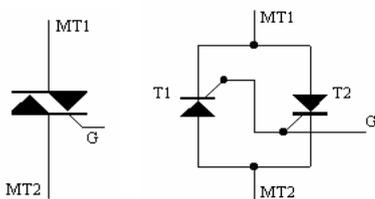
$$\text{level} = 45 - \left[\frac{t_{IN} \times 344 \text{ m/s}}{2} \right] \text{ meter}$$



Gambar 10 Ilustrasi cara kerja sensor PING.

2.7 TRIAC^[10]

Pengendalian fasa pada pengontrol tegangan AC, umumnya digunakan *bidirectional triode thyristor* atau yang biasa dikenal sebagai *triac*. *Triac* dapat bersifat konduktif dalam dua arah, *triac* dapat dianggap sebagai dua buah *thyristor* tersambung secara antiparalel dengan koneksi gerbang seperti ditunjukkan pada Gambar 11b. *Triac* merupakan komponen *bidirectional*, terminalnya tidak dapat ditentukan sebagai anode/katode. Koneksi-koneksinya diberi nama *main terminal 1* (MT1), *main terminal 2* (MT2) dan *gate* atau gerbang (G). Simbol untuk *triac* ditunjukkan pada Gambar 11a.

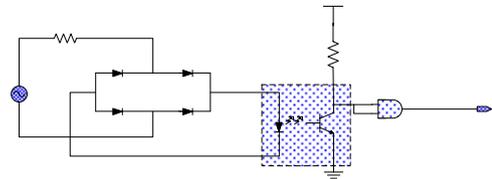


(a) Simbol Triac. (b) Rangkaian ekuivalen triac.
Gambar 11 Komponen *triac*.

2.8 Zero Crossing Detector^[14]

Penentuan waktu tunda dengan tepat agar mendapatkan hasil pengaturan daya yang akurat, mikrokontroler harus mengetahui saat titik nol (*zero crossing*). *Zero crossing detector* adalah rangkaian yang digunakan untuk mendeteksi

gelombang sinus AC 220 volt saat melewati titik tegangan nol. Seberangan titik nol yang dideteksi adalah peralihan dari positif menuju negatif dan peralihan dari negatif menuju positif. Seberangan-seberangan titik nol ini merupakan acuan yang digunakan sebagai awal pemberian nilai waktu tunda untuk pemicuan *triac*. Rangkaian *zero crossing detector* ditunjukkan pada Gambar 12.



Gambar 12 Rangkaian *zero crossing detector*.

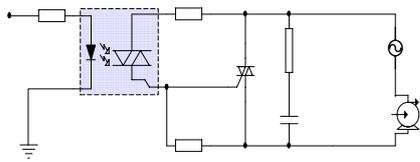
2.9 Kendali tegangan AC^[11]

Rangkaian pengendali tegangan AC merupakan rangkaian untuk pengendali tegangan AC gelombang penuh satu fasa seperti ditunjukkan pada Gambar 2.27. Komponen utama pada rangkaian ini adalah *triac* BT 139. *Triac* berhubungan langsung dengan beban yaitu pompa dan tegangan jala-jala listrik 220 volt. *Triac* BT 139 mempunyai nilai arus rms maksimal sebesar 16 A dan tegangan maksimal sebesar 800 VAC.

Pada perancangan ini sinyal kontrol untuk pemicuan *triac* adalah berasal dari mikrokontroler yang mempunyai level tegangan rendah. Sebagai pengaman agar tidak membahayakan bagian yang mempunyai level tegangan rendah (bagian pengontrol), maka digunakan IC MOC 3021 untuk mengkopel *triac* dengan bagian pengontrolnya. IC MOC 3021 merupakan optoisolator dengan *triac* output.

MOC 3021 terdiri dari 2 komponen utama, yaitu *LED* dan *photodiode*. *LED* berfungsi untuk mengubah energi listrik menjadi energi cahaya. *LED* akan memancarkan cahaya berupa pulsa-pulsa yang merupakan refleksi dari tegangan yang berbentuk kotak (*high dan low*) yang berasal dari mikrokontroler untuk mengendalikan pompa. Sinyal kotak ini akan diubah oleh *LED* menjadi sederetan pulsa dalam bentuk cahaya. *Photodiode* berfungsi sebagai penerima dari cahaya keluaran *LED*, *Photodiode* akan mengubahnya menjadi energi listrik yang akan mengendalikan *triac* akibatnya *triac* BT139 akan terpicu sehingga pompa (pump) akan teraliri arus listrik. Diaturnya waktu pemberian sinyal pemicuan maka besarnya tegangan yang diterima pompa juga akan bervariasi. Keuntungan penggunaan rangkaian ini adalah lebih terjaminnya keamanan rangkaian pengendali dari pengaruh jala-jala listrik. Hal ini disebabkan terpisahnya aliran arus antara beban dengan

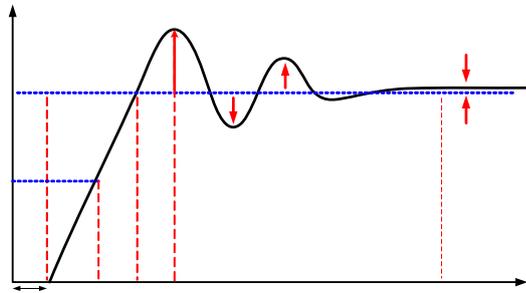
rangkaian pengendali oleh penggandeng cahaya didalam MOC3021.



Gambar 2.13 Rangkaian pengontrol tegangan AC.

2.10 Respon Sistem^[11]

Tanggapan transien suatu sistem kontrol secara praktek selalu menunjukkan osilasi teredam sebelum mencapai keadaan tunaknya, dalam menggolongkan karakteristik tanggapan transien suatu sistem kontrol terhadap sinyal masukan, umumnya dikelompokkan seperti ditunjukkan pada Gambar 14.



Gambar 14 Kurva respon transien sistem kontrol.

- Waktu Respon (*Dead Time*) adalah waktu yang diperlukan oleh sistem untuk memberikan respon terhadap sinyal masukan yang diberikan.
- Waktu Tunda (*Delay Time*) adalah waktu yang diperlukan oleh suatu sistem kontrol untuk mencapai setengah dari nilai referensi yang telah ditetapkan sebelumnya.
- Waktu Naik (*Rise Time*) adalah waktu yang diperlukan oleh suatu sistem kontrol untuk mencapai nilai referensi yang telah ditetapkan sebelumnya.
- Waktu Puncak (*Peak Time*) adalah waktu yang diperlukan oleh suatu sistem kontrol untuk mencapai puncak pertama *overshoot*.
- Waktu Stabil (*Steady State Time*) adalah waktu yang diperlukan sistem kontrol untuk mencapai kondisi tunaknya (*steady state*).
- *Offset* biasanya digunakan toleransi sekitar $\pm 5\%$ dari nilai referensi yang telah ditetapkan.

$$\text{Offset} = 0,05 \times \text{referensi}$$

- Lonjakan (*Overshoot*) adalah amplitudo (lonjakan)_{0,5} yang diberikan suatu sistem kontrol terhadap referensi (*set point*) yang

Td
Delay Time

Undershoot

Ts
Steady State Time

diberikan. Nilai *overshoot* dihitung dalam persen lonjakan maksimum yang didefinisikan :

$$\text{Overshoot} = \frac{C_{maks} - C_{reference}}{C_{reference}} \times 100\%$$

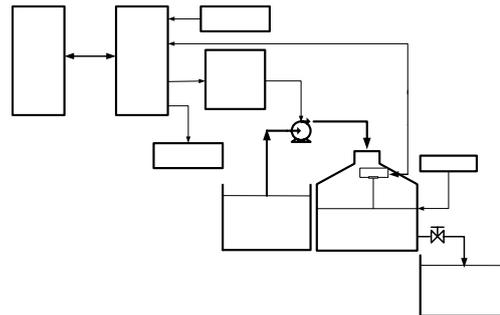
- Rasio Redaman (*Decay Ratio*) adalah nilai perbandingan antara *overshoot* kedua dengan *overshoot* pertama yang didefinisikan sebagai berikut :

$$\text{Decay Ratio} = \frac{A_3}{A_1}$$

III. PERANCANGAN

3.1 Perancangan Perangkat Keras

Blok rancangan perangkat keras sistem pengatur level cairan pada Tugas Akhir ini dapat dilihat pada Gambar 15. Perancangan perangkat keras meliputi mikrokontroler sebagai pengendali, kendali tegangan AC sebagai pengatur tegangan input pompa, pompa sebagai penggerak, unit masukan, perancangan *display*.

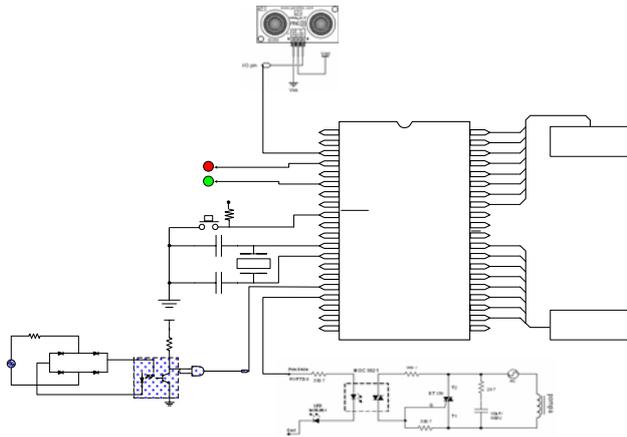


Gambar 15 Blok diagram sistem.

3.1.1 Sistem Minimum AVR ATmega 8535

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 *clock* input 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 Gambar 16.

A3



Gambar 16 Alokasi port pada sistem minimum mikrokontroler ATmega 8535.

3.2 Perancangan Perangkat Lunak (Software)

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*.

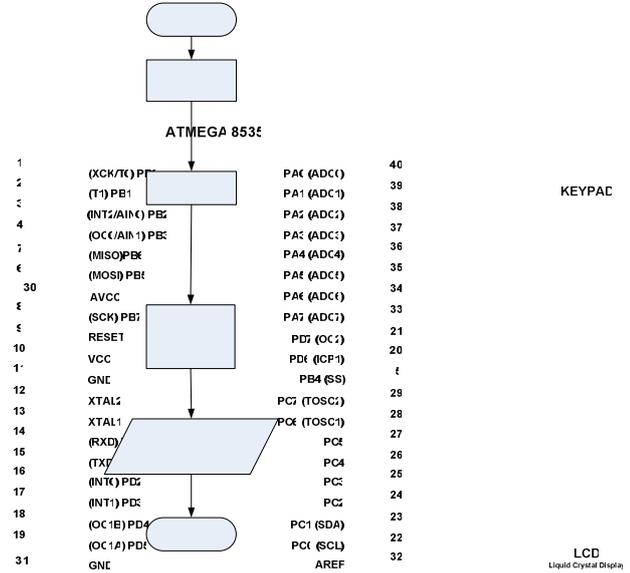
Penggunaan kompilator CodevisionAVR berdasarkan pertimbangan bahwa CodevisionAVR merupakan *cross-compiler* C yang memiliki pustaka khusus (*library*) di samping pustaka C standar untuk beberapa aplikasi yang memudahkan pemrograman mikrokontroler, mudah dipelajari, memiliki terminal komunikasi antara 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 Fuzzy sebagai *tuning* kendali PI.
- Program kendali PI.

3.2.1 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, seting parameter, pembacaan hasil sensor, proses pengaturan sinyal kontrol proses penghitungan sinyal *error* dan *delta error*. Program utama berperan sebagai jantung

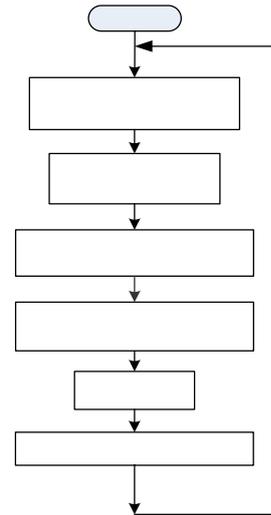
perangkat lunak yang akan mengatur keseluruhan operasi yang melibatkan fungsi-fungsi pendukung. Fungsi-fungsi pendukung akan melakukan kerja khusus sesuai kebutuhan dari program utama. Diagram alir program utama dapat dilihat pada Gambar 17.



Gambar 17 Diagram Alir Program Utama.

3.2.1 Program Fuzzy Sebagai Tuning Kendali PI.

Program kendali Fuzzy terdiri atas Fuzzifikasi, evaluasi aturan, mekanisme pengambilan keputusan, dan defuzzifikasi. Keluaran pada proses defuzzifikasi merupakan hasil dari proses sistem Fuzzy secara keseluruhan. Blok diagram program Fuzzy sebagai *tuning* kendali PI diperlihatkan pada Gambar 18.

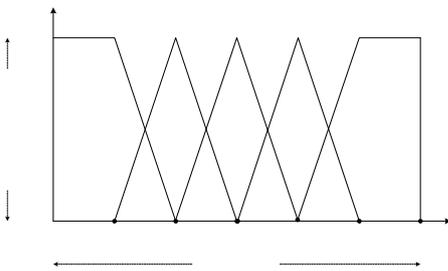


Gambar 18 Diagram blok utama sistem Fuzzy.

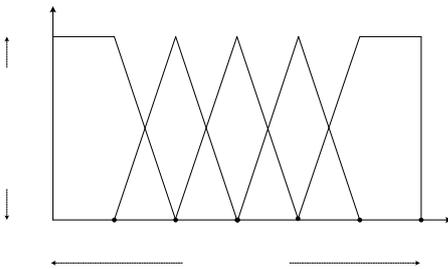
A. Fuzzifikasi

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*. Derajat keanggotaan bernilai antara 0 dan 1, sedangkan dalam pemrograman *tuning* dengan logika Fuzzy ini, nilai derajat keanggotaan dinormalisasi ke dalam nilai 0 hingga 200 dengan tujuan menghemat memori *flash* pada mikrokontroler. Seluruh proses perancangan kendali logika Fuzzy dilakukan dengan menggunakan mikrokontroler ATmega 8535, sehingga seluruh parameter fungsi keanggotaan *error* dan *d_error* pada gambar di atas dinormalisasi ke dalam bahasa pemrograman C yang dapat dilihat pada Gambar 19 dan Gambar 20.



Gambar 19 Fungsi keanggotaan *error* level.



Gambar 20 Fungsi keanggotaan *d_error* level.

B. Evaluasi aturan

Tahapan kedua dalam *Fuzzy Inference System* adalah evaluasi aturan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran Fuzzy. Himpunan Fuzzy keluaran yang digunakan dalam perancangan *tuning* parameter PI 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 lima fungsi keanggotaan adalah dua puluh lima aturan. Basis aturan yang dibuat berdasarkan tingkah laku plant yang diinginkan. Keluaran Fuzzy akan menentukan nilai konstanta PI. Dua puluh lima aturan yang digunakan terlihat pada Tabel 1 dan Tabel 2.

Tabel 1 Basis Aturan tuning PI dengan Logika Fuzzy untuk Konstanta K_p .

D. Error / Error	NB	N	Z	P	PB
NB	B	B	B	B	B
N	K	S	S	S	K
Z	KS	KS	KS	KS	KS
P	K	S	S	S	K
PB	B	B	B	B	B

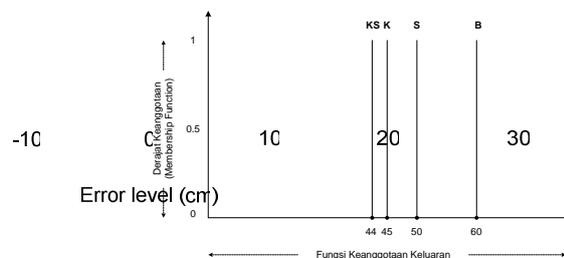
Tabel 2 Basis Aturan tuning PI dengan Logika Fuzzy untuk Konstanta K_i .

D. Error / Error	NB	N	Z	P	PB
NB	B	B	B	B	B
N	K	S	S	S	K
Z	KS	KS	KS	KS	KS
P	K	S	S	S	K
PB	B	B	B	B	B

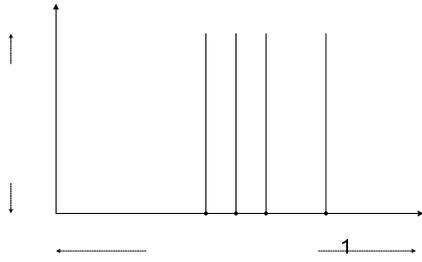
Metode pengambilan keputusan (inferensi) yang digunakan dalam pemrograman ini adalah metode Max-Min. Setelah semua aturan fuzzy dieksekusi, dilakukan proses agregasi dengan mengambil nilai maksimal dari masing-masing fungsi keanggotaan variabel keluaran.

C. Defuzzifikasi

Tahap terakhir dari inferensi Fuzzy adalah defuzzifikasi. Defuzzifikasi merupakan 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 parameter. Perancangan ini, menggunakan sebuah himpunan Fuzzy keluaran dengan fungsi keanggotaannya berupa *singleton*, seperti tampak pada Gambar 21 dan Gambar 22.



Gambar 21 Fungsi keanggotaan keluaran konstanta K_p .

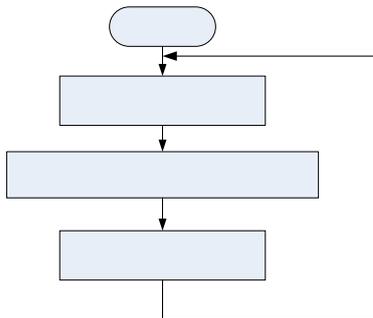


Gambar 22 Fungsi keanggotaan keluaran konstanta K_i .

Nilai tegas (*crisp*) keluaran diperoleh dari himpunan-himpunan Fuzzy dengan menggunakan metode rata-rata.

3.2.2 Program Kendali PI.

Pada tugas akhir ini digunakan kendali PI untuk mendapatkan tinggi muka cairan yang diinginkan. Nilai konstanta K_p dan K_i diperoleh dari Sistem Fuzzy sebagai penala. Perhitungan dengan algoritma PI dimulai dengan menghitung *error* 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 23.



Gambar 23 Diagram alir algoritma PI

IV. PENGUJIAN DAN ANALISA

4.1 Pengujian Perangkat Keras

4.1.1 Pengujian Sensor PING™

Pengujian sensor PING™ dilakukan dengan mengukur level pada plant terhadap level *setpoint*. Pembacaan sensor dibatasi pada level 5 cm – 40 cm dan benda yang akan diukur berupa benda padat sebagai pelampung yang mendeteksi level ketinggian cairan. Data hasil pengukuran yang dilakukan adalah sebagai berikut.

Tabel 3 Hasil perbandingan level terukur dengan pembacaan sensor PING™.

No	Level terukur (cm)	Sensor PING
1	5	5.1
2	10	10.1
3	15	15.1

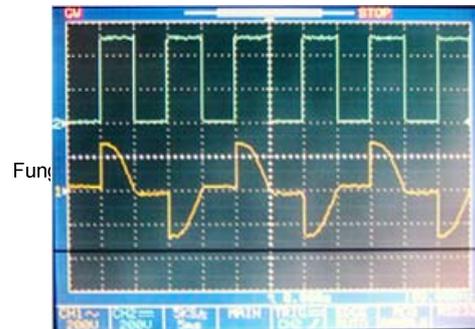
4	20	20
5	25	24.9
6	30	29.8
7	35	34.8
8	40	39.9

Dari Tabel 3 terlihat bahwa sensor PING™ memiliki *error* sebesar 0.1-0.2 cm.

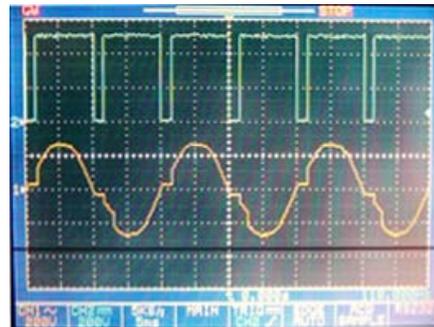
KS K S B

4.1 Pengujian Rangkaian Kendali Tegangan AC

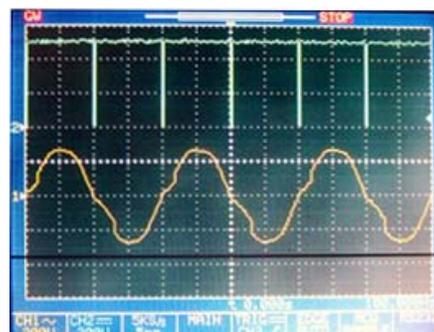
Pengamatan dilakukan dengan osiloskop terhadap gelombang picu dari mikrokontroler dan keluaran pengendali tegangan AC yang diberikan ke beban berdasarkan pengaturan sinyal kontrol. Hasil pengujian terlihat pada Gambar 24, Gambar 25, dan Gambar 26.



Gambar 24. Sinyal picu dan tegangan beban untuk $C_o = 0\%$.



Gambar 25. Sinyal picu dan tegangan beban untuk $C_o = 80\%$.



Gambar 26. Sinyal picu, tegangan beban untuk $C_o = 100\%$.

Ketiga bentuk gambar gelombang tegangan pada beban yang diuji, dapat diketahui bahwa semakin besar sinyal kontrol yang diberikan ke pemucuan (TCNT0), maka tegangan AC yang diberikan ke beban untuk tiap fasenya (fase positif dan fase negative)

akan semakin besar, yang berarti bahwa tegangan listrik yang diberikan ke beban akan semakin besar.

Pengujian juga dilakukan dengan cara mengukur tegangan sinyal picu dari mikrokontroler pinD.3 (TCNT0) dan tegangan keluaran dari rangkaian pengendali tegangan AC dengan variasi berdasarkan sinyal kontrol yang diberikan ke pemacu. Data hasil pengukuran dapat dilihat pada Tabel 4.

Table 4 Data pengujian tegangan sinyal picu dan kendali tegangan AC.

No	Nilai kontrol (%)	Nilai TCNT0	Tegangan ke beban (Volt)
1	0	148 (94h)	140
2	10	159 (9Fh)	155
3	20	169 (A9h)	180
4	30	180 (B4h)	195
5	40	191 (BFh)	200
6	50	201 (C9h)	204
7	60	212 (D4h)	205
8	70	223 (DFh)	206
9	80	234 (Eah)	210
10	90	244 (F4h)	215
11	100	255 (FFh)	225

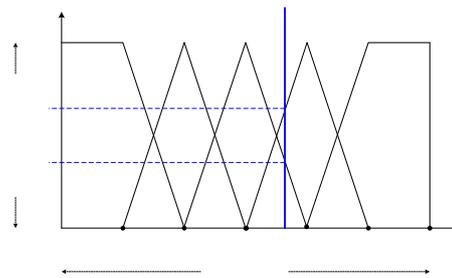
4.2 Pengujian Perangkat Lunak

4.2.1 Pengujian Pemrograman Sistem Logika Fuzzy

Pengujian pemrograman kendali logika Fuzzy dilakukan dalam tiga tahap, yaitu pengujian pemrograman Fuzzifikasi, pengujian pemrograman evaluasi aturan dan pengujian pemrograman defuzzifikasi.

A. Pengujian Pemrograman Fuzzifikasi

Pengujian pemrograman Fuzzifikasi bertujuan untuk mengetahui derajat keanggotaan tiap-tiap fungsi keanggotaan masukan setelah subrutin program Fuzzifikasi dijalankan. Sebagai contoh Pada Gambar 27 akan dicari derajat keanggotaan dari suatu masukan yaitu $error = 6$ cm.



Gambar 27 Fuzzifikasi dengan masukan $error = 6$ cm.

Derajat keanggotaan dapat ditentukan sebagai berikut :

$$\begin{aligned} Err &= 30.000 + ((e1_L * 30.000) / (30)) \\ &= 30.000 + (6 * 30.000) / (30) \\ &= 36.000 \end{aligned}$$

$$\begin{aligned} Error_Z &= (40.000 - 36.000) / 50 \\ &= 80 \end{aligned}$$

$$\begin{aligned} Error_P &= (36.000 - 30.000) / 50 \\ &= 120 \end{aligned}$$

Langkah selanjutnya yaitu menguji program Fuzzifikasi dengan melihat hasil keluaran $Error_P$ dan $Error_Z$ pada *display* LCD. Hasil eksekusi program memperlihatkan nilai yang sama dengan hasil Fuzzifikasi secara perhitungan lihat Gambar 28.

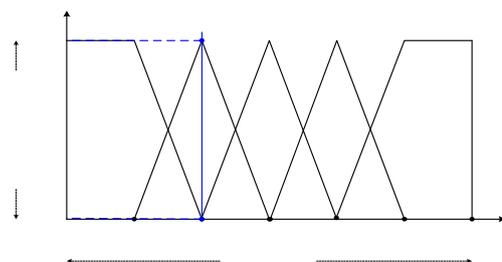


Gambar 28 Hasil pengujian program Fuzzifikasi dengan masukan $error = 6$ cm.

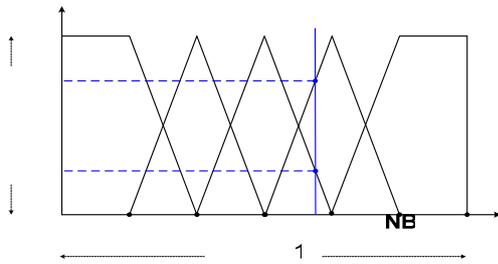
Dari hasil eksekusi program didapatkan nilai yang sama dengan hasil fuzzifikasi secara perhitungan.

B. Pengujian Pemrograman Evaluasi Aturan

Pengujian pemrograman evaluasi aturan bertujuan untuk mengetahui nilai keluaran Fuzzy setelah subrutin program evaluasi aturan dijalankan. Sebagai contoh akan dicari hasil keluaran dari suatu masukan yaitu $error = -10$ cm, $d_error = 4$ cm, fungsi keanggotaan diperlihatkan pada Gambar 29 dan Gambar 30.



Gambar 29 Masukan $Error = -10$ cm.



Gambar 30 Masukan $D_Error = 4$ cm.

Derajat keanggotaan dapat ditentukan sebagai berikut :

$$\begin{aligned} Err &= 30.000 + ((d_error_L * 30.000) / (reff/2)) \\ &= 30.000 + ((4 * 30.000) / (60/2)) \\ &= 20.000 \end{aligned}$$

$$\begin{aligned} Error_NB &= (20.000 - err) / 50 \\ &= (20.000 - 20.000) / 50 \\ &= 0 \end{aligned}$$

$$\begin{aligned} Error_N &= (20.000 - 10.000) / 50 \\ &= 200 \end{aligned}$$

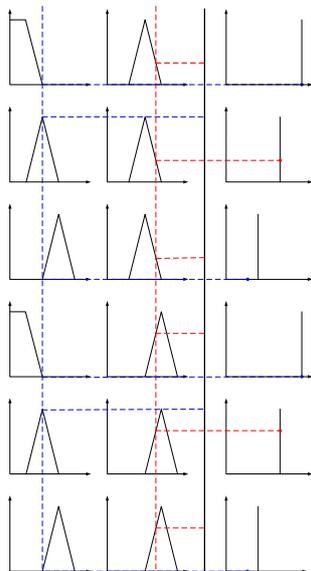
$$Error_Z = 0$$

$$\begin{aligned} D_Err &= 15.000 + (d_error_L * 15.000) / (reff/4) \\ &= 15.000 + (4 * 15.000) / 15 \\ &= 19.000 \end{aligned}$$

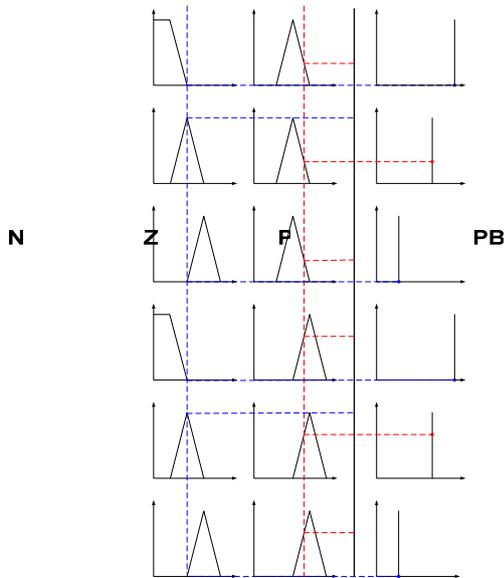
$$\begin{aligned} D_Error_Z &= (20.000 - d_err) / 25 \\ &= (20.000 - 19.000) / 25 \\ &= 40 \end{aligned}$$

$$\begin{aligned} D_Error_P &= (d_err - 15.000) / 25 \\ &= (19.000 - 15.000) / 25 \\ &= 160 \end{aligned}$$

Dengan menggunakan metode Implikasi Min:



Gambar 31 Pengujian evaluasi aturan konstanta K_p .



Gambar 32 Pengujian evaluasi aturan konstanta K_i .

Dengan pengujian evaluasi aturan di atas, maka dapat dihitung nilai *crisp* yaitu :

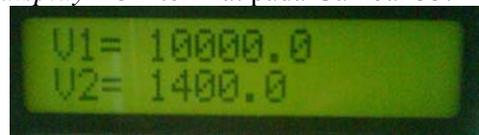
- Nilai pengujian evaluasi aturan K_p .

$$V_1 = (0 \times 60) + (40 \times 50) + (0 \times 44) + (0 \times 60) + (160 \times 50) + (0 \times 44) = 10000$$

- Nilai pengujian evaluasi aturan K_i .

$$V_2 = (0 \times 10) + (40 \times 7) + (0 \times 5) + (0 \times 10) + (160 \times 7) + (0 \times 5) = 1400$$

Langkah selanjutnya yaitu menguji program evaluasi aturan dengan melihat hasil keluaran pada *display* LCD terlihat pada Gambar 33.



Gambar 33 Tampilan hasil implikasi dan evaluasi aturan pada LCD.

C. Pengujian Pemrograman Fuzzy Reasoning

Pengujian program ini bertujuan untuk mendapatkan nilai *crisp* parameter kendali PI hasil eksekusi program maupun secara perhitungan. Sebagai contoh akan dicari hasil keluaran dari suatu masukan yaitu $error = -10$ cm, $d_error = 4$ cm .

$$Crisp\ output = \frac{\sum_i [(Fuzzy\ Output_i) \times (Posisi\ singleton\ di\ X\ axis_i)]}{\sum_i (Fuzzy\ Output_i)}$$

$$K_p = \frac{(0 \times 60) + (40 \times 50) + (0 \times 44) + (0 \times 60) + (160 \times 50) + (0 \times 44)}{0 + 40 + 0 + 0 + 160 + 0} = 50$$

$$K_i = \frac{(0 \times 10) + (40 \times 7) + (0 \times 5) + (0 \times 10) + (160 \times 7) + (0 \times 5)}{0 + 40 + 0 + 0 + 160 + 0} = 7$$

Langkah selanjutnya menguji program ini dengan melihat hasil keluaran pada *display* LCD pada Gambar 34 hasil eksekusi program

didapatkan nilai yang sama dengan hasil secara perhitungan.



Gambar 34 Tampilan Hasil Defuzzifikasi pada LCD.

4.2.2 Pengujian Program Kendali PI.

Pengujian program Kendali PI bertujuan untuk mengetahui nilai sinyal kontrol hasil eksekusi program.

contoh perhitungan :

$$error = 5, error1 = 0, Kp = 46, Ki = 6, Ts = 0.1,$$

$$Co1 = 0$$

$$Ti = 46/6 = 7.67$$

$$K1 = 46 \times 0.1/7.67 = 0.6$$

$$K2 = 46 + 0.6 = 46.6$$

$$K3 = 46$$

$$Co = 0 + 46.6 \times 5 - 46 \times 0 = 233$$

Untuk $Co > 100$, maka $Co = 100\%$

Untuk $Co < 0$, maka $Co = 0\%$

Tabel 5 Hasil perhitungan program nilai Co , Kp , Ki terhadap $error$ dan $error1$.

No	Error (cm)	Error1 (cm)	Kp	Ki	Co
1	8	0	48,80	6,30	100
2	5	0	47,00	6,10	100
3	2	1	45,20	5,50	46,20
4	1	1	44,60	5,20	0.5
5	0	1	44,00	5,05	0

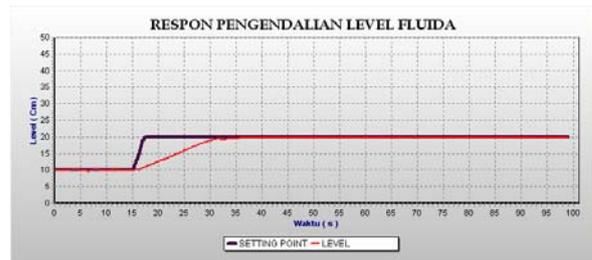
4.2.3 Pengujian Respon Sistem dengan PC.

A. Respon sistem kendali level tanpa gangguan.

Tanggapan pengendalian sistem level tanpa gangguan dengan memberikan nilai referensi level yang berbeda yaitu 15 Cm, 20 Cm, dan 25 Cm. Respon sistem dapat dilihat pada Gambar 35, Gambar 36, dan Gambar 37.



Gambar 35 Tanggapan sistem kendali level tanpa gangguan dengan referensi 15 Cm.



Gambar 36 Tanggapan sistem kendali level tanpa gangguan dengan referensi 20 Cm.



Gambar 37 Tanggapan sistem kendali level tanpa gangguan dengan referensi 25 Cm.

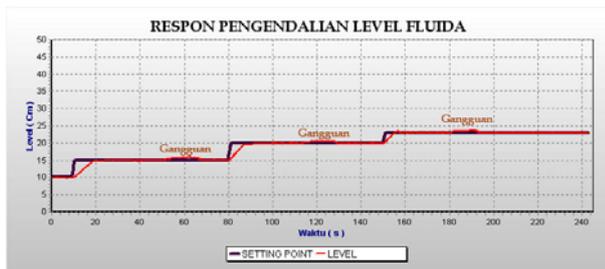
Gambar 35 adalah grafik respon kendali level dengan referensi 15 Cm. Respon sistem stabil tidak terjadi kesalahan keadaan tunak, setelah detik ke 100 pengujian dihentikan. Waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 8 detik. Gambar 36 adalah grafik respon kendali level dengan referensi 20 Cm. Respon sistem stabil tidak terjadi kesalahan keadaan tunak, setelah detik ke 100 pengujian dihentikan. Waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 15 detik. Gambar 37 adalah grafik respon kendali level dengan referensi 25 Cm. Respon sistem stabil tidak terjadi kesalahan keadaan tunak, setelah detik ke 100 pengujian dihentikan. Waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 24 detik.

Ketiga hasil pengujian sistem kendali level tanpa gangguan dapat disimpulkan bahwa semakin besar referensi level yang diberikan, maka waktu yang diperlukan untuk mencapai keadaan stabil akan semakin lama, dengan level awal yang sama. Kendali PI dengan penalaan logika Fuzzy pada sistem tanpa gangguan yang diuji sudah bekerja secara optimal, dimana semua respon sistem untuk referensi yang berbeda dapat mencapai keadaan stabil.

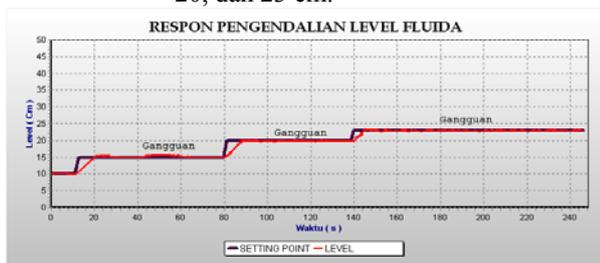
B. Respon sistem Kendali Level terhadap gangguan.

Pengujian terhadap pengaruh gangguan pada sistem kendali level dilakukan dengan dua variasi, yaitu pengujian pengaruh gangguan sesaat pada sistem dan pengaruh gangguan kontinyu. Pengujian ini dilakukan untuk mengetahui kemampuan atau untuk kerja kendali PI dengan penalaan logika Fuzzy terhadap gangguan luar.

Pengaruh gangguan sesaat pada sistem kendali level dengan variasi pemberian referensi level dilakukan dengan cara memberikan tambahan aliran pada bagian inlet plant. Pemberian gangguan dilakukan ketika mencapai keadaan stabil, sesuai dengan referensinya. Pemberian gangguan sesaat dilakukan selama 10 sekon setelah itu aliran inlet dihentikan. Gangguan dilakukan pada referensi 15 cm, 20 cm, dan 23 cm. Hasil pengujian respon sitem ditunjukkan pada Gambar 38.



Gambar 38 Tanggapan sistem kendali level dengan gangguan sesaat dengan referensi 15, 20, dan 23 cm.



Gambar 39 Tanggapan sistem kendali level dengan gangguan kontinyu dengan referensi 15cm, 20cm, dan 23 cm.

V. PENUTUP

5.1 Kesimpulan

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

1. Pada pengujian dengan nilai referensi tetap, mampu menghasilkan respon yang baik yaitu *rise time* yang cepat dan *overshoot* yang sangat kecil, untuk referensi 15 cm menghasilkan waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 8 detik, untuk referensi 20 cm menghasilkan waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 15 detik, untuk referensi 25 cm menghasilkan waktu tunda (t_d) yang diperlukan adalah 1 detik, waktu naik (t_r) 25 detik dari ketinggian awal 10 cm.
2. Pada pengujian dengan gangguan sesaat, pada referensi 15 cm, 20 cm, dan 23 cm, gangguan diberikan dengan memberikan aliran inlet, selama 10 detik level sistem berangsur-angsur

naik sampai range 1-2 cm. Setelah itu aliran inlet dihentikan, dan waktu pemulihan menuju kestabilan selama 20 detik setelah terjadi lonjakan.

3. Pada pengujian dengan gangguan kontinyu, diberikan perubahan debit pada sisi *inlet*, respon keluaran mengalami *offset* keadaan tunak sebesar 0,1 cm.
4. Penalaan konstanta PI dengan logika Fuzzy akan terus mengalami perubahan konstanta sesuai dengan perubahan nilai *error* dan *delta error*.

5.2 Saran

Untuk pengembangan sistem lebih lanjut, maka ada beberapa saran yang dapat dilakukan yaitu sebagai berikut:

1. Dapat dilakukan perancangan kembali penalaan konstanta PI dengan logika Fuzzy, dengan basis aturan maupun fungsi keanggotaan (masukan/keluaran) yang berbeda untuk memperoleh respon sistem yang lebih baik dan memuaskan.
2. 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 Pompa 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 Pengering 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 Pengering 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>.
- (17) -----, *PING)))™ Ultrasonic Distance Sensor Data Sheet*, <http://www.parallax.com>.

Eduward Tigor H.L (L2F 306 025)



Lahir di Pomalaa, 7 Juni 1983. 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. 132 086 662

Tanggal: _____

Iwan Setiawan, ST, MT

NIP. 132 283 183

Tanggal: _____