



**SISTEM PENGATURAN LAJU ALIRAN AIR PADA
PLANT WATER TREATMENT SKALA RUMAH
TANGGA DENGAN KONTROL *FUZZY-PID***

SKRIPSI

Oleh

**Abdur Rohman
NIM 111910201030**

**PROGRAM STUDI STRATA 1
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**



**SISTEM PENGATURAN LAJU ALIRAN AIR PADA
PLANT WATER TREATMENT SKALA RUMAH
TANGGA DENGAN KONTROL *FUZZY*-PID**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh

**Abdur Rohman
NIM 111910201030**

**PROGRAM STUDI STRATA 1
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**

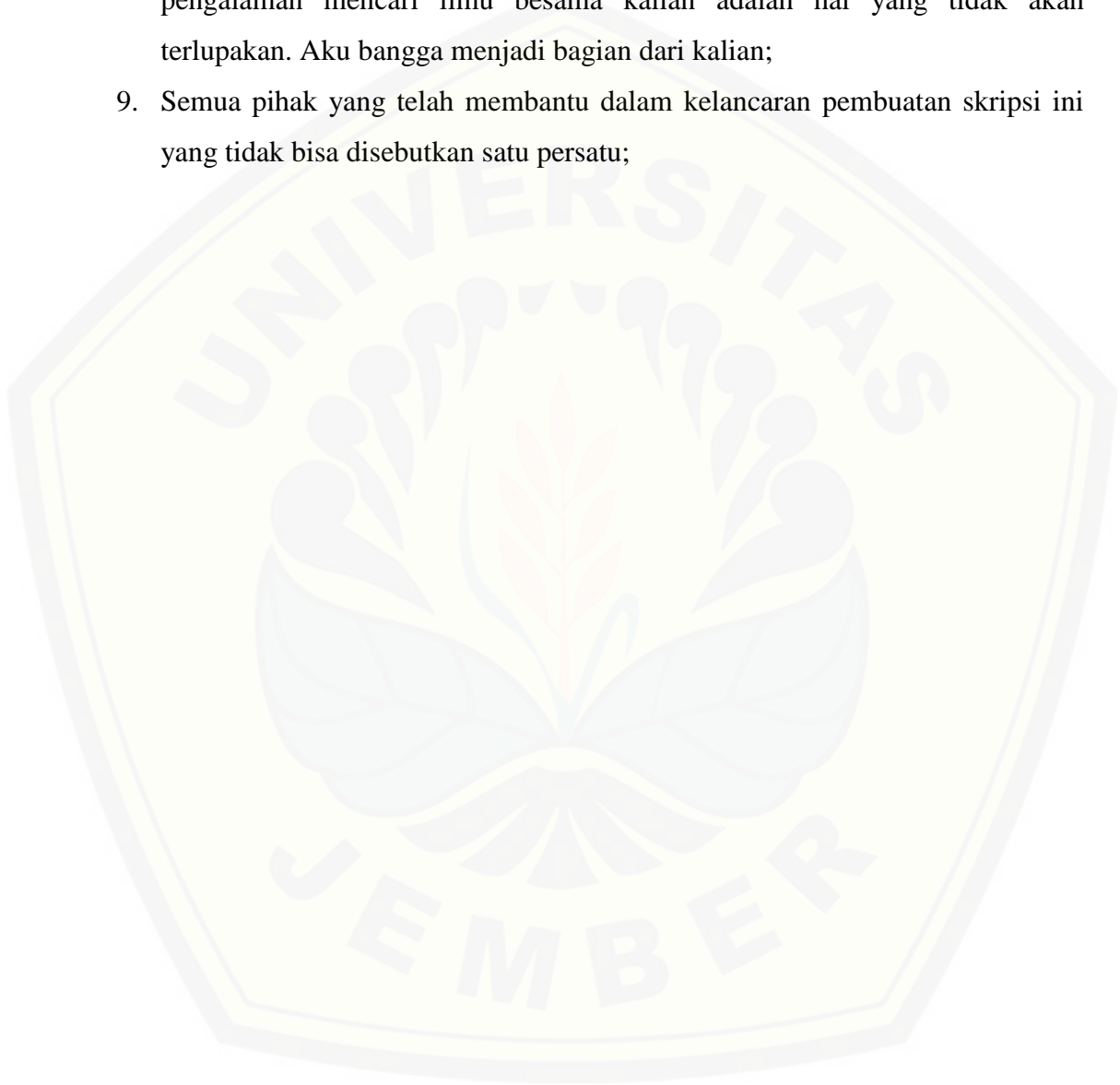
PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT yang telah memberikan limpahan rahmat, taufiq, serta hidayah yang sangat luar biasa kepada penulis sehingga skripsi ini dapat selesai dengan baik. Tidak lupa sholawat serta salam kita haturkan kepada Nabi Besar Muhammad SAW yang menunjukkan kita sebagai manusia menuju jalan yang terang benderang dengan kehidupan yang lebih baik. Skripsi ini merupakan karya yang tidak pernah ternilai dan terlupakan bagi penulis yang selain sebagai syarat menyelesaikan program studi juga untuk kemajuan umat manusia agar lebih baik. Oleh karenanya karya ini ingin saya persembahkan untuk:

1. Allah SWT, karena perlindungan, pertolongan, dan ridho-Nya penulis mampu menyelesaikan tugas akhir ini dengan baik serta Nabi Besar Muhammad SAW;
2. Kedua orang tuaku, Ayahanda Mohammad Ra'i dan Ibunda Murdiana, Adikku Nur aini, Hidayatullah, dan Abdur Rohim terima kasih dukungan, bantuan, serta doa restunya hingga selesainya studi ini;
3. Kerabat dan sanak keluarga, dan semua keluargaku yang telah memberikan dukungan, motivasi, dan doa;
4. Dosen pembimbing skripsiku, M. Agung Prawira Negara, S.T., M.T. selaku DPU dan Bapak Bambang Supeno, S.T., M.T. selaku DPA yang bersedia meluangkan waktu dan pikirannya guna memberikan bimbingan dan arahan demi terselesainya skripsi ini;
5. Dosen penguji 1, Bapak Satryo Budi Utomo, S.T. dan Dosen penguji 2, Bapak Widjonarko, S.T., M.T. yang telah meluangkan banyak waktu dan pikiran guna memberikan pengarahan demi kemajuan dan terselesainya penulisan skripsi ini dengan baik;
6. Semua Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Jember yang telah memberikan ilmu dan membimbing selama kurang lebih empat tahun ini. Penulis sampaikan banyak terima kasih atas semua ilmu, didikan, dan pengalaman yang sangat luar biasa;
7. Sahabat-sahabatku Ahmad Syafiul Umam dan Aditya Dharmawan, yang telah menemani, mengenalkan Jember serta membuat hidup di Jember lebih

berwarna dari awal masuk kuliah sampai semester akhir ini. Tidak lupa terima kasih kepada jajaran keluarga KOPI ASAP yang telah menjadi keluarga kedua selama berada di bangku perkuliahan;

8. Teman-teman elektro yang telah berjuang bersama di almamater tercinta, pengalaman mencari ilmu bersama kalian adalah hal yang tidak akan terlupakan. Aku bangga menjadi bagian dari kalian;
9. Semua pihak yang telah membantu dalam kelancaran pembuatan skripsi ini yang tidak bisa disebutkan satu persatu;



MOTTO

*“Barang siapa bersungguh-sungguh, sesungguhnya kesungguhannya itu
adalah untuk dirinya sendiri”
(QS Al-Ankabut : 6)*

*“Belajarlah dari hari kemarin, jalani hari ini, berharaplah untuk hari esok,
Yang penting jangan berhenti bertanya”
(Albert Einstein)*

*“Pendidikan merupakan perlengkapan paling baik untuk hari tua”
(Aristoteles)*

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Abdur Rohman

NIM : 111910201030

menyatakan dengan sesungguhnya bahwa karya tulis yang berjudul “**Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga Dengan Kontrol *Fuzzy-PID***” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan subtransi disebutkan sumbernya, dan belum pernah diajukan pada institusi mana pun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 24 Mei 2016
Yang menyatakan,

Abdur Rohman
NIM 111910201030

SKRIPSI

**SISTEM PENGATURAN LAJU ALIRAN AIR PADA
PLANT WATER TREATMENT SKALA RUMAH TANGGA
DENGAN KONTROL FUZZY-PID**

Oleh

Abdur Rohman
NIM 111910201030

Pembimbing:

Dosen Pembimbing Utama : M. Agung Prawira Negara, S.T., M.T.

Dosen Pembimbing Anggota : Bambang Supeno, S.T., M.T.

PENGESAHAN

Skripsi berjudul “**SISTEM PENGATURAN LAJU ALIRAN AIR PADA PLANT WATER TREATMENT SKALA RUMAH TANGGA DENGAN KONTROL FUZZY-PID**” telah diuji dan disahkan oleh Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember pada:

Hari : Selasa

Tanggal : 24 Mei 2016

Tempat : Ruang Sidang Fakultas Teknik, Universitas Jember

Tim Penguji:

Dosen Pembimbing Utama

Dosen Pembimbing Anggota

M. Agung Prawira Negara, S.T., M.T.
NIP. 19871217 201212 1 003

Bambang Supeno, S.T., M.T.
NIP. 19690630 199512 1 001

Dosen Penguji I

Dosen Penguji II

Satryo Budi Utomo, S.T., M.T.
NIP. 19850126 200801 1 002

Widjonarko, AMd., S.T., M.T.
NIP. 19710908 199903 1 001

Mengesahkan
Dekan Fakultas Teknik

Dr. Ir. Entin Hidayah, M.U.M.
NIP. 19661215 199503 2 001

Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga Dengan Kontrol *Fuzzy-PID*

Abdur Rohman

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Air merupakan kebutuhan yang sangat vital bagi kehidupan manusia. Karena itu jika kebutuhan air tersebut belum tercukupi maka dapat memberikan dampak yang besar terhadap kerawanan kesehatan maupun sosial. Pengadaan air bersih di Indonesia khususnya untuk skala yang besar masih terpusat di daerah perkotaan, dan dikelola oleh PAM pada kota yang bersangkutan. Namun demikian secara nasional jumlahnya masih belum mencukupi dan dapat dikatakan relatif kecil. Untuk daerah yang belum mendapatkan pelayanan air bersih dari PAM umumnya mereka menggunakan beberapa sumber mata air diantaranya: air tanah, air sungai, air hujan, dan sumber mata air lainnya. Permasalahan yang timbul yaitu sering dijumpai bahwa kualitas air tanah maupun air sungai yang digunakan masyarakat kurang memenuhi syarat sebagai air minum yang sehat bahkan di beberapa tempat bahkan tidak layak untuk diminum. Untuk menanggulangi masalah tersebut, salah satu alternatif yaitu mengolah air tanah atau air sumur sehingga didapatkan air dengan kualitas yang memenuhi syarat kesehatan dengan merancang sebuah alat filtrasi air. Pada penelitian ini dirancang sebuah alat filtrasi air skala rumah tangga dengan metode pengontrolan pada aliran air pada proses filtrasi. Metode kontrol yang digunakan yaitu *fuzzy_PID*. Kontrol *fuzzy_PID* digunakan untuk mengatur debit aliran air yang masuk kedalam *filter* dengan mengatur kecepatan pompa air yang dipengaruhi oleh kejernihan air hasil filtrasi. *Fuzzy logic* digunakan untuk menghasilkan parameter K_p , K_i , K_d pada kontrol PID dengan *input* berupa *error* dan perubahan *error*. Sensor yang digunakan pada penelitian ini ialah sensor kejernihan 1, sensor kejernihan 2 dan sensor *flowmeter*. Sensor kejernihan 1 digunakan untuk mendeteksi nilai kejernihan air *input* dan sensor kejernihan 2 digunakan untuk kejernihan air *output*. Kemudian sebagai pusat kendalinya yaitu arduino uno. Pada penelitian yang dilakukan diketahui bahwa kecepatan aliran air dapat mempengaruhi nilai kekejernihan air yang dihasilkan pada proses filtrasi. Ketika kejernihan air *output* dibawah *setpoint*, maka kecepatan aliran air akan diperlambat sampai kejernihan air *output* sama dengan *setpoint*. Pada percobaan *setpoint* 94 %, Kecepatan debit aliran air berubah dari 872 (Liter/jam) menjadi 752 (Liter/jam) pada saat terjadi perubahan kejernihan air *output* filtrasi dari 94 % menjadi 93%.

Kata kunci: Arduino Uno, *Fuzzy_PID*, Kejernihan, Sensor *Flowmeter*.

***Water Flow Rate Control System At Water Treatment Plant Household Scale
With Fuzzy-PID Control***

Abdur Rohman

Electrical Engineering, Engineering Faculty, Jember University

ABSTRACT

Water is a vital requirement for human life. Therefore if the water needs have not been fulfilled, it can have a large impact on health and social insecurity. Water supply in Indonesia, especially for a large scale is still concentrated in urban areas, and managed by PAM in the city concerned. However, nationally the number is still insufficient and can be said to be relatively small. For areas that do not receive water services from the PAM generally they use some water sources include: ground water, river water, rain water, and other water sources. The problems that arise are often found that the quality of ground water and river water used less qualified people as healthy drinking water even in some places even unfit for drinking. To overcome the problem, one alternative is to process groundwater or well water to obtain water quality meets the health requirements to design a water filtration apparatus. In this study designed a tool household water filtration by a method of controlling the flow of water in the filtration process. Control method used is fuzzy_PID. Fuzzy_PID controls used to regulate the flow of water into the filter by adjusting the speed of the water pump is influenced by the clarity of the water filtration results. Fuzzy logic is used to generate parameters K_p , K_i , K_d in PID control with the input of the error and error change. Sensors used in this study is clarity sensor 1, sensor 2 and sensor flowmeter clarity. Clarity sensor 1 is used to detect the value of clarity water clarity sensor inputs and 2 outputs are used for water clarity. Then as the control center that is arduino uno. In the study conducted in mind that the water flow rate can affect the value kekejernihan water produced in the process of filtration. When water clarity below the setpoint output, the speed of the water flow will be slowed until the water clarity equal to the output setpoint. At trial setpoint of 94%, speed of water flow changed from 872 (Liter / h) to 752 (liter / h) in the event of changes in water clarity filtration output from 94% to 93%.

Keywords: Arduino Uno, Fuzzy_PID, flowmeter sensor, turbidity.

RINGKASAN

Sistem Pengaturan Laju Aliran Air Pada Plant Water Treatment Skala Rumah Tangga Dengan Kontrol Fuzzy-PID; Abdur Rohman, 111910201030; 2016: 84 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Air merupakan kebutuhan yang sangat vital bagi kehidupan manusia. Karena itu jika kebutuhan air belum tercukupi maka dapat memberikan dampak yang besar terhadap kerawanan kesehatan maupun sosial. Daerah yang belum mendapatkan pelayanan air bersih dari PAM umumnya mereka menggunakan beberapa sumber mata air diantaranya: air tanah, air sungai, air hujan, dan sumber mata air lainnya. Pemakaian air yang tidak memenuhi standar kualitas dapat menimbulkan gangguan kesehatan. Oleh karena itu, dibutuhkan pengolahan air tanah atau air sumur sehingga didapatkan air dengan kualitas yang memenuhi syarat kesehatan dengan merancang sebuah alat filtrasi air. Kita ketahui bahwa semakin lama filter air mengalami kinerja yang semakin lemah akibat kotoran yang semakin bertambah dalam media filter sehingga perlu adanya pengurasan dan pencucian filter. Apabila filter telah jenuh, maka proses penyaringan menjadi kurang sempurna, kurang cepat, dan kurang efektif. Oleh karena itu, untuk menjaga performansi alat filtrasi dibutuhkan sebuah *control system* dari alat filtrasi itu sendiri. Dalam judul skripsi ini yaitu “Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga Dengan Kontrol *Fuzzy-PID*” merupakan teknologi pengolahan air dengan *monitoring* kondisi air hasil pengolahan serta *control* kecepatan pompa air yang mengatur debit air yang masuk kedalam tabung filter. Selain itu ditambahkan pula proses *backwash* pada media filter.

Desain perancangan alat yang digunakan dalam penelitian ini yaitu pada *box control* terdapat rangkaian elektronika yaitu: Arduino uno, rangkaian *power supply*, dan *driver* pompa. Pada tabung *filter* 1 dan 2 menggunakan pipa besi dengan ukuran tabung *filter* 1 = 6 *inchi*, tabung *filter* 2 = 3 *inchi*. Pada tabung *filter* 1 terdapat media *filter* berupa mangan zeolit, silica, karbon aktif, sedangkan pada tabung *filter* 2, media *filter*nya berupa kasa. Kemudian pompa air dengan daya 125 watt dan

tegangan kerja 220 volt digunakan untuk mengalirkan air pada proses penjernihan. Sensor kejernihan menggunakan LED dan LDR yang berada pada akuarium penampungan air hasil filtrasi. Terdapat dua sensor yaitu pada akuarium 1 dan akuarium 2. Pada akuarium 1 berfungsi untuk mendeteksi kejernihan air yang masuk menuju alat filtrasi. Sensor kejernihan 2 digunakan untuk mendeteksi kejernihan aktual air air filtrasi. Sensor *flowmeter* dalam penelitian ini digunakan untuk *monitoring* kecepatan aliran air pada saat sistem bekerja. Aliran air yang diukur ialah air yang masuk pada tabung filtrasi. Pengukuran kecepatan aliran air ini digunakan untuk bahan analisis pengaruh kecepatan aliran air terhadap kejernihan hasil proses filtrasi. Sensor *flowmeter* yang digunakan ialah ukuran $\frac{3}{4}$ inch yang terdiri dari katup plastik, rotor air, dan sebuah sensor *hall-effect*.

Pengendali *fuzzy*_PID yang dirancang memiliki dua buah masukan yaitu *error* dan perubahan *error*. Himpunan *fuzzy* untuk masukan *error* dan perubahan *error*, masing-masing dibagi menjadi 5 kelas. Himpunan *fuzzy* untuk keluarannya dibagi dalam 5 tingkat sinyal *control* yang berbentuk *discrete*, nama linguistik kelima tingkatan ini adalah parameter. Mekanisme defuzifikasi menggunakan *output* berupa konstanta yang dibagi dalam 5 tingkat yang berbentuk *discrete*, nama linguistik kelima tingkatan ini adalah parameter. Kelima parameter keluaran tersebut K (kecil), S (sedang), B (besar), SB (sedang besar) dan BS (Besar sekali). Untuk nilai K_p , K bernilai 0, S bernilai 2, B bernilai 4, SB bernilai 10, dan BS bernilai 15. Pada penalaan parameter T_i , K bernilai 0, S bernilai 5, B bernilai 10, SB bernilai 15 dan BS bernilai 20. Untuk parameter T_d dirancang dengan K bernilai 0, S bernilai 1, B bernilai 3, SB bernilai 5 dan BS bernilai 7. Untuk mendapatkan nilai *crisp output* dari himpunan *fuzzy* ini dapat digunakan metode rata-rata berbobot (*Center average defuzzifier*).

Pada pengujian yang pertama sistem dengan tingkat kejernihan air *input* sebesar 64 %. Nilai *setpoint* yang diberikan sebesar 92 %. Diketahui bahwa nilai $T_{out} > setpoint$. Pada kondisi ini air hasil filtrasi telah memenuhi kondisi kejernihan air filtrasi di atas *setpoint*, sehingga kecepatan aliran air konstan pada nilai 872 liter/jam. Pada pengujian yang pertama diketahui bahwa kemampuan media filtrasi dapat memfilter air dengan tingkat kejernihan air diatas 92 %, sehingga tidak terjadi

perubahan kecepatan aliran air. Oleh karena itu untuk mengetahui pengaruh perubahan kecepatan aliran air terhadap nilai kejernihan air *output* dilakukan dengan menaikkan nilai *setpoint* yang diberikan yaitu sebesar 94 %.

Pada pengujian yang kedua dengan tingkat kejernihan air *input* dibuat sama yaitu rata-rata sebesar 64 %. Air yang di filtrasi didapat dari kolam ikan kedua dengan *volume* 12.000 liter. Diketahui bahwa nilai *Tout* pada awal filtrasi lebih besar dari *setpoint* karena pada awal filtrasi sensor kejernihan belum mendeteksi hasil kejernihan *output* (*Tout*), sehingga kecepatan motor masih pada kondisi normal, yaitu pada kecepatan maksimal. Setelah sensor kejernihan *Tout* mendeteksi perubahan hasil kejernihan air, yaitu dari nilai 94 % menjadi 93 %, kecepatan motor pompa melambat sesuai dengan sinyal kontrol keluaran *fuzzy_PID* yaitu dengan persamaan (kontrol motor = 255 – sinyal *control*). Kemudian setelah kecepatan motor melambat kejernihan *output* (*Tout*) kembali terjadi kenaikan sampai pada tingkat kejernihan 94 %. Pada kondisi ini, air hasil filtrasi telah memenuhi kondisi kejernihan air filtrasi sesuai dengan *setpoint* sehingga motor pompa kembali pada kecepatan normal dengan *output* PWM 254 dan kecepatan aliran air rata rata 872 L/H.

Pada pengujian yang ketiga dengan tingkat kejernihan air *input* dibuat sama yaitu rata-rata sebesar 64 %. Air yang akan di filtrasi didapat dari kolam ikan kedua dengan *volume* 12.000 liter. Berikut gambar data grafik hasil pengujian sistem melakukan proses filtrasi. Diketahui bahwa nilai *Tout* pada awal filtrasi lebih besar dari *setpoint* karena pada awal filtrasi sensor kejernihan belum mendeteksi hasil kejernihan *output* (*Tout*), sehingga kecepatan motor masih pada kondisi normal, yaitu pada kecepatan maksimal. Setelah sensor kejernihan *Tout* mendeteksi hasil kejernihan air dibawah 96 %, kecepatan motor pompa melambat dengan aliran air rata-rata 753(L/H). Kemudian setelah kecepatan motor melambat kejernihan *output* (*Tout*) kembali terjadi kenaikan sampai pada tingkat kejernihan 96 %. Pada kondisi ini, air hasil filtrasi telah memenuhi kondisi kejernihan air filtrasi sesuai dengan *setpoint* sehingga motor pompa kembali pada kecepatan normal dengan *output* PWM 254 dan kecepatan aliran air rata rata 872 L/H. Kemudian terjadi penurunan kejernihan kembali sebesar 95%, namun pada kondisi ini, sistem sudah tidak dapat

mencapai *setpoint* kejernihan 96% dan *output* filtrasi bertahan pada rentang kejernihan 93% - 95%.

Dari hasil pengujian yang telah dilakukan diketahui bahwa sistem sulit untuk mencapai kondisi kejernihan air hasil filtrasi pada *setpoint* yang telah ditetapkan pada awal proses. Hal ini dipengaruhi oleh faktor utama yaitu pada desain tabung filtrasi. Perbandingan antara debit filtrasi dengan ukuran tabung filtrasi tidak seimbang. Tabung filtrasi yang berukuran kecil, tidak mampu menampung media filter lebih banyak, sehingga proses filtrasi kurang maksimal dalam menyaring air dengan debit air yang besar. Selain itu penggunaan kontrol logika *fuzzy_PID* pada plant ini memiliki kelemahan, yaitu ketika sistem telah mencapai kejernihan pada nilai *setpoint*, kecepatan aliran air menjadi tidak stabil. Keadaan tersebut disebabkan, karena pada saat (kejernihan hasil filtrasi = *setpoint*), motor kembali ke kecepatan maksimal, sehingga membuat kejernihan air *output* kembali menurun.

Dari beberapa data pengujian dapat diketahui bahwa hasil dari penelitian sesuai dengan teori. Perubahan kecepatan aliran air yang masuk ke media filtrasi dapat mempengaruhi hasil dari kejernihan air yang dihasilkan. Semakin lambat aliran air yang masuk kedalam media filtrasi maka hasil air dari proses filtrasi semakin maksimal. Begitu pula sebaliknya, semakin cepat kecepatan aliran air yang masuk kedalam media filtrasi, maka hasil air dari proses filtrasi semakin kurang maksimal. Selain itu faktor lain yang mempengaruhi hasil kejernihan dari filtrasi yaitu pada media filtrasi yang digunakan. Pada proses filtrasi dengan debit aliran air yang tinggi diperlukan media filtrasi yang memiliki pori-pori sangat kecil.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul "**Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga Dengan Kontrol *Fuzzy-PID***". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan Skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak M. Agung Prawira Negara, S.T., M.T., selaku Dosen Pembimbing Utama dan Bapak Bambang Supeno, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Bapak Dr. Azmi Saleh, S.T., M.T., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
3. Ayahanda Mohammad Ra'i dan Ibunda Murdiana, Adikku Nur aini, Hidayatullah, dan Abdur Rohim terima kasih dukungan, bantuan, kasih sayang serta do'a restunya;
4. Teman-Teman ELEKTRO 11 yang tidak bisa saya sebutkan satu persatu, terima kasih atas dukungan dan bantuannya selama proses penyusunan Skripsi ini;
5. Pihak-pihak lain yang tidak bisa saya sebutkan satu persatu, terima kasih atas dukungan dan motivasi kalian dalam penyusunan Skripsi ini.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat.

Jember, 24 Mei 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTTO	v
HALAMAN PERNYATAAN	vi
HALAMAN PEMBIMBINGAN	vii
HALAMAN PENGESAHAN	viii
ABSTRAK	ix
RINGKASAN	xi
PRAKATA	xiii
DAFTAR ISI	xiv
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN	xix
BAB 1. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Batasan Masalah Penelitian	3
1.6. Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA	
2.1. Filtrasi Air	5
2.2. Filter Pasir Cepat	5
2.3. Tipe Filtrasi Declining Rate	6
2.4. Proses Backwash	7
2.5. Sensor Kejernihan Air	7
2.6. Pengaturan kecepatan motor AC dengan Triac	8
2.6.1. Triac	8

2.6.2. MOC 3041	10
2.7. Arduino uno	10
2.8. Pengendali logika fuzzy	11
2.8.1. Fuzzifikasi	12
2.8.2. Basis Pengetahuan	13
2.8.3. Defuzzifikasi.....	16
2.9. Kontrol PID.....	17
2.10. Desain Kontrol Logika Fuzzy_PID	23
BAB 3. METODOLOGI PENELITIAN	
3.1. <i>Study literature</i>	25
3.2. Tempat dan Waktu Penelitian	25
3.3. Alat dan Bahan	25
3.4. Perancangan Alat	26
3.5. Diagram Blok dan <i>Flowchart</i>	27
3.6. Realisasi Pembuatan Alat	28
3.7. Rangkaian Sensor	30
3.8. <i>Display</i> dan <i>Pushbutton</i>	34
3.9. Rangkaian <i>Driver PWM Digital Dimmer</i> untuk Pompa	37
3.10. Perancangan Pengendali <i>Fuzzy_PID</i>	40
BAB 4. HASIL DAN PEMBAHASAN	
4.1. Pengujian Rangkaian Sensor.....	47
4.1.1. Pengujian Sensor Kejernihan Air	47
4.1.2. Pengujian Sensor <i>Flowmeter</i>	50
4.2. Pengujian Kontrol <i>Fuzzy_PID</i>	51
4.3. Pengujian Kemampuan Sistem Melakukan Filtrasi	55
BAB 5. PENUTUP	
5.1. Kesimpulan	60
5.2. Saran.....	61

DAFTAR PUSTAKA	62
LAMPIRAN	63



DAFTAR TABEL

	Halaman
Tabel 3.1. <i>Output</i> Tegangan <i>Driver</i> PWM digital <i>Dimmer</i>	39
Tabel 3.2. Aturan <i>Fuzzy</i> Keluaran <i>Kp</i>	42
Tabel 3.3. Aturan <i>Fuzzy</i> Keluaran <i>Ti</i>	43
Tabel 3.4. Aturan <i>Fuzzy</i> Keluaran <i>Td</i>	43
Tabel 4.1. Pengujian Sensor Kejernihan Air	48
Tabel 4.2. Pengujian Sensor <i>Flowmeter</i>	50
Tabel 4.3. Hasil Implikasi Aturan <i>Fuzzy</i>	52
Tabel 4.4. Keluaran <i>Kp</i> , <i>Ti</i> , <i>Td</i>	53
Tabel 4.5. Pengujian Kontrol <i>fuzzy_PID</i>	54
Tabel 4.6. Pengaruh kejernihan Output air hasil filtrasi terhadap kecepatan aliran air	56
Tabel 4.7. Pengaruh kejernihan Output air hasil filtrasi terhadap kecepatan aliran air	58

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Sensor LDR.....	8
Gambar 2.2. Simbol Triac.....	9
Gambar 2.3. Kurva karakteristik Triac	10
Gambar 2.4. Sistem <i>loop</i> tertutup dengan pengendali <i>Fuzzy</i>	12
Gambar 2.5. Struktur dasar pengendali <i>fuzzy</i>	12
Gambar 2.6. Proses fuzzifikasi	13
Gambar 2.7. Tanggapan sistem terhadap masukan undak satuan.....	14
Gambar 2.8. Proses pengambilan keputusan metode sugeno	16
Gambar 2.9. Diagram blok kontroler proporsional.....	17
Gambar 2.10. <i>Proporsional band</i> dari kontroler <i>proporsional</i>	18
Gambar 2.11. Kurva sinyal kesalahan $E(T)$ terhadap T dan kurva $U(T)$ Terhadap T pada pembangkit kesalahan Nol	19
Gambar 2.12. Diagram blok hubungan antara besaran kesalahan dengan <i>Controller</i> integral.....	20
Gambar 2.13. Perubahan keluaran sebagai akibat penguatan dan kesalahan	20
Gambar 2.14. Kurva waktu hubungan <i>input-output</i> kontroler diferensial..	21
Gambar 2.15. Diagram blok kontroler PID <i>analog</i>	22
Gambar 2.16. Hubungan dalam fungsi waktu antara sinyal keluaran Dengan masukan <i>controller</i> PID.....	23
Gambar 2.17. Fungsi keanggotaan.....	24
Gambar 3.1. Desain perancangan alat.....	26
Gambar 3.2. Diagram blok sistem	27
Gambar 3.3. <i>Flowchart</i> sistem.....	27
Gambar 3.4. Sistem keseluruhan.....	29
Gambar 3.5. Rangkaian sensor kejernihan air	31
Gambar 3.6. Pembacaan sensor awal.....	32
Gambar 3.7. Pembacaan sensor setelah dikalibrasi	33
Gambar 3.8. Sensor <i>flowmeter</i> pada sistem	34

Gambar 3.9.	Antar muka LCD 4 Bit.....	34
Gambar 3.10.	Konfigurasi pin <i>pushbutton</i>	35
Gambar 3.11.	Pembacaan <i>keypad right</i>	36
Gambar 3.12.	Pembacaan <i>keypad Left</i>	36
Gambar 3.13.	Pembacaan <i>keypad Up</i>	37
Gambar 3.14.	Pembacaan <i>keypad down</i>	37
Gambar 3.15.	Pembacaan <i>keypad select</i>	38
Gambar 3.16.	Rangkaian <i>driver PWM digital Dimmer</i>	39
Gambar 3.17.	<i>Driver PWM digital Dimmer</i>	39
Gambar 3.18.	Grafik output tegangan driver	40
Gambar 3.19.	Diagram blok pengendali <i>fuzzy_PID</i>	41
Gambar 3.20.	Keanggotaan <i>error</i>	41
Gambar 3.21.	Keanggotaan perubahan <i>error</i>	42
Gambar 3.22.	Pemograman <i>rule base</i> pada arduino IDE	44
Gambar 3.23.	Pemograman aturan <i>fuzzy</i>	45
Gambar 3.24.	Pemograman kontrol PID.....	46
Gambar 4.1.	Sensor kejernihan air.....	47
Gambar 4.2.	Grafik pembacaan sensor kejernihan air	49
Gambar 4.3.	Gambar pengukuran air baku dengan <i>turbidymeter</i>	49
Gambar 4.4.	Grafik pembacaan sensor <i>flowmeter</i>	51
Gambar 4.5.	Grafik respon sistem dengan <i>setpoint 92 %</i>	56
Gambar 4.6.	Grafik respon sistem dengan <i>setpoint 94 %</i>	57

DAFTAR LAMPIRAN

	Halaman
A. LAMPIRAN HARDWARE.....	64
B. LAMPIRAN LISTING PROGRAM ARDUINO.....	71



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Air merupakan kebutuhan yang sangat vital bagi kehidupan manusia. Karena itu jika kebutuhan air tersebut belum tercukupi maka dapat memberikan dampak yang besar terhadap kerawanan kesehatan maupun sosial. Pengadaan air bersih di Indonesia khususnya untuk skala yang besar masih terpusat di daerah perkotaan, dan dikelola oleh PAM pada kota yang bersangkutan. Namun demikian secara nasional jumlahnya masih belum mencukupi dan dapat dikatakan relatif kecil. Untuk daerah yang belum mendapatkan pelayanan air bersih dari PAM umumnya mereka menggunakan beberapa sumber mata air diantaranya: air tanah, air sungai, air hujan, dan sumber mata air lainnya.

Permasalahan yang timbul yaitu sering dijumpai bahwa kualitas air tanah maupun air sungai yang digunakan masyarakat kurang memenuhi syarat sebagai air minum yang sehat bahkan di beberapa tempat bahkan tidak layak untuk diminum. Air yang layak diminum mempunyai persyaratan tertentu yakni persyaratan fisis, kimiawi dan bakteriologis, syarat tersebut merupakan satu kesatuan. Jadi jika ada satu saja parameter yang tidak memenuhi syarat maka air tersebut tidak layak untuk diminum. Pemakaian air yang tidak memenuhi standar kualitas tersebut dapat menimbulkan gangguan kesehatan, baik secara langsung dan cepat maupun tidak langsung dan secara perlahan.

Untuk menanggulangi masalah tersebut, salah satu alternatif yaitu mengolah air tanah atau air sumur sehingga didapatkan air dengan kualitas yang memenuhi syarat kesehatan dengan merancang sebuah alat filtrasi air.

Teknologi penjernih air yang saat ini ada masih menggunakan *system manual* dimana dalam prosesnya tentu memerlukan pengontrolan lebih terutama pada media *filter*. Kita ketahui bahwa semakin lama *filter* air mengalami kinerja yang semakin lemah akibat kotoran yang semakin bertambah dalam media *filter* sehingga perlu adanya pengurasan dan pencucian *filter*. Apabila *filter* telah jenuh,

maka proses penyaringan menjadi kurang sempurna, kurang cepat, dan kurang efektif.

Oleh karena itu, untuk menjaga performansi alat filtrasi dibutuhkan sebuah *control system* dari alat filtrasi itu sendiri. Sistem tersebut didukung dengan teknologi *digital* untuk *monitoring* dan *control* proses filtrasi serta proses *backwash* sehingga didapatkan hasil yang maksimal.

Dalam judul skripsi ini yaitu “Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga Dengan Kontrol *Fuzzy-PID*” merupakan teknologi pengolahan air dengan *monitoring* kondisi air hasil pengolahan serta *control* kecepatan pompa air yang mengatur debit air yang masuk kedalam tabung *filter* yang dipengaruhi oleh jumlah *particle count* pada media penampungan hasil filtrasi. Selain itu ditambahkan pula proses *backwash* pada media *filter*.

Tujuan teknologi pengolahan air ini secara umum adalah untuk meningkatkan kesehatan masyarakat, khususnya masyarakat yang masih menggunakan air tanah atau air sumur sebagai sumber kebutuhan air bersih selain itu untuk mengaplikasikan pengetahuan mengenai teknologi *control* filtrasi berbasis *digital* agar diperoleh hasil filtrasi yang efektif dan efisien. Sedangkan sasarannya adalah menyebarluaskan paket teknologi pengolahan air sumur kepada masyarakat yang memerlukan.

Judul skripsi ini juga didukung beberapa penelitian yang telah dilakukan sebelumnya dengan judul “Perancangan Kontrol PID Untuk Mengatur Tingkat Kejernihan Air Pada Sistem Pengolahan Air Bersih Skala Rumah Tangga” (Zaid Marhi Nugraha, 2014). Metode yang digunakan ialah menganalisa pengaruh laju aliran air pada sistem pengolahan air bersih skala rumah tangga untuk mengetahui pada laju aliran berapa sistem pengolahan air bersih mampu menghasilkan tingkat kejernihan air tertentu. Dalam sistemnya LDR digunakan sebagai sensor elemen *feedback*.

Penelitian lain dengan judul “Pengontrolan Tekanan Air Dan Pengaturan Sistem *Backwash* Pada *Plant* Pengolahan Air Limbah Teknik Elektro Universitas Diponegoro” (Arif Widagdo, 2013). Metode yang digunakan ialah mengontrolan tekanan air dan mengatur sistem *backwash* pada *Plant* pengolahan air limbah

kemudian dilakukan beberapa pengujian yang bertujuan untuk mengetahui respon sistem. Pengujian tersebut dilakukan dengan memvariasikan nilai K_p dengan nilai *setpoint* yang sama dan juga menaikkan *setpoint* dengan nilai K_p terbaik. Serta beberapa jurnal lain juga mendukung dalam perancangan alat pada judul skripsi ini.

1.1 Rumusan Masalah

Dari latar belakang dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara merancang *plant water treatment* skala rumah tangga ?
2. Bagaimana menerapkan *fuzzy*-PID dalam *control* kecepatan pompa air ?
3. Bagaimana cara menerapkan sensor untuk sistem kendali kejernihan air hasil filtrasi pada *plant water treatment* skala rumah tangga ?

1.2 Tujuan

Tujuan dari penelitian ini dapat dirumuskan sebagai berikut:

1. Merancang *plant water treatment* skala rumah tangga.
2. Menerapkan kontrol *fuzzy*-PID dalam *control* kecepatan pompa air.
3. Merancang sensor untuk sistem kendali kejernihan air hasil filtrasi pada *plant water treatment* skala rumah tangga.

1.3 Manfaat

Manfaat yang dapat diperoleh dari judul penelitian pembuatan alat ini diantaranya ialah:

1. Meningkatkan kesehatan masyarakat, khususnya masyarakat yang masih menggunakan air tanah atau air sumur sebagai sumber kebutuhan air bersih.
2. Mengembangkan teknologi *digital* dan otomasi pada alat penjernih air.
3. Unit alat pengolahan air ini dapat digunakan untuk meningkatkan kualitas air sumur atau air tanah dengan sistem *Backwash* dan *monitoring* kondisi air.
4. Mengaplikasikan pengetahuan mengenai teknologi *control* filtrasi berbasis *digital* agar diperoleh hasil filtrasi yang efektif dan efisien.

1.4 Batasan Masalah

Batasan masalah digunakan untuk membatasi masalah-masalah diluar konsep dari penelitian ini. Batasan masalah tersebut diantaranya ialah:

1. Air yang dihasilkan tidak untuk langsung diminum.
2. Sensor yang digunakan ialah sensor kejernihan air dan sensor *flowmeter*.
3. Air yang digunakan adalah air tidak memiliki tingkat kekeruhan yang sangat tinggi.
4. Tidak membahas media filter secara detail.
5. *Monitoring* debit aliran air hanya pada bagian air yang masuk pada media filtrasi.
6. Proses *monitoring* hanya pada Tingkat kejernihan (*particle count*) air.
7. Proses *backwash* dilakukan *manual* dengan menekan tombol *backwash* dan mengkonfigurasi posisi *valve-valve* yang ada pada PVC.
8. Sistem berupa *prototype*.

1.5 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penelitian

Berisi tentang metode penelitian dan perancangan alat.

BAB IV Hasil dan Pembahasan

Berisi tentang proses pengujian alat, pengambilan data, dan analisa data yang kemudian dimasukkan dalam pembahasan.

BAB V Kesimpulan dan Saran

Berisi tentang kesimpulan dan saran dari alat yang sudah dibuat.

BAB 2. TINJAUAN PUSTAKA

2.1 Filtrasi Air

Di Indonesia standar air minum yang berlaku dibuat pada tahun 1975 yang kemudian diperbarui pada tahun 1990 dan tahun 2002. Dimana dalam standar tersebut dimuat.

Air minum adalah air yang melalui proses pengolahan atau tanpa proses pengolahan yang memenuhi syarat kesehatan dan dapat langsung diminum. Hal inilah yang secara prinsip membedakan kualitas yang harus dimiliki antar air bersih dan air minum. Kualitas air minum setingkat lebih tinggi daripada kualitas air bersih ditinjau dari beberapa komponen pendukungnya. Agar air dapat dikategorikan sebagai air minum maka dipersyaratkan harus memenuhi ketentuan pemerintah berdasarkan peraturan Menteri Kesehatan Republik Indonesia No. 416/MENKES/PER/IX/2002, yang merupakan Standar Nasional Indonesia (SNI) air minum.

Di dalam Permenkes tersebut, ada puluhan jenis parameter yang diatur. Tiap jenis parameter diatur berbeda untuk air minum dan air bersih. secara fisik, keduanya tidak boleh berbau dan tidak boleh berasa. Untuk warna diukur dengan skala TCU (*True Color Unit*), maksimal: air bersih 50 TCU, air minum 5 TCU. Kejernihan diukur dengan skala NTU (*Nephelometric Turbidity Unit*), maksimal: air bersih 25 NTU, air minum 5 NTU.

2.2 Filter Pasir Cepat

Filtrasi adalah suatu proses pemisahan zat padat dari *fluida* yang membawanya menggunakan suatu medium berpori atau bahan berpori lain untuk menghilangkan sebanyak mungkin zat padat halus yang tersuspensi dan koloid. Pada pengolahan air minum, Filtrasi digunakan untuk menyaring air dari proses koagulasi-flokulasi-sedimentasi sehingga dihasilkan air minum yang berkualitas tinggi. Selain mereduksi kandungan zat padat, filtrasi dapat pula mereduksi kandungan bakteri, menghilangkan warna, rasa, bau, besi dan mangan. Filtrasi dibutuhkan untuk sebagian besar air permukaan sebagai pencegah transmisi dari

waterborne diseases. Walaupun dewasa ini desinfeksi merupakan cara yang lebih utama untuk mencegah transmisi tersebut, filtrasi dapat membantu mengurangi proses desinfeksi secara signifikan dan meningkatkan efisiensi dari proses desinfeksi.

Secara umum filtrasi berdasarkan kecepatan penyaringan, dibagi menjadi 2 yaitu;

1. Saringan Pasir Lambat (SSF)
2. Saringan Pasir Cepat (RSF)

Saringan Pasir Cepat merupakan salah satu jenis unit filtrasi yang mampu menghasilkan debit air yang lebih banyak dibandingkan *Slow Sand Filter*, namun kurang efektif untuk mengatasi bau dan rasa yang ada pada air yang disaring. Selain itu, debit air yang cepat menyebabkan lapisan bakteri yang berguna untuk menghilangkan patogen tidak terbentuk sebaik pada tipe *Slow Sand Filter*, sehingga membutuhkan proses desinfeksi yang lebih intensif. Perbedaan utama dari RSF dan SSF adalah bahwa pada SSF arah aliran airnya dari atas ke bawah, sedangkan pada RSF dari bawah ke atas (*up flow*). Selain itu pada RSF umumnya dapat melakukan *backwash* atau pencucian saringan tanpa membongkar keseluruhan saringan.

Media *filter* dapat tersusun dari pasir silika alami, anthrasit, atau pasir garnet. Media ini umumnya memiliki variasi dalam ukuran, bentuk, dan komposisi kimia. RSF dapat menggunakan media tunggal, media ganda, atau multi media. Pada media tunggal digunakan pasir kwarsa saja, media ganda digunakan pasir kwarsa dan antrasit, multi media digunakan pasir kwarsa, anthrasit, dan karbon aktif. RSF memiliki ukuran media pasir berkisar antara 0,5-2,0 mm, dengan laju aliran 5-15 m/jam dan waktu operasi berkisar antara 1-3 hari (Risky Permana, 2011).

2.3 Tipe Filtrasi *Declining Rate*

Menurut tipe filtrasi berdasarkan *system control* kecepatan *filter* dikelompokkan menjadi 2 yaitu:

1. *Constant Rate*: debit hasil filtrasi konstan pada level tertentu. Hal ini dilakukan dengan memberikan kebebasan kenaikan level muka air media *filter*.
2. *Declining Rate* atau *constant head* debit hasil proses filtrasi menurun seiring dengan proses filtrasi, atau level muka air media *filter* dirancang pada nilai yang tetap (Ketut Sumada, 2012).

2.4 Proses *Backwash*

Pengertian '*Backwash*' dalam *water treatment* adalah membalik arah masuknya air ke dalam tabung *filter* air.

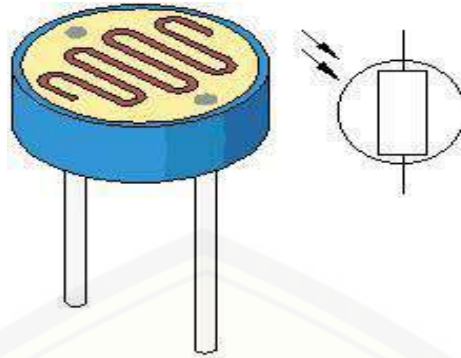
Pada kondisi normal dimana air masuk dari atas *filter* (untuk tipe *filter* yang *vertical*) kemudian menembus *filter* media (pasir atau karbon aktif) kemudian keluar menuju proses berikutnya. Dengan berjalanya waktu dan karena pemakaian dari *filter* itu sendiri, media *filter* menjadi kotor oleh polutan – polutan dalam air yang terperangkap di dalamnya.

Untuk mengembalikan kondisi *filter* media seperti semula maka di perlukan pembersihan / pencucian media *filter* secara berkala. Proses ini yang dinamakan *Backwashing* yaitu mencuci media *filter* tanpa harus mengeluarkan media *filter* itu sendiri dari dalam tabung *filter* (*vesell*).

Pembuangan sedimentasi (*backwash*) secara sederhana dapat dijelaskan dengan menukar arah masuk air (*intake*) dengan (*outlet*) arah keluarnya (Riezky Rafaeel, 2014).

2.5 Sensor Kejernihan Air

Sensor Cahaya LDR (*Light Dependent Resistor*) adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada sensor cahaya LDR (*Light Dependent Resistor*) tergantung pada besar kecilnya cahaya yang diterima oleh LDR itu sendiri.



Gambar 2.1. Sensor LDR (Agus Purnama, 2012)

LDR sering disebut dengan alat atau sensor yang berupa resistor yang peka terhadap cahaya. Biasanya LDR terbuat dari cadmium sulfida yaitu merupakan bahan semikonduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya. Resistansi LDR pada tempat yang gelap biasanya mencapai sekitar $10\text{ M}\Omega$, dan ditempat terang LDR mempunyai resistansi yang turun menjadi sekitar $150\ \Omega$. Seperti halnya resistor konvensional, pemasangan LDR dalam suatu rangkaian sama persis seperti pemasangan resistor biasa (Agus Purnama, 2012).

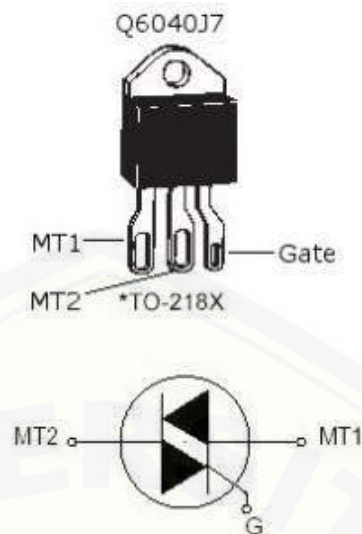
2.6 Pengaturan Kecepatan Motor Pompa AC Dengan Triac

2.6.1 Triac

Triac merupakan komponen semikonduktor yang tersusun atas diode empat lapis berstruktur p-n-p-n dengan tiga p-n *junction*. Triac memiliki tiga buah elektrode, yaitu: *gate*, MT1, MT2. Triac biasanya digunakan sebagai pengendali dua arah (*bi-directional*). Apabila kita akan menggunakan triac dalam pembuatan perangkat atau sistem kontrol elektronik, ada beberapa hal yang harus diketahui dalam memilih triac sebagai berikut.

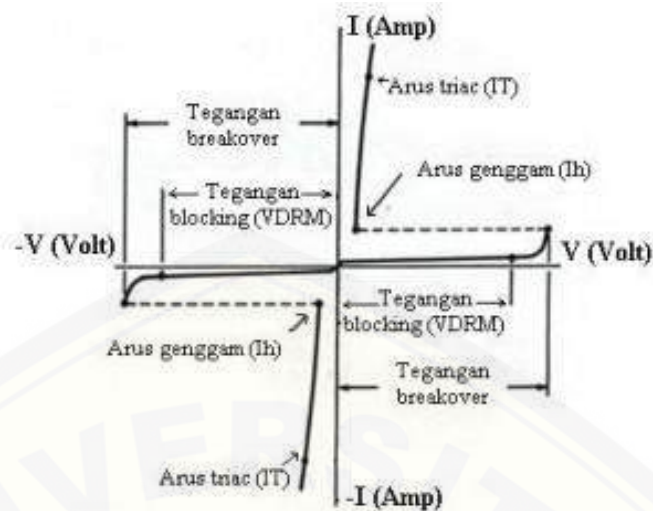
Hal-Hal Yang Perlu Diperhatikan Dalam Memilih Triac:

1. Tegangan *break over* maju dan mundur.
2. Arus maksimal ($I_T\text{ maks}$).
3. Arus genggam minimal ($I_h\text{ min}$).
4. Tegangan dan arus picu *gate* yang diperlukan.
5. Kecepatan pensaklaran tegangan maksimal dV/dt .
6. Tegangan *blocking* triac (V_{DRM}).



Gambar 2.2 Simbol Triac (Agus Purnama, 2012).

Simbol dan bentuk triac akan tersambung (*on*) ketika berada di kuadran I yaitu saat arus *positif* kecil melewati terminal *gate* ke MT1, dan polaritas MT2 lebih tinggi dari MT1, saat triac terhubung dan rangkaian *gate* tidak memegang kendali, maka triac tetap tersambung selama polaritas MT2 tetap lebih tinggi dari MT1 dan arus yang mengalir lebih besar dari arus genggamnya (*holding current/I_h*), dan triac juga akan tersambung saat arus *negatif* melewati terminal *gate* ke MT1, dan polaritas MT1 lebih tinggi dari MT2, dan triac akan tetap terhubung walaupun rangkaian *gate* tidak memegang kendali selama polaritas MT1 lebih tinggi dari MT2. Selain dengan cara memberi pemicuan melalui terminal *gate*, triac juga dapat dibuat tersambung (*on*) dengan cara memberikan tegangan yang tinggi sehingga melampaui tegangan *breakover*-nya terhadap terminal MT1 dan MT2, namun cara ini tidak diizinkan karena dapat menyebabkan triac akan rusak. Pada saat triac tersambung (*on*) maka tegangan jatuh maju antara terminal MT1 dan MT2 sangatlah kecil yaitu berkisar antara 0.5 volt sampai dengan 2 volt (Agus Purnama, 2012).



Gambar 2.3 Kurva Karakteristik Triac (Agus Purnama, 2012).

2.6.2 MOC 3041

Optotriac tipe MOC 3041 ini dilengkapi dengan rangkaian detektor pelintas nol (*Zero Crossing Detector*) yang mampu membuat optotriac ini mulai akan konduksi pada saat siklus tegangan masukannya pada nol. Hal ini akan mencegah terjadinya lonjakan arus yang besar secara tiba-tiba pada beban yang dikendalikan. Keuntungan dengan menggunakan IC ini adalah lebih terjaminnya keamanan rangkaian pengendali dari hubungan langsung terhadap tegangan jala-jala PLN. Hal ini dikarenakan terpisahnya aliran arus antara beban pengendali dengan penggunaan optotriac. Optotriac MOC3041 bekerja pada level tegangan ac antara 200-400 V_{ac} dengan tegangan masukan pada LED 2,3 V_{DC} sedangkan arus kerjanya 200 Ma (Herlambang, 2010).

2.7 Arduino

Arduino Uno adalah *board* berbasis mikrokontroler pada ATmega 328. *Board* ini memiliki 14 *digital input / output pin* (dimana 6 *pin* dapat digunakan sebagai *output PWM*), 6 *input analog*, 16 MHz osilator kristal, koneksi USB, *jack* listrik dan tombol *reset*. *Pin - pin* ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau

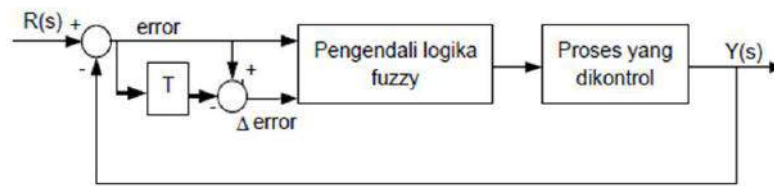
sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya (Arduino, Inc, 2009).

Arduino Uno R3 berbeda dengan semua *board* sebelumnya karena *Arduino Uno R3* ini tidak menggunakan *chip driver* FTDI USB-to-serial. Melainkan menggunakan fitur dari ATmega 16U2 yang diprogram sebagai *konverter* USB-to-serial. *Board* Arduino Uno memiliki fitur – fitur baru sebagai berikut:

- *Pin Out*: menambahkan SDA dan SCL *pin* yang dekat ke *pin* aref dan dua *pin* baru lainnya ditempatkan dekat ke *pin* RESET, dengan I/O REF yang memungkinkan sebagai *buffer* untuk beradaptasi dengan tegangan yang disediakan dari *board* sistem. Pengembangannya, sistem lebih kompatibel dengan *prosesor* yang menggunakan AVR, yang beroperasi dengan 5V dan dengan Arduino karena beroperasi dengan 3,3V. Yang kedua adalah *pin* yang tidak terhubung, yang disediakan untuk tujuan pengembangannya.
- Sirkuit *reset*
- ATmega 16U2 ganti 8U yang digunakan sebagai *konverter* USB-to-serial.

2.8 Pengendali Logika Fuzzy

Kendali logika *fuzzy* bekerja berdasarkan aturan linguistik yang dibuat mirip dengan seorang operator ahli dalam melakukan proses kendali. Pada proses *manual*, kinerja memuaskan atau tidak tergantung dari pengalaman operator tersebut, pengalaman butuh waktu dan *trial-And-error*. Pada kendali logika *fuzzy*, kinerja kendali memuaskan atau tidak juga tergantung dari *trial-And-error*, tidak ada deskripsi yang baku untuk menentukan kendali logika *fuzzy* agar dapat berjalan memuaskan, karena di samping ilmu bidang kendali, logika *fuzzy* masih relatif baru, kendali logika *fuzzy* juga memiliki beberapa parameter yang dapat diatur, yaitu masukan dan keluaran aktual serta semesta pembicaraan, faktor skala dari variabel masukan dan keluaran, tipe dan bentuk fungsi keanggotaan, aturan *fuzzy* yang digunakan, metode implikasi dan agregasi, serta metode defuzzifikasi yang digunakan. Mekanisme kendali logika *fuzzy loop* tertutup ditunjukkan pada Gambar 2.4.



Gambar 2.4 Sistem *loop* tertutup dengan pengendali *fuzzy*

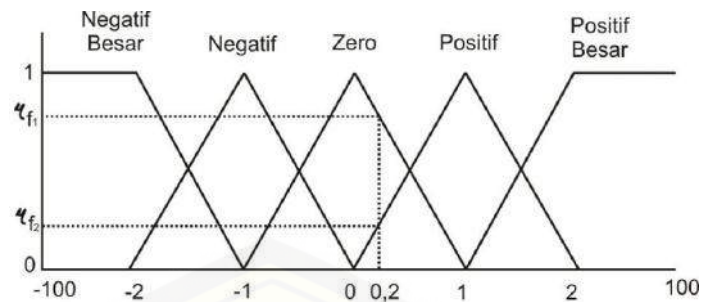
Pada gambar 2.4 terdapat dua masukan berbentuk *crisp* (tegas), yaitu masukan *error* dan perubahan *error* yang diperoleh dari nilai referensi, keluaran *plant*, dan *error* sebelumnya. Dua masukan tersebut akan diolah oleh pengendali logika *fuzzy*. Nilai keluaran yang dihasilkan pengendali logika *fuzzy* berupa *crisp*. Struktur dasar sebuah pengendali logika *fuzzy* ditunjukkan pada Gambar 2.5, yang meliputi empat bagian utama yaitu fuzzifikasi, basis pengetahuan, logika pengambilan keputusan, dan defuzzifikasi.



Gambar 2.5 Struktur dasar pengendali *fuzzy*

2.8.1. Fuzzifikasi (*Fuzzification*)

Fuzzifikasi merupakan sebuah proses dalam kendali *fuzzy* yang berfungsi untuk memetakan masukan data tegas ke dalam himpunan *fuzzy* menjadi nilai *fuzzy* dari beberapa variabel linguistik masukan. Suatu contoh proses fuzzifikasi dimana nilai masukan tegas dipetakan ke dalam fungsi keanggotaan segitiga dengan label (umumnya berupa variabel linguistik *fuzzy*) sebagai berikut: *Negatif Besar*, *Negatif*, *Zero*, *Positif* dan *Positif Besar*. Proses fuzzifikasi ditunjukkan pada Gambar 2.6.



Gambar 2.6 Proses fuzzifikasi

Gambar 2.6 menunjukkan variabel yang digunakan sebagai contoh adalah besar perubahan pengukuran data dengan jangkauan antara -100 sampai 100. Sistem *fuzzy* tersebut memiliki 5 *membership functions* yaitu *Negatif Besar*, *Negatif*, *Zero*, *Positif* dan *Positif Besar*. Hasil dari proses fuzzifikasi nilai *crisp* 0.2 mempunyai derajat keanggotaan μ_{f1} pada *membership functions* *Zero* dan μ_{f2} pada *membership functions* *Positif*.

2.8.2 Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan terdiri dari basis data dan basis aturan. Basis data mendefinisikan himpunan *fuzzy* atas ruang-ruang masukan dan keluaran. Basis aturan berisi aturan-aturan kendali *fuzzy* yang digunakan untuk pengendalian proses. Pembentukan basis data mencakup perancangan fungsi keanggotaan untuk masing-masing variabel masukan dan keluaran, pendefinisian semesta pembicaraan dan penentuan variabel linguistik setiap variabel masukan dan keluaran. Basis aturan kendali *fuzzy* adalah kumpulan aturan-aturan kendali *fuzzy* yang dibuat berdasarkan pengetahuan manusia dalam pengendalian suatu sistem. Aturan yang ditetapkan digunakan untuk menghubungkan antara variabel-variabel masukan dan variabel-variabel keluaran. Aturan ini berbentuk, "JIKA – MAKA" (*IF – THEN*), sebagai contoh adalah:

Aturan 1: JIKA x adalah A1 DAN y adalah B1 MAKA z adalah C1.

Aturan 2: JIKA x adalah A2 DAN y adalah B2 MAKA z adalah C2.

Aturan 3: JIKA x adalah A3 DAN y adalah B3 MAKA z adalah C3.

Aturan i: JIKA x adalah Ai DAN y adalah Bi MAKA z adalah Ci.

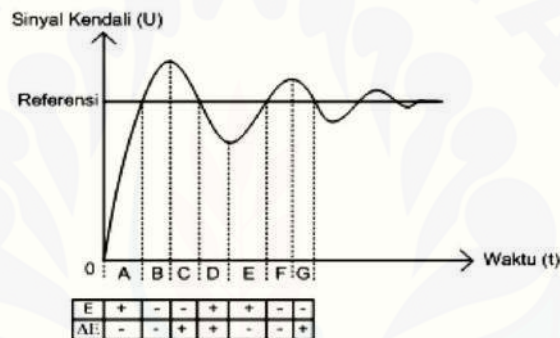
dengan:

A_i ($i = 1, 2, 3, \dots$) adalah himpunan *fuzzy* untuk variabel masukan x .

B_i ($i = 1, 2, 3, \dots$) adalah himpunan *fuzzy* untuk variabel masukan y .

C_i ($i = 1, 2, 3, \dots$) adalah himpunan *fuzzy* untuk variabel keluaran z .

Metode yang digunakan dalam penentuan aturan *fuzzy* adalah menggunakan pendekatan *heuristis*. Pendekatan *heuristis* menggunakan pengetahuan pakar dan analisis perilaku sistem terkendali untuk mendapatkan aturan–aturan *fuzzy*. Pada Gambar 2.7 terlihat grafik tanggapan sistem terhadap masukan undak satuan dengan berbagai macam keadaan polaritas *error* ($error = E$) dan perubahan *error* ($Change\ of\ Error = \Delta E$).



Gambar 2.7 Tanggapan sistem terhadap masukan undak satuan (Rohmat Hidayat, 2014)

Dari berbagai macam keadaan polaritas E dan ΔE tersebut, dapat didesain aturan–aturan *fuzzy* yang dapat memberikan keluaran sinyal kendali (U) untuk mengurangi lonjakan tanggapan sistem, waktu naik menuju referensi, osilasi, serta melakukan pengereman tanggapan sistem agar tanggapan sistem tidak terlalu cepat. Jumlah aturan *fuzzy* yang digunakan tergantung kepada beberapa faktor, seperti kinerja sistem terkendali, efisiensi komputasi, perilaku operator, pemilihan variabel linguistik. Empat contoh perumusan aturan ditunjukkan di bawah ini.

1. Aturan tipe I (untuk mengurangi lonjakan tanggapan sistem): Jika E adalah *Negatif* dan ΔE adalah *Negatif* Maka U adalah *Negatif*.
2. Aturan tipe II (untuk mengurangi waktu naik): Jika E adalah *Positif* dan ΔE adalah *Negatif* Maka U adalah *Positif*.

3. Aturan tipe III (untuk melakukan pengereman tanggapan sistem): Jika E adalah *Negatif* dan ΔE adalah *Positif* Maka U adalah Nol.
4. Aturan tipe IV (untuk mengurangi osilasi): Jika E adalah *Positif* dan ΔE adalah *Positif* Maka U adalah *Positif*.

Secara umum, prinsip-prinsip yang dipegang untuk mendapatkan aturan-aturan *fuzzy* adalah:

- A. Jika keluaran sudah mencapai titik referensi dan perubahan kesalahan adalah nol, maka keluaran sistem kendali tetap dijaga agar konstan.
- B. Jika keluaran menyimpang dari titik referensi, maka aksi kendali tergantung pada polaritas *error* dan perubahan *error*. Jika *error* dapat diperbaiki dengan aksi kendali terakhir, maka aksi kendali tetap dijaga konstan atau hampir konstan. Selain keadaan itu, keluaran kendali berubah-ubah sesuai dengan *error* dan perubahan *error* serta tanggapan sistem terhadap referensi
- C. Mekanisme Pengambilan Keputusan (*Fuzzy Reasoning*)

Pada tugas akhir ini pengambilan keputusan menggunakan metode Sugeno karena metode ini memberikan efisiensi komputasi dimana metode ini bagian *consequent* direpresentasikan menggunakan sebuah konstanta (C) sehingga mempercepat perhitungan.

$$\text{If } x \text{ is } A \text{ AND } y \text{ is } B \text{ then } z = C \dots\dots\dots (2.1)$$

Keterangan:

x = variabel masukan 1

A = himpunan *fuzzy* dalam semesta pembicaraan U

y = variabel masukan 2

B = himpunan *fuzzy* dalam semesta pembicaraan V

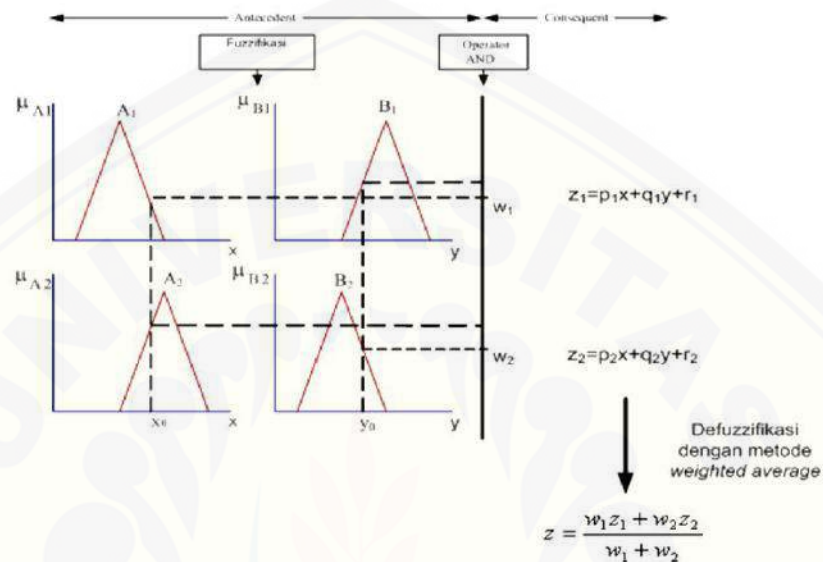
z = variabel keluaran

C = konstanta tegas pada semesta pembicaraan W

Implikasi dalam metode ini menggunakan nilai minimal antara keluaran bagian *antacent* dengan menggunakan operator *AND*. Agregasi dalam metode Sugeno merupakan pengumpulan semua fungsi keanggotaan hasil dari implikasi. Proses implikasi metode Sugeno ditunjukkan pada gambar 2.8 dengan memisalkan bahwa

fungsi keanggotaan fungsi keanggotaan masukan adalah fungsi segitiga dan mempunyai 2 aturan *fuzzy*, yaitu:

- Jika x adalah A1 dan y adalah B1 maka z adalah Z1
- Jika x adalah A2 dan y adalah B2 maka z adalah Z2



Gambar 2.8 Proses Pengambilan Keputusan Metode Sugeno (Rohmat Hidayat, 2014)

Langkah pertama pengambilan keputusan metode Sugeno adalah melakukan proses fuzzifikasi untuk memetakan data *crisp* masukan *error* dan perubahan *error* ke dalam data *fuzzy* sesuai dengan tipe dan bentuk fungsi keanggotaan. Langkah kedua adalah melakukan proses terhadap kedua data *fuzzy* tersebut dengan operator *AND* yang mengambil nilai paling minimal dari dua data tersebut. Langkah ketiga dengan implikasi memotong derajat keanggotaan variabel keluaran pada nilai keluaran setelah melalui operator *AND*.

2.8.3 Defuzzifikasi

Defuzzifikasi dapat didefinisikan sebagai proses perubahan besaran *fuzzy* yang ditunjukkan dalam bentuk himpunan-himpunan *fuzzy* keluaran dengan fungsi keanggotaan untuk mendapatkan kembali bentuk tegasnya (*crisp*). Hal ini diperlukan karena *plant* hanya mengenal nilai tegas sebagai besaran sebenarnya untuk regulasi prosesnya. Ada beberapa macam proses defuzzifikasi diantaranya:

1. Metode Rerata Berbobot (*Center Average Defuzifier*)

Metode ini dapat dihitung dengan persamaan 2.2.

$$\frac{\sum_{i=1}^N w_i \cdot z_i}{\sum_{i=1}^N w_i} \dots\dots\dots (2.2)$$

2. Metode Titik Tengah (*Center of Area*)

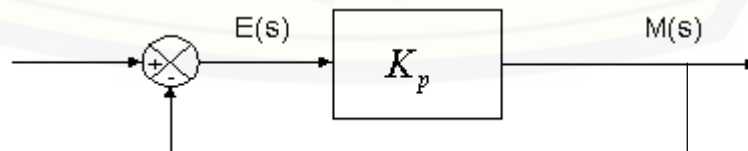
Metode ini dapat dihitung menggunakan persamaan 2.3.

$$\frac{\int \mu(z)zdz}{\int \mu(z)dz} \dots\dots\dots (2.3)$$

2.9 Kontrol PID

Kontroler proposional memiliki keluaran yang sebanding/proposional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya) (Sharon, 1992, 19). Secara lebih sederhana dapat dikatakan, bahwa keluaran *controller* proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

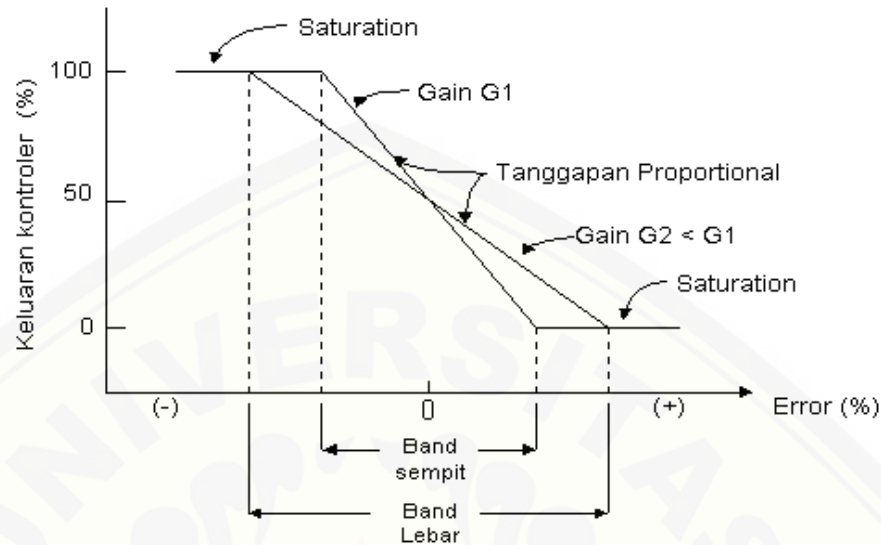
Gambar 2.9 menunjukkan diagram blok yang menggambarkan hubungan antara besaran *setting*, besaran aktual dengan besaran keluaran *controller* proporsional. Sinyal keasalahan (*error*) merupakan selisih antara besaran setTing dengan besaran aktualnya. Selisih ini mempengaruhi *controller*, untuk mengeluarkan sinyal *positif* (mempercepat pencapaian harga *setting*) atau *negatif* (memperlambat tercapainya harga yang diinginkan).



Gambar 2.9 Diagram Blok Kontroler Proporsional (Gunterus, 1994, 6-24).

Kontroler proporsional memiliki 2 parameter, pita proporsional (*proportional band*) dan konstanta proporsional. Daerah kerja *controller* efektif

dicerminkan oleh Pita proporsional (Gunterus, 1994, 6-24), sedangkan konstanta proporsional (K_p) menunjukkan nilai faktor penguatan terhadap sinyal kesalahan.



Gambar 2.10 *Proportional band* dari kontroler proporsional tergantung pada penguatan (Johnson, 1988, 372).

Gambar 2.5 menunjukkan grafik hubungan antara PB, keluaran kontroler dan kesalahan yang merupakan masukan *controller*. Ketika konstanta proporsional bertambah semakin tinggi, pita proporsional menunjukkan penurunan yang semakin kecil, sehingga lingkup kerja yang dikuatkan semakin sempit (Johnson, 1988, 372).

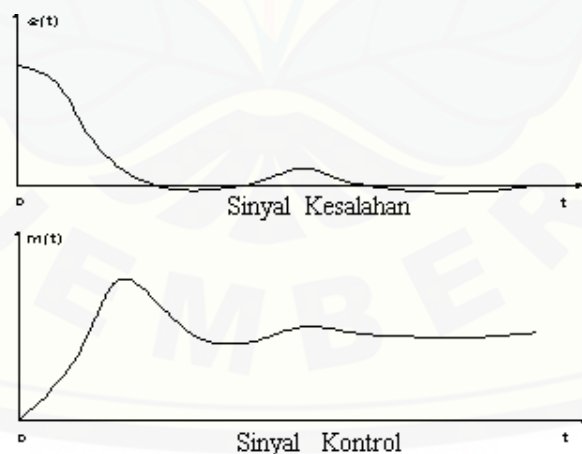
Ciri-ciri kontroler proporsional harus diperhatikan ketika kontroler tersebut diterapkan pada suatu sistem. Secara eksperimen, pengguna *controller* proporsional harus memperhatikan ketentuan-ketentuan berikut ini:

1. Kalau nilai K_p kecil, kontroler proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. Kalau nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan *steady state*.
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosilasi (Pakpahan, 1988, 193).

Controller integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol. Kalau sebuah plant tidak memiliki unsur *integrator* ($1/s$), *controller* proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol. Dengan *controller* integral, respon sistem dapat diperbaiki, yaitu mempunyai kesalahan keadaan mantapnya nol.

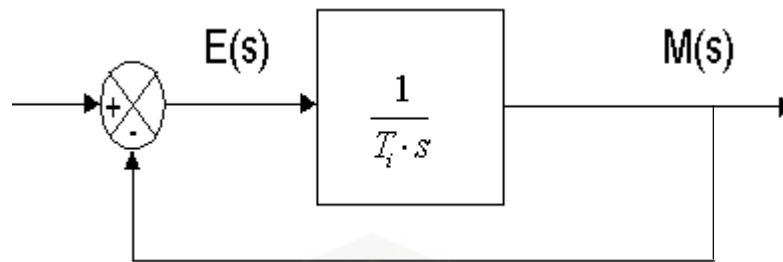
Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran *controller* sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan (Rusli, 1997, 18). Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran dapat menjaga keadaan seperti sebelum terjadinya perubahan masukan.

Sinyal keluaran kontroler integral merupakan luas bidang yang dibentuk oleh kurva kesalahan penggerak. Sinyal keluaran bernilai sama dengan harga sebelumnya ketika sinyal kesalahan berharga nol. Gambar 2.11 (Ogata, 1997, 236) menunjukkan contoh sinyal kesalahan yang disulutkan ke dalam *controller* integral dan keluaran *controller* integral terhadap perubahan sinyal kesalahan tersebut.



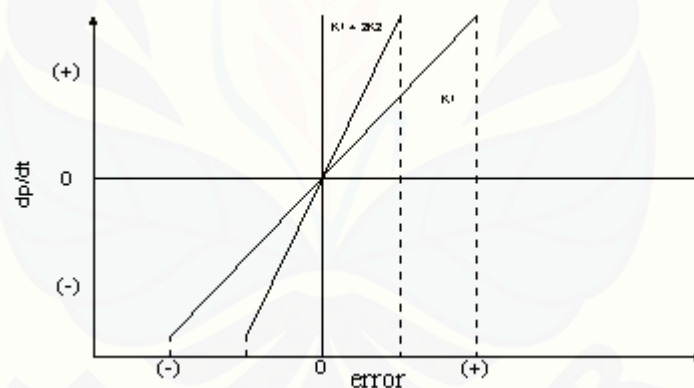
Gambar 2.11 Kurva Sinyal Kesalahan $E(T)$ Terhadap T Dan Kurva $U(T)$ Terhadap T Pada Pembangkit Kesalahan Nol (Ogata, 1997, 236).

Gambar 2.12 Menunjukkan diagram blok antara besaran kesalahan dengan keluaran suatu *controller* integral.



Gambar 2.12 Diagram Blok Hubungan Antara Besaran Kesalahan Dengan *Controller* Integral (Ogata, 1997, 236).

Pengaruh perubahan konstanta integral terhadap keluaran integral ditunjukkan oleh Gambar 2.13. Ketika sinyal kesalahan berlipat ganda, maka nilai laju perubahan keluaran kontroler berubah menjadi dua kali dari semula. Jika nilai konstanta *integrator* berubah menjadi lebih besar, sinyal kesalahan yang relatif kecil dapat mengakibatkan laju keluaran menjadi besar (Johnson, 1993, 375).



Gambar 2.13 Perubahan Keluaran Sebagai Akibat Penguatan Dan Kesalahan (Johnson, 1993, 375).

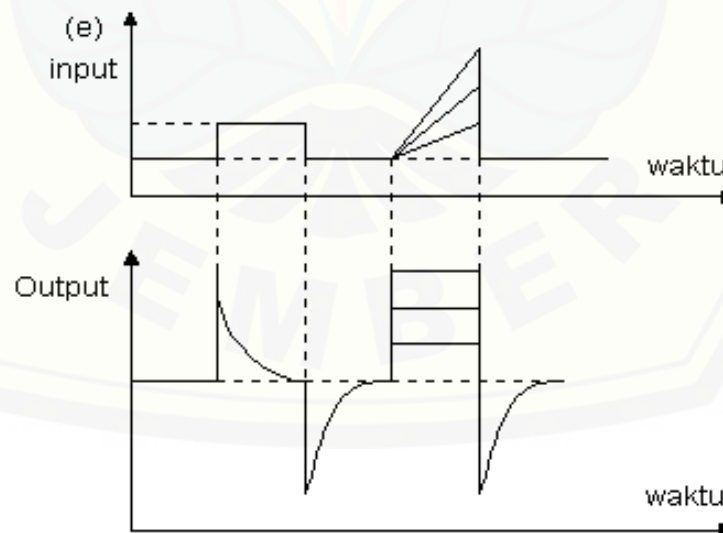
Ketika digunakan, kontroler integral mempunyai beberapa karakteristik berikut ini:

1. Keluaran kontroler membutuhkan selang waktu tertentu, sehingga kontroler integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran kontroler bertahan pada nilai sebelumnya.

3. Jika sinyal kesalahan tidak berharga nol, keluaran menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i (Johnson, 1993, 376).
4. Konstanta integral K_i yang berharga besar mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta K_i mengakibatkan peningkatan osilasi dari sinyal keluaran kontroler (Guterus, 1994, 7- 4).

Keluaran kontroler diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, mengakibatkan perubahan yang sangat besar dan cepat.

Gambar 2.14 menyatakan hubungan antara sinyal masukan dengan sinyal keluaran kontroler diferensial. Ketika masukannya tidak mengalami perubahan, keluaran kontroler juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*), keluaran menghasilkan sinyal berbentuk *impuls*. Jika sinyal masukan berubah naik secara perlahan (fungsi *ramp*), keluarannya justru merupakan fungsi *step* yang besar magnitudnya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan faktor konstanta diferensialnya T_d (Guterus, 1994, 8-4).



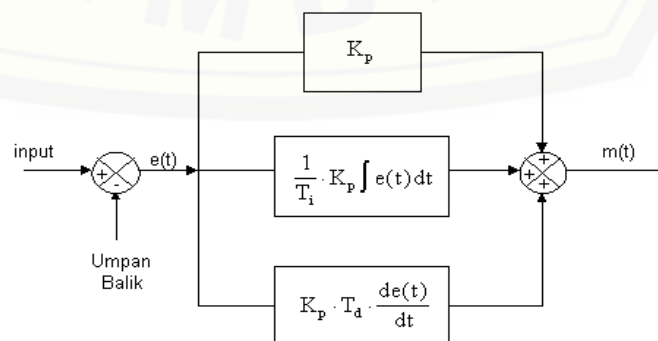
Gambar 2.14. Kurva waktu hubungan *input-output* kontroler diferensial (Guterus, 1994).

Karakteristik kontroler diferensial adalah sebagai berikut:

1. Kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukannya (berupa sinyal kesalahan).
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan kontroler tergantung pada nilai T_d dan laju perubahan sinyal kesalahan. (Powel, 1994, 184).
3. Kontroler diferensial mempunyai suatu karakter untuk mendahului, sehingga kontroler ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi kontroler diferensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem (Ogata, 1997, 240).

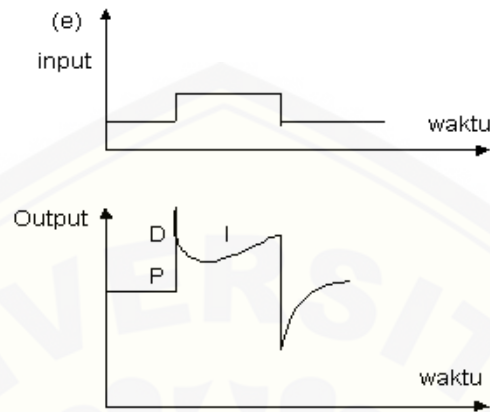
Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler diferensial tidak pernah digunakan tanpa ada kontroler lain sebuah sistem (Sutrisno, 1990, 102).

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proposional *plus* integral *plus* diferensial (*controller* PID). Elemen-elemen *controller* P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Guterus, 1994, 8-10).



Gambar 2.15 Diagram Blok Kontroler PID Analog (Guterus, 1994, 8-10).

Gambar 2.16 Menunjukkan Diagram Blok Kontroler PID. Keluaran *controller* PID merupakan jumlahan dari keluaran kontroler proporsional, keluaran kontroler integral. Gambar 2.16 menunjukkan hubungan tersebut.



Gambar 2.16 Hubungan dalam fungsi waktu antara sinyal keluaran dengan masukan untuk *controller* PID (Gunterus, 1994, 8-10).

Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , T_i , dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diatur lebih menonjol dibanding yang lain. Konstanta yang menonjol dapat memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.10 Desain Kontrol Logika *Fuzzy*-PID

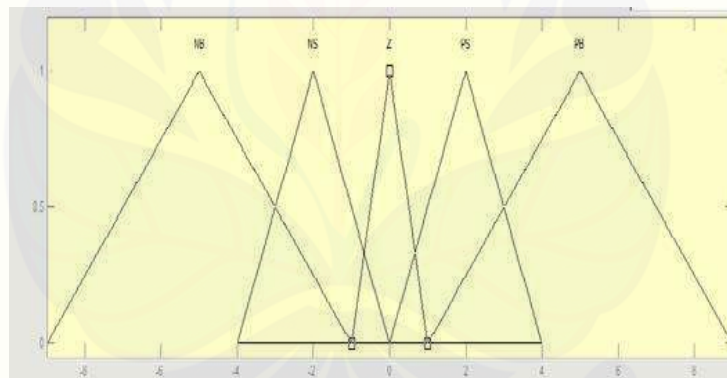
Sistem *fuzzy* adalah inti dari sistem pengendalian ini. Sistem ini terdiri dari fuzzifikasi, basis aturan, inferensi *fuzzy*, dan defuzzifikasi. fuzzifikasi memetakan nilai *input* menjadi nilai *fuzzy* yaitu derajat keanggotaan *error* yang bernilai 0 sampai dengan 1 yang akan dikalikan dengan nilai *gain* yang sesuai dengan basis aturan yang menghubungkan nilai *input* dengan nilai *gain* pengendalian. Inferensi *fuzzy* adalah proses pengambilan keputusan untuk mendapatkan *gain* parameter proporsional, integral dan derivatif berdasarkan rancangan basis aturan. Defuzziikasi adalah kebalikan dari fuzzifikasi yaitu mengubah nilai *fuzzy* menjadi nilai *output* yang tegas atau *crisp*. Dimana hasil dari *fuzzy*-PID adalah nilai sinyal

kontrol yang sebanding dengan nilai PWM tegangan pompa (Fista Rachma Danianta, 2013).

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu metode yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

Fungsi keanggotaan *input* sistem berupa *triangular*. *Error* dapat dicari dari nilai derajat keanggotaannya. Dan perubahan *error* bersama dengan nilai *error* digunakan sebagai penentu keputusan yang diambil oleh sistem pengendali.

Ketika fungsi keanggotaan *error* telah ditentukan, selanjutnya dapat dicari nilai derajat keanggotaan dari *error* yang nantinya dapat digunakan sebagai pengali dari *gain* PID (Fista Rachma Danianta, 2013).



Gambar 2.17 Fungsi Keanggotaan (Fista Rachma Danianta, 2013).

Fungsi keanggotaan dari *error* berupa fungsi keanggotaan segitiga (lihat Gambar 2.17). Bentuk dari fungsi keanggotaan segitiga ditentukan oleh tiga parameter {a,b,c} (dengan $a < b < c$). Adapun persamaan yang dapat digunakan adalah:

$$\text{Segitiga} = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \dots\dots\dots (2.4)$$

BAB 3. METODE PENELITIAN

Langkah-langkah dalam perancangan alat ini diantaranya *Studi literature*, perancangan alat, pembuatan alat, pengujian alat dan pengambilan kesimpulan.

3.1 Study literature

Study literature dilakukan dengan mengacu pada spesifikasi alat yang dirancang dan dasar teori pendukung yang diperlukan untuk merealisasikan alat. *Study literature* yang dilakukan meliputi studi mengenai teori dasar, mengenai perancangan sistem *hardware* dan *software* yang meliputi sensor kejernihan air, sensor *flowmeter*, *driver* pompa air, pompa air, arduino uno, LCD *Keypad*, kontrol *Fuzzy-PID*.

3.2 Tempat & Waktu Penelitian

3.2.1 Tempat Penelitian

Penelitian dilaksanakan di Laboratorium Elektronika Terapan, Jurusan Elektro, Fakultas Teknik. Pengujian alat dilakukan di rumah peneliti Desa Klenang Lor, Kecamatan Banyuwanyar, Kabupaten Probolinggo.

3.2.2 Waktu Penelitian

Waktu penelitian dilaksanakan selama kurang lebih 3 bulan yaitu dari bulan September 2015 sampai bulan November 2015.

3.3 Alat dan Bahan

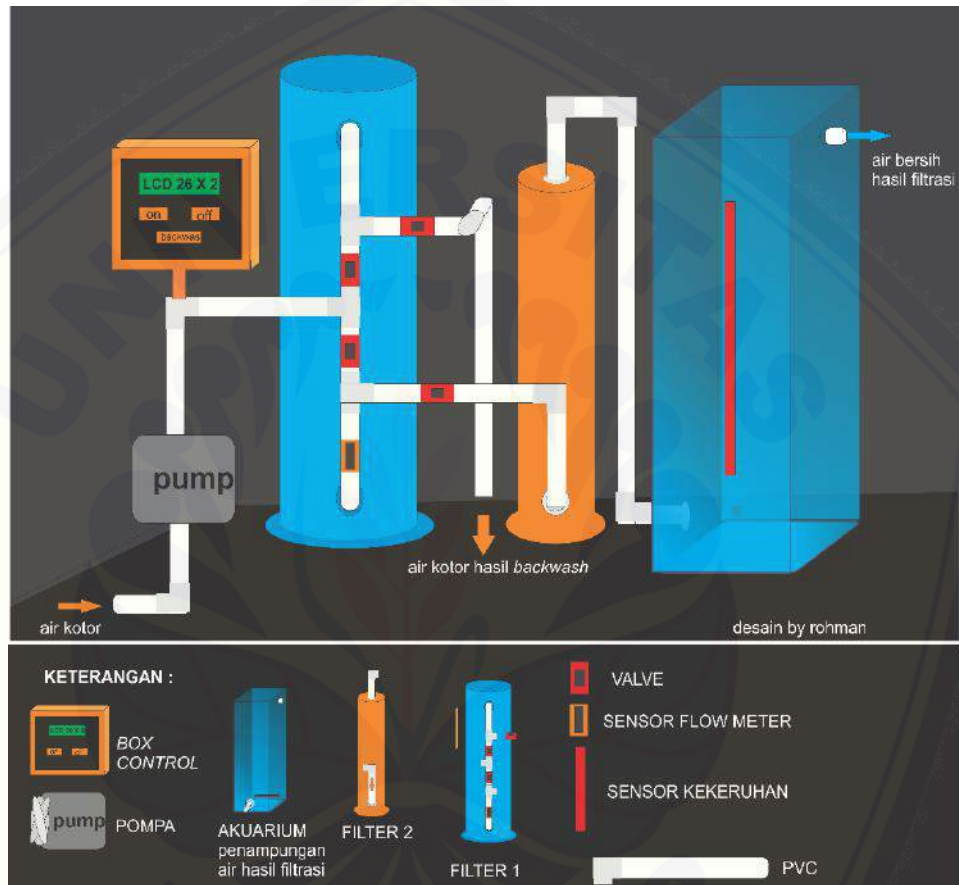
3.3.1 Hardware

1. Arduino UNO R3
2. *Power Supply*
3. LCD 2x16
4. Rangkaian LCD
5. *Driver* Pompa
6. Rangkaian Sensor Kejernihan
7. Rangkaian Sensor *Flowmeter*

3.3.2 Software

1. Arduino IDE
2. PCB *express*

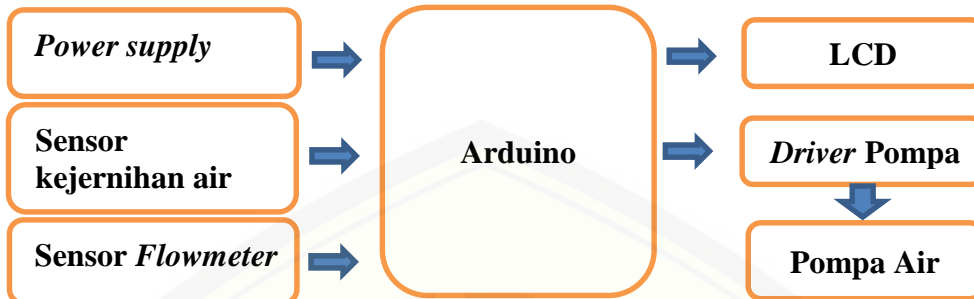
3.4 Perancangan alat



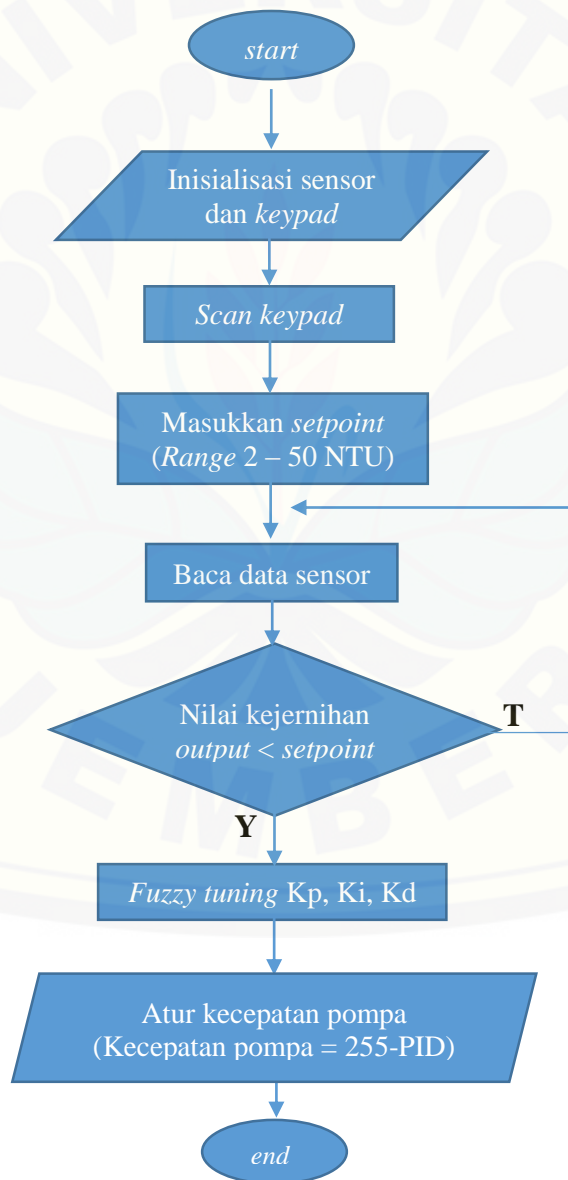
Gambar 3.1 Desain Perancangan Alat

Pada gambar 3.1, pada *box control* terdapat rangkaian elektronika yaitu: Arduino uno, rangkaian *power supply*, dan *driver* pompa. Pada tabung *filter 1* dan *filter 2* menggunakan pipa besi dengan ukuran tabung *filter 1* = 6 *inchi*, tabung *filter 2* = 3 *inchi*. Pada tabung *filter 1* terdapat media *filter* berupa mangan zeolit, silica, karbon aktif, sedangkan pada tabung *filter 2*, media *filter*nya berupa kasa. Kemudian pompa air dengan daya 125 watt dan tegangan kerja 220 volt digunakan untuk mengalirkan air pada proses penjernihan. Sensor kejernihan menggunakan LED dan LDR yang berada pada akuarium penampungan air hasil filtrasi.

3.5 Diagram Blok dan Flowchart



Gambar 3.2 Diagram Blok Sistem



Gambar 3.3 Flowchart Sistem

Pada gambar 3.2, diagram blok sistem terdiri dari *input system* dan *output system* serta *power supply*. Pada bagian *input system* berupa sensor kejernihan dan sensor *flowmeter*, sedangkan *output system* berupa LCD dan *driver* motor AC untuk pompa air. Terdapat 2 sensor kejernihan yaitu sensor kejernihan pada air *input* dan sensor kejernihan pada air *output*. Sensor kejernihan air *input* digunakan untuk mendeteksi tingkat kejernihan air yang difiltrasi, dan sensor kejernihan air *output* digunakan untuk mendeteksi tingkat kejernihan air hasil proses filtrasi. Pada bagian *output*, LCD digunakan untuk menampilkan data, sedangkan *driver* pompa digunakan untuk menggerakkan pompa air. Kemudian *power supply* digunakan sebagai sumber tegangan pada sistem yang meliputi rangkaian sensor dan arduino.

Pada gambar *flowchart* 3.3, pada awal *start* sistem dimulai dengan proses inialisasi sensor (sensor kejernihan 1, sensor kejernihan 2, sensor *flowmeter*) dan *keypad*, kemudian *scan keypad*. Selanjutnya yaitu memasukkan nilai *setpoint* dengan *range* masukan *setpoint* dari 2 sampai 50 NTU, apabila nilai sensor terbaca maka sistem akan memulai proses filtrasi. Kemudian nilai K_p , K_i , K_d akan diatur sesuai dengan nilai *error* dan perubahan *error* kejernihan *output* air hasil filtrasi. Nilai keluaran PID akan dirubah dalam bentuk PWM dan dipetakan pada *range* 0 – 255. Untuk mengontrol kecepatan pompa digunakan fungsi *transfer* sebagai berikut:

$$CO = 255 - Out_PID \dots\dots\dots(3.1)$$

Keterangan:

- CO : nilai PWM *output* untuk *driver* pompa
 255 : nilai PWM maksimal arduino
 Out_PID : nilai keluaran PID

3.6 Realisasi pembuatan alat

Pada tugas akhir ini perangkat keras meliputi arduino uno sebagai pengendali dan sensor kejernihan sebagai sensor pendeteksi kejernihan air. Sensor *flowmeter* sebagai pendeteksi aliran air dan *driver* PWM *digital dimmer* sebagai kendali tegangan AC pada motor pompa air, sedangkan *push button* sebagai unit masukan untuk memberikan nilai *setpoint* dan LCD sebagai *display*.



Gambar 3.4 Sistem Keseluruhan

Pada Gambar 3.4, dapat diketahui bahwa ada dua tabung filtrasi dan 2 akuarium penampung air. Tabung *filter* 1 berisi bahan *filter* yaitu silica, pasir dan zeolite. Tabung *filter* 2 berisi kain kasa. Kedua tabung *filter* ini digunakan untuk proses filtrasi pada air. Akuarium 1 sebagai tempat penampungan untuk air yg akan difiltrasi, kemudian akuarium 2 digunakan sebagai tempat penampungan air yang akan difiltrasi. Sensor kejernihan ditempatkan pada kedua akuarium ini untuk mendeteksi kejernihan air. Pada akuarium 2 kejernihan air dikontrol lebih besar atau sama dengan *setpoint* nilai tingkat kejernihan air yang diinginkan. Pada penelitian ini kedua sensor mendeteksi kejernihan air sebelum difiltrasi dan air setelah difiltrasi kemudian diproses oleh arduino uno untuk mengendalikan kecepatan aliran air yang masuk kedalam alat filtrasi.

Dasar dari penelitian ini yaitu mengenai pengaruh kecepatan aliran air yang masuk kedalam alat filtrasi. Pada teorinya semakin lambat aliran air filtrasi maka semakin maksimal kejernihan hasil air filtrasi, dan begitu pula sebaliknya. Metode *control* yang digunakan dalam penelitian ini yaitu *control fuzzy_PID*.

Plant terdiri dari beberapa komponen utama diantaranya sebagai berikut:

1. Sensor kejernihan air merupakan sensor yang mendeteksi perubahan nilai kejernihan pada air sebelum masuk ke alat filtrasi dan air setelah melalui proses filtrasi. Keluaran sensor ini berupa tegangan *analog*. Terdapat dua sensor kejernihan yaitu pada akuarium pertama dan pada akuarium kedua.
2. Pompa air AC yang digunakan untuk mengalirkan air kedalam alat filtrasi dengan sumber tegangan 220 V AC.
3. Sebuah *driver digital pwm dimmer* yang berfungsi sebagai pengendali tegangan pada pompa air.
4. *Push Button* berfungsi sebagai masukan untuk mengatur *set point* kejernihan air hasil filtrasi dan sebagai tombol untuk menjalankan proses pengendalian.
5. LCD (*Liquid Crystal Display*) dan *driver LCD* berfungsi sebagai media tampilan selama proses pengendalian berlangsung.
6. Kran berfungsi untuk memutar aliran air pada proses pencucian media *filter* pada tabung *filter 1* dan tabung *filter 2*.
7. Catu daya berfungsi sebagai *power supply* sistem secara keseluruhan.
8. Arduino uno berfungsi sebagai pusat pengendalian yang pemograman menggunakan Arduino IDE dengan metode *control fuzzy-PID*.
9. *Personal computer* sebagai unit *monitoring* yang menampilkan data kejernihan aktual air. Kemudian data dirubah dalam bentuk grafik untuk keperluan analisis.

Pada media penampungan air, realisasinya akuarium 1 dibuat memiliki *volume* yang lebih besar. Sedangkan dan akuarium 2 dibuat dengan *volume* yang lebih kecil. Hal ini dimaksudkan supaya akuarium 1 dapat menampung air lebih banyak karena digunakan untuk proses proses penjernihan sedangkan akuarium 2 digunakan untuk menampung hasil penjernihan agar pendeteksian sensor lebih maksimal dengan perubahan sedikit kejernihan pada air hasil filtrasi.

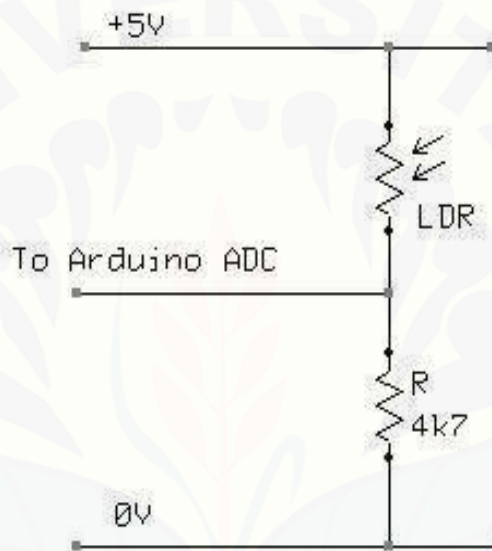
3.7 Rangkaian Sensor

Sensor yang digunakan pada sistem ini yaitu sensor kejernihan air masuk (sensor kejernihan 1), sensor kejernihan air hasil filtrasi (sensor kejernihan 2) dan

sensor *flowmeter*. Hasil perancangan dan pengujian sensor-sensor tersebut dapat dijelaskan sebagai berikut:

3.7.1 Rangkaian Sensor Kejernihan Air

Sensor kejernihan yang digunakan dalam sistem ini adalah LDR dan lampu led. Led sebagai *transmitter* dan LDR sebagai *receiver*, dibutuhkan untuk mendeteksi tingkat kejernihan air. Tingkat kejernihan air inilah yang akan menjadi *feedback* ke arduino untuk untuk mengontrol kecepatan aliran air yang masuk kedalam media filtrasi.



Gambar 3.5 Rangkaian Sensor Kejernihan Air

Terdapat dua sensor yaitu pada akuarium 1 dan akuarium 2. Pada akuarium 1 berfungsi untuk mendeteksi kejernihan air yang masuk menuju alat filtrasi. Sensor kejernihan 2 digunakan untuk mendeteksi kejernihan aktual air air filtrasi.

Pada sensor kejernihan, LDR dan lampu led tidak tahan air (*water proof*) maka diperlukan adanya pelindung untuk menjaga agar sensor tidak terkena air sehingga sensor tidak rusak. Sensor dimasukan ke dalam pipa PVC ukuran $\frac{1}{2}$ yang telah dilubangi dari sisi samping dan penutup pada sisi atas dan bawah, kemudian pada rangkaian diberi lem bakar sehingga rangkaian dan kabel tidak terkena air. Hal ini sangat berguna dalam menjaga sensor dapat mendeteksi perubahan kejernihan dan menghindarkan dari kerusakan.

Untuk konversi nilai ADC menjadi tegangan menggunakan persamaan sebagai berikut:

$$V_{out} = V_{in} * (5.0 / 1023.0) \dots\dots\dots (3.2)$$

Untuk konversi nilai tegangan menjadi nilai kejernihan menggunakan persamaan sebagai berikut:

$$x = (V_{out} / 5) * 100 \dots\dots\dots (3.3)$$

Kemudian untuk mendapatkan kejernihan dalam satuan NTU digunakan persamaan sebagai berikut:

$$y = -1.6047x + 153.01 \dots\dots\dots (3.4)$$

keterangan:

V_{out} = tegangan *output* analog sensor

V_{in} = tegangan *input* analog sensor

y = nilai NTU

x = nilai kejernihan dalam bentuk *percent* (%)

Kalibrasi rangkaian sensor kejernihan 1 dan sensor kejernihan 2 diperlukan agar kedua sensor memiliki keluaran yang sama. Cara kerja kalibrasi kedua sensor tersebut dilakukan dengan mengukur tingkat kejernihan masing masing sensor pada Tingkat kejernihan air yang sama, kemudian perbandingan pembacaan kedua sensor ditambahkan pada fungsi *transfer* pembacaan sensor sehingga dihasilkan *output* yang sama. persamaan yang digunakan adalah sebagai berikut :

$$a = b + (100 - c) \dots\dots\dots (3.5)$$

keterangan:

a = nilai kejernihan

b = hasil pembacaan sensor

c = hasil pembacaan sensor

Range pembacaan sensor dibuat pada nilai 0-100. Nilai 0 merupakan tingkat kejernihan air terkecil sedangkan 100 merupakan tingkat kejernihan air terbesar. Berikut gambar hasil dari pembacaan sensor.



Gambar 3.6 Pembacaan Sensor Awal

Pada gambar 3.6 diketahui bahwa sensor membaca nilai kejernihan yaitu Tin (sensor kejernihan 1) = 89 dan $Tout$ (sensor kejernihan 2) = 87, kemudian dengan mengacu pada persamaan 3.2 didapatkan selisih nilai Tin yaitu $100 - 89 = 11$ dan $Tout$ yaitu $100 - 87 = 13$. Sehingga untuk mendapatkan nilai kejernihan digunakan persamaan 3.2 yaitu ($Tin = 89 + 11$) dan ($Tout = 87 + 13$). Berikut gambar pembacaan sensor setelah dikalibrasi.



Gambar 3.7 Pembacaan Sensor Setelah dikalibrasi

3.7.2 Sensor *Flowmeter*

Sensor *flowmeter* dalam penelitian ini digunakan untuk *monitoring* kecepatan aliran air pada saat sistem bekerja. Aliran air yang diukur ialah air yang masuk pada tabung filtrasi. Pengukuran kecepatan aliran air ini digunakan untuk bahan analisis pengaruh kecepatan aliran air terhadap kejernihan hasil proses filtrasi. Sensor *flowmeter* yang digunakan ialah ukuran $\frac{3}{4}$ inch yang terdiri dari katup plastik, rotor air, dan sebuah sensor *hall-effect*.

Pada pengaplikasiannya sensor *flowmeter* ini diberikan tegangan 5 volt dari arduino uno. Pin *interrupt* yang digunakan ialah pin 2 pada arduino uno. Untuk perhitungan frekuensi digunakan persamaan sebagai berikut:

$$X = 5,5 * Y * t \dots\dots\dots (3.6)$$

keterangan:

- X = Kecepatan aliran
- Y = unit aliran (L/min)
- t = waktu (detik).

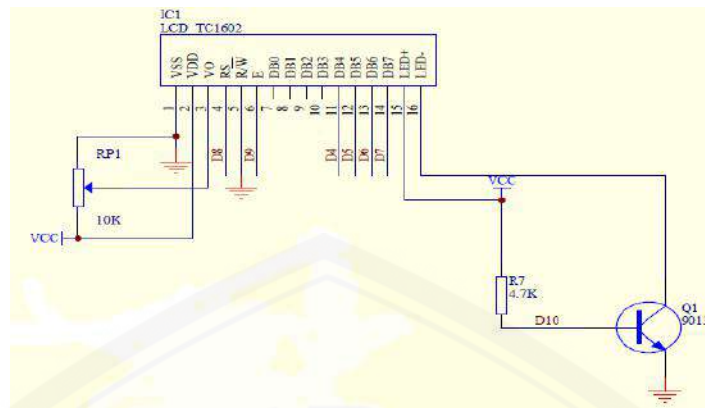
Berikut gambar sensor *flowmeter* pada sistem.



Gambar 3.8 Sensor *flowmeter* (seedstudio.com)

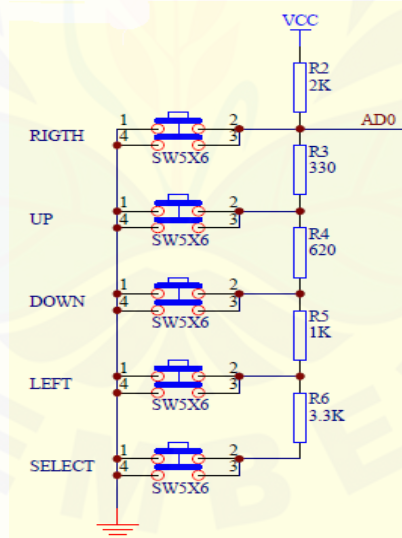
3.8 Display dan Push Button

Pada tugas akhir ini *display* untuk menampilkan kecepatan aliran air yang masuk kedalam *plant*, nilai K_p , K_i dan K_d , *setpoint* dan beberapa parameter serta hasil pengaturan yang dilakukan dipakai sebuah LCD *shield* 2x16. Konfigurasi *pin* dari LCD dan alokasinya pada *port* mikrokontroler dapat dilihat pada Gambar 3.9.



Gambar 3.9 Antarmuka LCD 4 Bit (sainsmart.com)

Mode transfer data yang digunakan adalah 4 bit. Pada gambar 3.10 merupakan antarmuka *push button* yang menggunakan konsep pembagi tegangan. Selama ini tombol atau *push button* disambung ke *pin* mikrokontroler langsung, namun yang ini berbeda yaitu menggunakan rangkaian pembagi tegangan yang dihubungkan ke *port input analog* A0 (masukan *analog* pertama) dari *board* Arduino uno.

Gambar 3.10 Konfigurasi *pin push button* (sainsmart.com)

Cara kerja tombol menggunakan rangkaian 5 tingkat pembagi tegangan intinya adalah menggunakan resistor (perhatikan Gambar 3.10). Jika tidak ada satupun tombol yang ditekan maka AD0 tersambung ke Vcc melalui R2 (22K). Jika tombol *SELECT* ditekan maka terjadi penjumlahan resistor secara seri ($R2 + R3 + R4 + R5 + R6$) kemudian menuju GND. Demikian juga dengan masing-masing

tombol yang ditekan menghasilkan nilai tegangan yang berbeda-beda, hal inilah yang digunakan untuk mendeteksi tombol mana yang ditekan.

Pengaplikasian dari *keypad* pada penelitian ini yaitu digunakan untuk memasukkan nilai *setpoint* kejernihan air pada hasil filtrasi yang diinginkan. Berikut gambar tampilan LCD dan hasil pembacaan nilai *input keypad* pada sistem.



Gambar 3.11 Pembacaan *keypad right*

Pada gambar 3.11 pembacaan nilai ADC *keypad right* sebesar 0 Bit



Gambar 3.12 Pembacaan *keypad left*

Pada gambar 3.12 pembacaan nilai ADC *keypad left* sebesar 410 Bit



Gambar 3.13 Pembacaan *keypad up*

Pada gambar 3.13 pembacaan nilai ADC *keypad up* sebesar 100 Bit



Gambar 3.14 Pembacaan *keypad down*

Pada gambar 3.14 pembacaan nilai ADC *keypad down* sebesar 256 Bit.



Gambar 3.15 Pembacaan *keypad select*

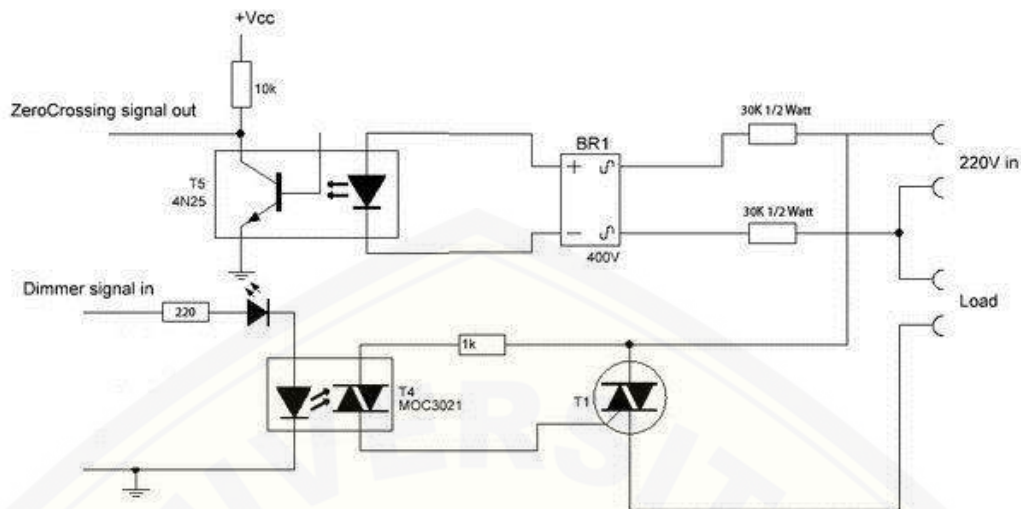
Pada gambar 3.15 diketahui bahwa pembacaan nilai ADC *keypad select* sebesar 256 Bit.

Dari hasil pembacaan beberapa *keypad* tersebut kemudian nilai masing-masing *keypad* digunakan dalam pemrograman untuk mengaktifkan fungsi dari masing masing *keypad*.

3.9 Rangkaian *Driver PWM Digital Dimmer Untuk Pompa Air*

Secara teori suatu *analog output* akan mengeluarkan *output* tegangan bervariasi sesuai dengan nilai yang dikehendaki, maka seharusnya *pin output analog* Arduino mampu mengeluarkan tegangan *output* dengan kisaran tegangan dari 0 V sampai 5V. Akan tetapi tidak demikian adanya, karena *pin-pin* Arduino yang difungsikan sebagai *output* sebenarnya hanya mampu sebagai *digital output* yaitu hanya mampu mengeluarkan tegangan 0V atau 5V. Namun Arduino juga menangani *Analog Output* menggunakan *Pulsa Wide Modulasi (PWM)* atau modulasi lebar *pulsa* untuk menghasilkan *analog output* yang dikehendaki. *Pin* yang difungsikan sebagai *PWM analog output* akan mengeluarkan sinyal *pulsa digital* dengan frekuensi 490 Hz. Nilai tegangan *analog* diperoleh dengan merubah *Duty Cycle* atau perbandingan lamanya *pulsa HIGH* terhadap periode (T) dari sinyal *digital* tersebut. Jika *pulsa HIGH* muncul selama setengah dari periode sinyal maka akan menghasilkan *duty cycle 50%* yang berarti sinyal *analog* yang dihasilkan sebesar setengah dari tegangan *analog* maksimal yaitu $1/2$ dari 5 V atau sama dengan 2,5 V. Begitu juga halnya jika *pulsa HIGH* hanya seperempat bagian dari periode sinyal maka tegangan *analog* identik yang dihasilkan adalah $1/4$ dari 5V = 1,25 V dan seterusnya. *Pin PWM* yang digunakan dalam penelitian ini yaitu pin 3.

Pengendalian besarnya arus yang melewati motor pompa air yang di catu daya AC digunakan *dimmer circuit*, yang di dalamnya terdapat rangkaian untuk pemicuan *gate TRIAC* dan rangkaian *Zero Cross Detector*. Berikut gambar rangkaian *PWM Digital Dimmer*.



Gambar 3.16 Rangkaian *driver pwm digital dimmer*

TRIAC atau *Triode for Alternating Current* (Trioda untuk arus bolak-balik) adalah komponen yang tersusun atas dua buah *thyristor* dengan terminal *gate* yang sama. Nama resmi untuk TRIAC adalah *Bidirectional Triode Thyristor*. Ini menunjukkan sakelar dua arah yang dapat mengalirkan arus listrik ke kedua arah ketika dipicu. Dalam penggunaannya dibutuhkan DIAC sebagai *driver* dari TRIAC. Selain itu dibutuhkan juga komponen yang mampu berfungsi sebagai isolator antara tegangan 220V dan 5V untuk melindungi mikrokontroler dari tegangan balik, sehingga komponen yang sesuai dengan spesifikasi tersebut adalah *optocoupler*. *Optocoupler* yang komponen utamanya berupa DIAC adalah MOC3041.

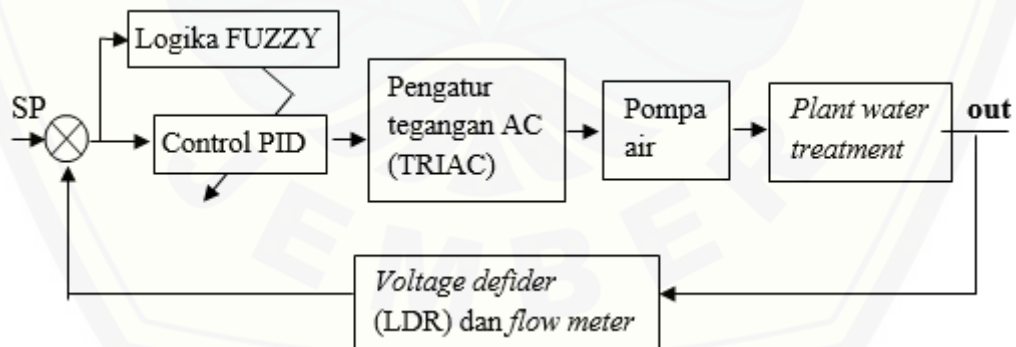
Zero cross detector adalah rangkaian yang digunakan untuk mendeteksi gelombang sinus AC 220 volt saat melewati titik tegangan nol. Dengan rangkaian *zero crossing detector*, kita bisa mendeteksi *zero point* sekaligus mengubah suatu sinyal sinusoida menjadi sinyal *digital*. Sinyal keluaran rangkaian *zero cross detector* ini masuk ke dalam mikrokontroler, oleh karena itu dibutuhkan juga komponen yang mampu memisahkan tegangan 5V dan 220V. Untuk rangkaian ini menggunakan komponen *optocoupler* 4N25.



Gambar 3.17 Driver pwm digital dimmer

3.10 Pengendali Fuzzy-PID

Pada penelitian tugas akhir ini kendali yang digunakan adalah *fuzzy*-PID. *fuzzy logic* digunakan untuk menghasilkan parameter pada kendali PID. Dari kendali PID inilah yang nantinya menghasilkan sinyal kontrol yang menggerakkan aktuator berupa motor pompa air. Masukan dari pengendali *fuzzy logic* adalah *error* dan perubahan *error* sedangkan masukan dari PID adalah *error*. Kendali *fuzzy logic* yang akan menghasilkan parameter K_p , K_i , dan K_d untuk kendali PID.



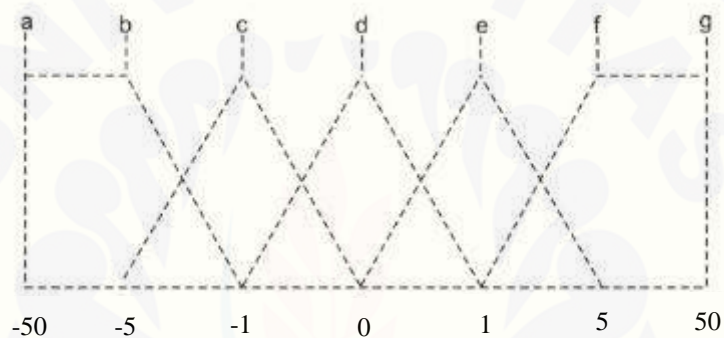
Gambar 3.18 Diagram blok pengendali fuzzy_PID

Kendali *fuzzy* dirancang dengan dua masukan yaitu *error* dan perubahan *error*, serta satu keluaran. Masukan berupa *error* dan perubahan *error* diolah melalui serangkaian proses. Pertama dimulai dari fuzzifikasi hingga defuzzifikasi dan menghasilkan sinyal kendali untuk mengendalikan *plant* yaitu kendali kejernihan air hasil filtrasi. Serangkaian proses kendali *fuzzy* tersebut membutuhkan

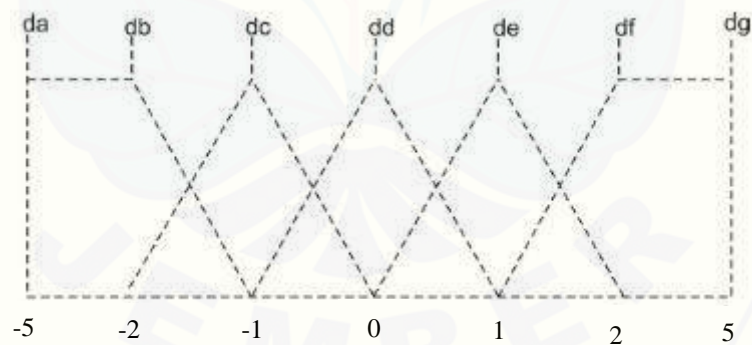
dua parameter utama yaitu fungsi keanggotaan (*membership function*) dan aturan dasar (*rule base*).

A. Fungsi Keanggotaan

Pengendali *fuzzy* yang dirancang memiliki dua buah masukan yaitu *error* dan perubahan *error*. Himpunan *fuzzy* untuk masukan *error* dan perubahan *error*, masing-masing dibagi menjadi 5 kelas. Himpunan *fuzzy* untuk keluarannya dibagi dalam 5 tingkat sinyal *control* yang berbentuk *discrete*, nama linguistik kelima tingkatan ini adalah parameter. Gambar 3.19 dan 3.20 menunjukkan bentuk dan batasan masukan *error* dan perubahan *error*.



Gambar 3.19 Keanggotaan *error*



Gambar 3.20 Keanggotaan perubahan *error*

Masukan *error* mempunyai data tegas antara -50 sampai 50 dan perubahan *error* -5 sampai 5, masukan *error* dipetakan dalam nilai linguistik menjadi 5 kelas yaitu *Negatif Besar* (NB), *Negatif* (N), *Zero* (Z), *Positif* (P), *Positif Besar* (PB), dan untuk masukan perubahan *error* dipetakan dalam nilai linguistik menjadi 5 kelas *Negatif Besar* (NB), *Negatif* (N), *Zero* (Z), *Positif* (P), *Positif Besar* (PB). Derajat keanggotaan bernilai 0 sampai 1.

B. Rule Base (Basis Pengetahuan)

Pada tugas akhir ini *rule base* dan pengambilan keputusan terdiri dari dua masukan dan tiga keluaran. Eksekusi aturan diproses menggunakan implikasi yang akan mengambil nilai paling minimal dari kedua masukan yaitu derajat keanggotaan *error* dan perubahan *error* dan juga menentukan konstanta parameter keluaran yang digunakan. *Input error*, perubahan *error* dan keluaran masing-masing mempunyai 5 *membership functions*.

Tabel 3.1 Aturan Fuzzy Keluaran K_p

<i>Error</i> $\Delta error$	NB	N	Z	P	PB
NB	$K^{(w_1)}$	$K^{(w_2)}$	$K^{(w_3)}$	$K^{(w_4)}$	$K^{(w_5)}$
N	$K^{(w_6)}$	$K^{(w_7)}$	$K^{(w_8)}$	$S^{(w_9)}$	$B^{(w_{10})}$
Z	$K^{(w_{11})}$	$K^{(w_{12})}$	$K^{(w_{13})}$	$B^{(w_{14})}$	$SB^{(w_{15})}$
P	$K^{(w_{16})}$	$K^{(w_{17})}$	$K^{(w_{18})}$	$B^{(w_{19})}$	$SB^{(w_{20})}$
PB	$K^{(w_{21})}$	$K^{(w_{22})}$	$K^{(w_{23})}$	$SB^{(w_{24})}$	$BS^{(w_{25})}$

Tabel 3.2 Aturan Fuzzy Keluaran T_i

<i>Error</i> $\Delta error$	NB	N	Z	P	PB
NB	$K^{(w_1)}$	$K^{(w_2)}$	$K^{(w_3)}$	$S^{(w_4)}$	$S^{(w_5)}$
N	$K^{(w_6)}$	$K^{(w_7)}$	$K^{(w_8)}$	$S^{(w_9)}$	$S^{(w_{10})}$
Z	$K^{(w_{11})}$	$K^{(w_{12})}$	$K^{(w_{13})}$	$S^{(w_{14})}$	$S^{(w_{15})}$
P	$K^{(w_{16})}$	$K^{(w_{17})}$	$K^{(w_{18})}$	$S^{(w_{19})}$	$B^{(w_{20})}$
PB	$K^{(w_{21})}$	$K^{(w_{22})}$	$K^{(w_{23})}$	$B^{(w_{24})}$	$B^{(w_{25})}$

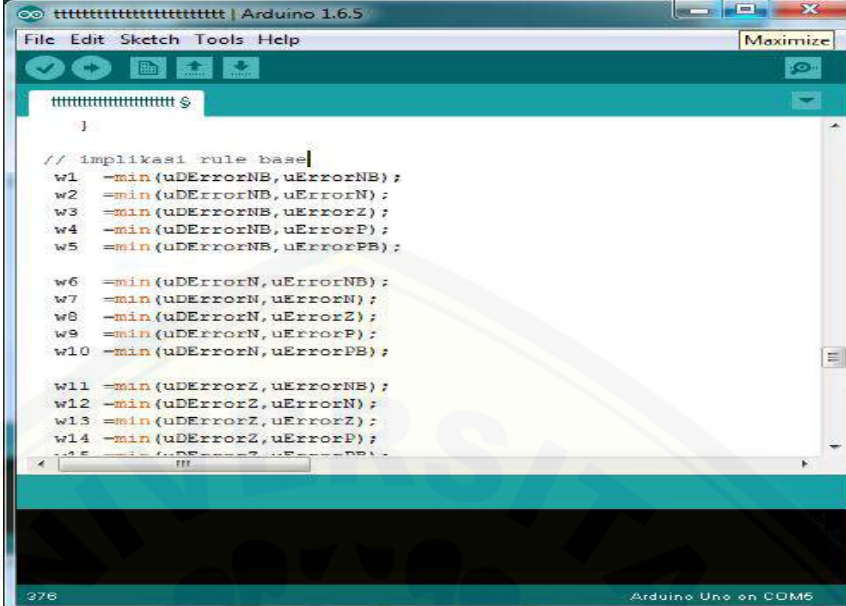
Tabel 3.3 Aturan fuzzy keluaran T_d

<i>Error</i> $\Delta error$	NB	N	Z	P	PB
NB	$BS^{(w_1)}$	$BS^{(w_2)}$	$S^{(w_3)}$	$K^{(w_4)}$	$K^{(w_5)}$
N	$BS^{(w_6)}$	$BS^{(w_7)}$	$S^{(w_8)}$	$K^{(w_9)}$	$K^{(w_{10})}$
Z	$BS^{(w_{11})}$	$SB^{(w_{12})}$	$K^{(w_{13})}$	$K^{(w_{14})}$	$K^{(w_{15})}$
P	$BS^{(w_{16})}$	$SB^{(w_{17})}$	$K^{(w_{18})}$	$K^{(w_{19})}$	$K^{(w_{20})}$
PB	$BS^{(w_{21})}$	$SB^{(w_{22})}$	$S^{(w_{23})}$	$K^{(w_{24})}$	$K^{(w_{25})}$

Dari ketiga tabel aturan *fuzzy*, dapat dibuat aturan *fuzzy* sebagai berikut:

1. *If error is NB AND $\Delta Error$ is N then $K_p = K$, $K_i = K$, dan $K_d = BS$*
2. *if Error is NB AND $\Delta Error$ is Z then $K_p = K$, $K_i = K$, dan $K_d = BS$*
3. *If error is NB AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = S$*
4. *If error is NB AND $\Delta Error$ is N then $K_p = B$, $K_i = S$, dan $K_d = K$*
5. *If error is NB AND $\Delta Error$ is Z then $K_p = SB$, $K_i = B$, dan $K_d = K$*
6. *If error is N AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = BS$*
7. *If error is N AND $\Delta Error$ is N then $K_p = K$, $K_i = K$, dan $K_d = BS$*
8. *If error is N AND $\Delta Error$ is Z then $K_p = K$, $K_i = K$, dan $K_d = S$*
9. *If error is N AND $\Delta Error$ is P then $K_p = S$, $K_i = S$, dan $K_d = K$*
10. *If error is N AND $\Delta Error$ is N then $K_p = B$, $K_i = S$, dan $K_d = K$*
11. *If error is Z AND $\Delta Error$ is Z then $K_p = K$, $K_i = K$, dan $K_d = BS$*
12. *If error is Z AND $\Delta Error$ is P then $K_p = B$, $K_i = K$, dan $K_d = SB$*
13. *If error is Z AND $\Delta Error$ is N then $K_p = K$, $K_i = K$, dan $K_d = K$*
14. *If error is Z AND $\Delta Error$ is Z then $K_p = B$, $K_i = S$, dan $K_d = K$*
15. *If error is Z AND $\Delta Error$ is P then $K_p = SB$, $K_i = S$, dan $K_d = K$*
16. *If error is P AND $\Delta Error$ is N then $K_p = K$, $K_i = K$, dan $K_d = BS$*
17. *If error is P AND $\Delta Error$ is Z then $K_p = K$, $K_i = K$, dan $K_d = SB$*
18. *If error is P AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = K$*
19. *If error is P AND $\Delta Error$ is N then $K_p = B$, $K_i = S$, dan $K_d = K$*
20. *If error is P AND $\Delta Error$ is Z then $K_p = SB$, $K_i = B$, dan $K_d = K$*
21. *If error is PB AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = BS$*
22. *If error is PB AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = SB$*
23. *If error is PB AND $\Delta Error$ is P then $K_p = K$, $K_i = K$, dan $K_d = S$*
24. *If error is PB AND $\Delta Error$ is P then $K_p = SB$, $K_i = B$, dan $K_d = K$*
25. *If error is PB AND $\Delta Error$ is P then $K_p = BS$, $K_i = B$, dan $K_d = K$*

Dari setiap keadaan hubungan antara *Error* dan $\Delta Error$ diberikan *indeks* berupa w_1 sampai dengan w_{25} sehingga memudahkan dalam pemrograman pada arduino IDE. Berikut gambar pemograman aturan *fuzzy*.



```

// implikasi rule base
w1 =min(uErrorNB, uErrorNB);
w2 =min(uErrorNB, uErrorN);
w3 =min(uErrorNB, uErrorZ);
w4 =min(uErrorNB, uErrorP);
w5 =min(uErrorNB, uErrorPB);

w6 =min(uErrorN, uErrorNB);
w7 =min(uErrorN, uErrorN);
w8 =min(uErrorN, uErrorZ);
w9 =min(uErrorN, uErrorP);
w10 =min(uErrorN, uErrorPB);

w11 =min(uErrorZ, uErrorNB);
w12 =min(uErrorZ, uErrorN);
w13 =min(uErrorZ, uErrorZ);
w14 =min(uErrorZ, uErrorP);
w15 =min(uErrorZ, uErrorPB);

```

Gambar 3.21 Pemrograman *rule base* pada arduino IDE

C. Mekanisme Defuzifikasi

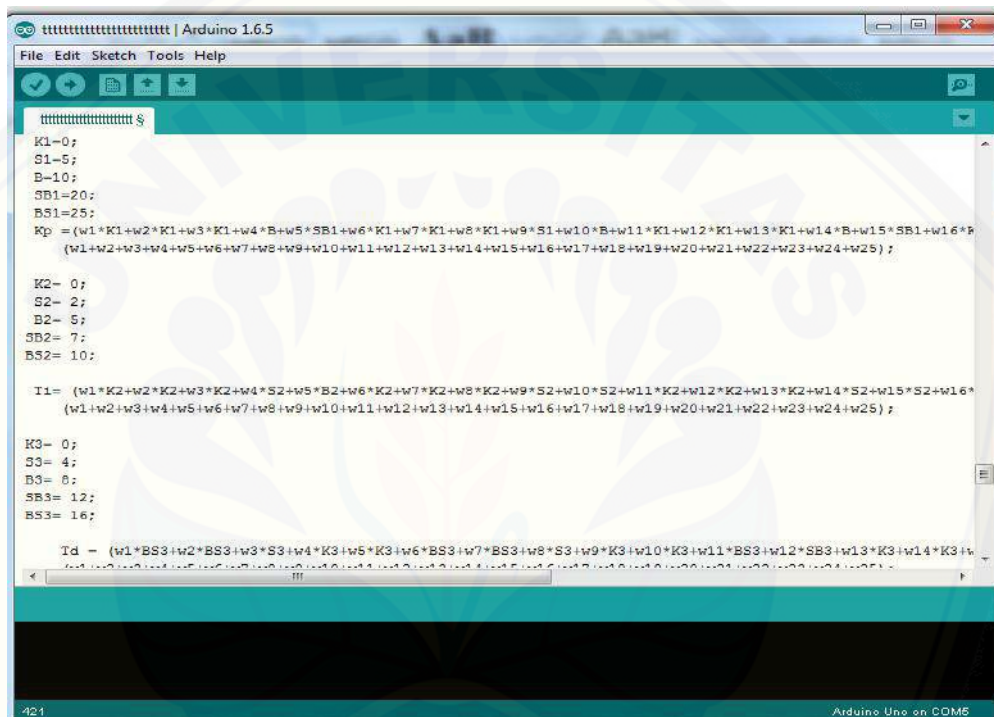
Gambar 3.22 Desain *output fuzzy*

Mekanisme defuzifikasi pada tugas akhir ini menggunakan *output* berupa konstanta. Keluaran dari logika *fuzzy* ini dibagi dalam 5 tingkat yang berbentuk *discrete*, nama linguistik kelima tingkatan ini adalah parameter. Kelima parameter keluaran tersebut K (kecil), S (sedang), B (besar), SB (sedang besar) dan BS (Besarnya sekali). Untuk nilai K_p , K bernilai 0, S bernilai 2, B bernilai 4, SB bernilai 10, dan BS bernilai 15. Pada penalaan parameter T_i , K bernilai 0, S bernilai 5, B bernilai 10, SB bernilai 15 dan BS bernilai 20. Untuk parameter T_d dirancang dengan K bernilai 0, S bernilai 1, B bernilai 3, SB bernilai 5 dan BS bernilai 7. Untuk mendapatkan nilai *crisp output* dari himpunan *fuzzy* ini dapat digunakan metode

rata-rata berbobot (*Center average defuzzifier*). Pada metode ini nilai *crisp* keluarannya diperoleh berdasarkan titik berat dari kurva hasil proses pengambilan keputusan yang dapat dirumuskan sebagai berikut:

$$Output = \frac{\sum_{i=1}^N wiKpi}{\sum_{i=1}^N wi} \dots\dots\dots (3.7)$$

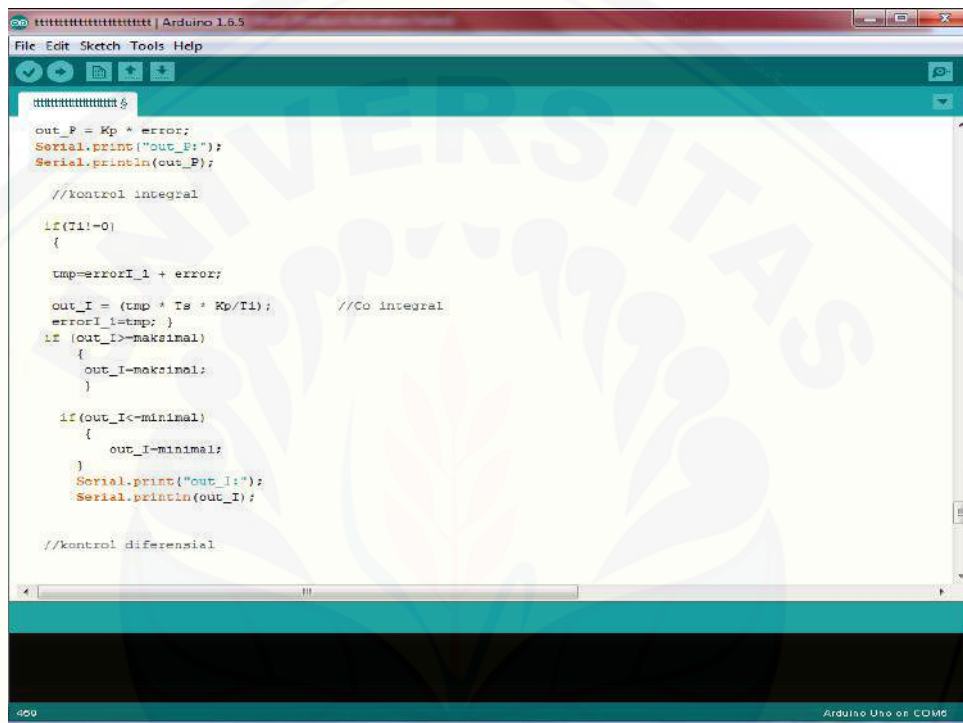
Berikut gambar pemograman aturan *fuzzy* pada arduino IDE.



Gambar 3.23 Gambar pemograman aturan *fuzzy*

Pada tugas akhir ini digunakan metode kontrol PID untuk mengendalikan kecepatan motor pompa. Sesuai dengan persamaan (3.7) maka dapat direalisasikan kendali PID dalam bahasa C. Kendali PID akan menghasilkan sinyal kontrol antara 0 sampai 255 yang akan mengatur kecepatan motor pompa air. Nilai parameter K_p , K_i , dan K_d didapatkan dari kendali *fuzzy logic*. *Error* digunakan sebagai masukan pada kontrol PID. Kendali PID dimulai dengan inialisasi parameter PID. Parameter ini didapatkan dari kendali *fuzzy logic*. Kemudian inialisasi *timer* yaitu nilai parameter dari T_s (*time sampling*). Setelah itu, kontrol PID menghitung dan menghasilkan nilai sinyal kontrol tertentu. Untuk mendapatkan aksi kontrol PID yang baik pada *plant* maka digunakan anti *wind up* pada algoritma PID. Anti *wind*

up digunakan untuk membatasi nilai sinyal kontrol. Jika sinyal kontrol melebihi batas maksimal yang telah ditentukan maka sinyal kontrol bernilai maksimal. Jika sinyal kontrol kurang dari batas minimal maka sinyal kontrol bernilai minimal sesuai dengan yang telah ditentukan (dapat dilihat pada lampiran *listing* program). Berikut gambar bagian kontrol PID pada pemrograman *fuzzy_PID* dengan *software* arduino IDE.



```
out_P = Kp * error;
Serial.print("out_P:");
Serial.println(out_P);

//kontrol integral
if(Ti!=0)
{
  tmp=errorI_i + error;
  out_I = (tmp * Ts + Kp/Ti); //Co integral
  errorI_i=tmp;
  if (out_I>maksimal)
  {
    out_I=maksimal;
  }

  if(out_I<minimal)
  {
    out_I=minimal;
  }
  Serial.print("out_I:");
  Serial.println(out_I);

//kontrol diferensial
```

Gambar 3.24 Pemograman kontrol *fuzzy_PID*

BAB 5. PENUTUP

1.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan dari penelitian dengan judul “Sistem Pengaturan Laju Aliran Air Pada *Plant Water Treatment* Skala Rumah Tangga dengan Kontrol *Fuzzy-PID*” maka dapat dihasilkan beberapa kesimpulan sebagai berikut:

1. Nilai pembacaan sensor kejernihan turun sebesar 2 % terhadap penambahan zat terlarut Tinta 2.5 ml pada air dengan rata rata *error percent* pembacaan sebesar 5.23 %. (Tabel 4.1 halaman 48)
2. Rata – rata nilai *error percent* pembacaan sensor *flowmeter* sebesar 3.94 %. (Tabel 4.2 halaman 50)
3. Kecepatan debit aliran air berubah dari 872 (Liter/jam) menjadi 752 (Liter/jam) pada saat terjadi perubahan kejernihan air *output* filtrasi dari 94 % menjadi 93%. (Tabel 4.7 halaman 58)
4. Kontrol logika *fuzzy_PID* pada *plant* ini memiliki kelemahan, yaitu ketika sistem telah mencapai kejernihan pada nilai *setpoint*, kecepatan aliran air menjadi tidak stabil. Keadaan tersebut disebabkan, karena pada saat (kejernihan hasil filtrasi = *setpoint*), motor kembali ke kecepatan maksimal, sehingga membuat kejernihan air *output* kembali menurun.

1.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, maka penulis memberikan saran guna pengembangan penelitian ini berikutnya.

1. Untuk mendapatkan hasil filtrasi yang lebih maksimal, diperlukan tabung atau tempat media filtrasi yang lebih besar dan untuk penambahan debit kecepatan aliran air pada proses filtrasi diharapkan menggunakan media filtrasi dengan kerapatan pori-pori yang tinggi.
2. Dapat dikembangkan sistem *backwash* secara otomatis.

DAFTAR PUSTAKA

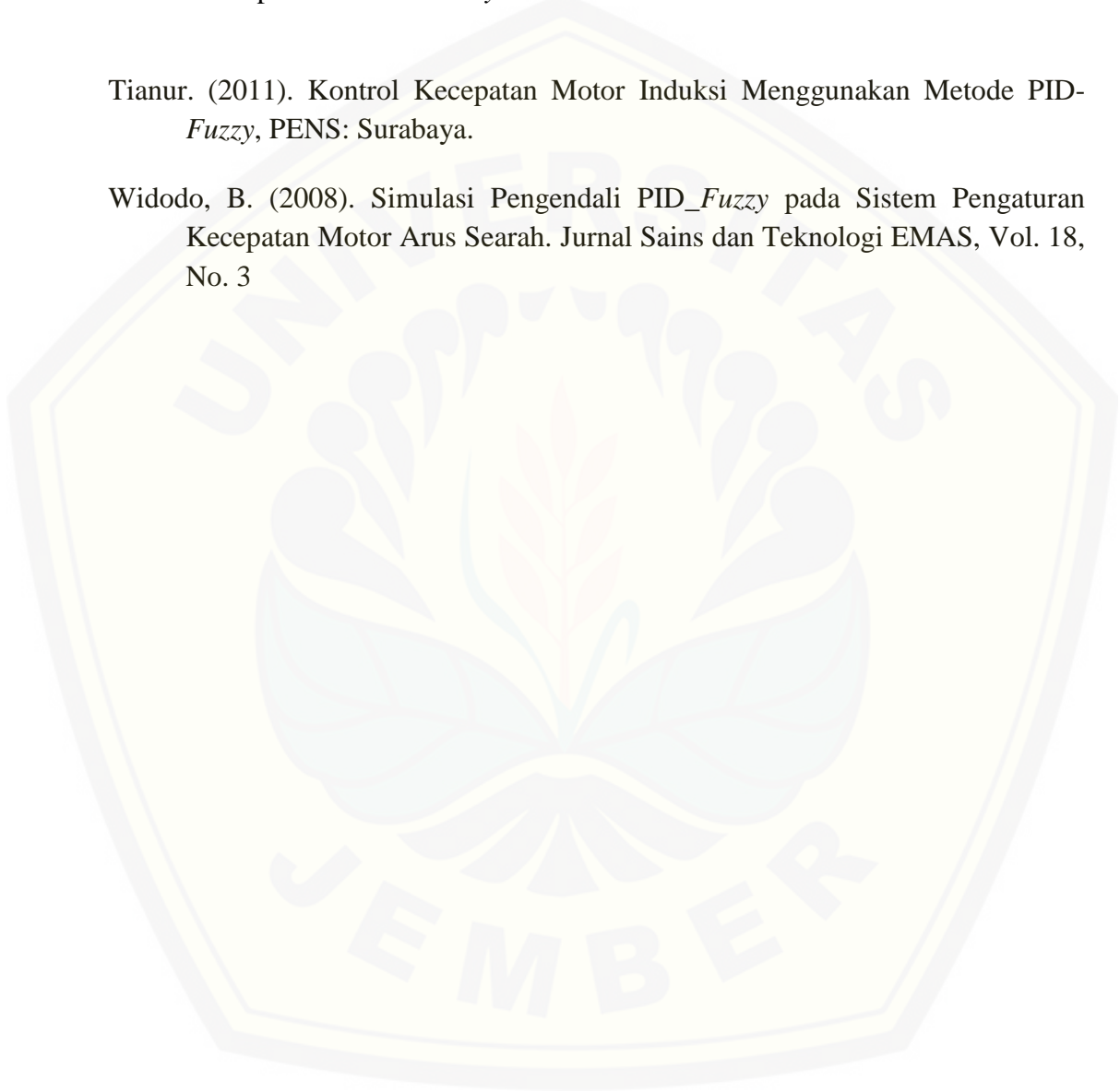
- Cordova, H. 2004. *PID Self-Tuning Based On Auto Swotch Algorithm To Control pH (Neutralization) Process*. Teknik Fisika, ITS: Surabaya.
- Djuandi, F. 2011. *Pengenalan Arduino*. Tenik Elektro Universitas Trisakti: Jakarta.
- Gerald Recktenwald. 2011. *Basic Pulse Width Modulation*. EAS 199.
- Rindho S Prestian. 2010. *Tuning Parameter Proporsional – Integral dengan Fuzzy Logic untuk Pengaturan Suhu Air pada Plant Heat Exchanger*. Teknik Elektro, Fakultas Teknik Universitas Diponegoro: Semarang.
- Kadir, Abdul. 1995. “Pemrograman C++ Edisi II”. Andi: Yogyakarta.
- Muchlas, Tole Sutikno, dkk. 2006. *Kendali Kecepatan Motor Induksi Satu Fasa Dengan Inverter Pwm Pulsa Tunggal Berbasis Mikrokontroler At89s51*. Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan: Yogyakarta.
- Ika Nuzula Nike, Endarko. 2013. *Perancangan dan Pembuatan Alat Ukur Kejernihan Air Berbasis Mikrokontroler ATMega 8535*. Jurusan Fisika, Fakultas MIPA, ITS: Surabaya.
- Ferdiansyah, D.S. (2010). *Pengaturan Kecepatan MotorInduksi 3 Phase dengan Kontrol PID melalui Metode Field Oriented Control*. PENS: Surabaya.
- Wisesa Tirta. 2014. *Perancangan Pengaturan Kecepatan Motor Induksi Satu Fasa Dengan PWM Menggunakan Pengendali PID Berbasis Arduino*. Program Studi Teknik Elektro, Universitas Bengkulu: Bengkulu
- Gunterus, F. (1994). *Falsafah Dasar Sistem Pengendalian Proses*. Jakarta: PT Elex Media Komputindo.
- Johnson, C. (1988). *Process Control Instrumentation Technology*. New Jersey: Englewood Cliffs.
- Nirali, R., & Shah, S. (2011). *Fuzzy Decision Based Soft Multi Agent Controller for Speed Control of Three Phase Induction Motor*. International Journal on Soft Computing (IJSC) Vol.2 No.3.

Heryanto, M. Ary dan Wisnu Adi P, Pemrograman Bahasa C Untuk Mikrokontroler ATMEGA8535. Penerbit Andi, Yogyakarta, 2008.

Sugeng. (2012). Penerapan *Fuzzy Inference System* Untuk Mengendalikan Kecepatan Motor *Conveyor*. STMIK Eresha: Jakarta.

Tianur. (2011). Kontrol Kecepatan Motor Induksi Menggunakan Metode PID-*Fuzzy*, PENS: Surabaya.

Widodo, B. (2008). Simulasi Pengendali PID-*Fuzzy* pada Sistem Pengaturan Kecepatan Motor Arus Searah. Jurnal Sains dan Teknologi EMAS, Vol. 18, No. 3



A. LAMPIRAN HARDWARE DAN SOFTWARE



Gambar nilai ADC keypad up



gambar nilai ADC keypad right



Gambar nilai ADC keypad Left



Gambar nilai ADC keypad Select



Gambar Driver PWM digital dimmer



Gambar Tabung Filtrasi



Gambar Sensor kekeruhan



Gambar aquarium hasil filtrasi



Gambar kontrol alat



Gambar Tampilan LCD



Gambar sistem keseluruhan



Gambar sensor flowmeter



Gambar tampilan start awal sistem



Gambar akuarium air keruh



Gambar medium air terlarut



Gambar pengukuran medium air terlarut 1



Gambar pengukuran medium air terlarut 2



Gambar pengukuran medium air terlarut 3



Gambar pengukuran medium air terlarut 4



Gambar pengukuran medium air terlarut 5



Gambar pengukuran medium air terlarut 6



Gambar pengukuran medium air terlarut 7

B. LAMPIRAN LISTING PROGRAM ARDUINO

```
#include <LiquidCrystal.h>
#include <LCDKeypad.h>
#define pompapin 3
#define Pinkekeruhan1 A2
#define Pinkekeruhan2 A1
#define Ts 0.5
#define MINVAL 0
#define MAXVAL 100

float Kp ,Ti, Td;
float nilaikekeruhan1,nilaikekeruhan2 ;
float sensorkekeruhan1 =0 ;
float sensorkekeruhan2 =0 ;
float kontrol_pompa = 0 ;
int co, Sp, tmp,tmpD;
float out_P, out_I, out_D,out_PID, co_PID;
float error, d_error,errorI_1,errorD_1,prev_error;
float a, b, c, d, e, f, g, da, db, dc, dd, de, df, dg, j, k;
float w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11, w12, w13, w14, w15, w16, w17,
w18, w19, w20, w21, w22, w23, w24, w25;
int K1,S1,B,SB1,BS1,K2,S2,B2, SB2, BS2,K3, S3,B3,SB3, BS3;
float uErrorNB, uErrorN, uErrorZ, uErrorP, uErrorPB, uDErrorNB, uDErrorN, uDErrorZ,
uDErrorP, uDErrorPB;
float last,aktual;
float maksimal= 50;
float minimal= 0.5;
int bottom=MINVAL, top=MAXVAL;
int trynumber=0;
```

```
int guess, buttonPressed;  
volatile int NbTopsFan; //measuring the rising edges of the signal  
int Calc;  
int hallsensor = 2; //The pin location of the sensor
```

```
LCDKeypad lcd;
```

```
byte c_up[8] = {  
    B00100,  
    B01110,  
    B10101,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
};
```

```
byte c_down[8] = {  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B00100,  
    B10101,  
    B01110,  
    B00100,  
};
```

```
byte c_select[8] = {
```

```
B00000,  
B01110,  
B11111,  
B11111,  
B11111,  
B11111,  
B11111,  
B01110,  
B00000,  
};  
void rpm () //This is the function that the interupt calls  
{  
    NbTopsFan++; //This function measures the rising and falling edge of the hall effect  
sensors signal  
}  
  
void setup()  
{  
    pinMode(pompapin, OUTPUT);  
    pinMode(Pinkekeruhan1, INPUT);  
    pinMode(Pinkekeruhan2, INPUT);  
    pinMode(hallsensor, INPUT); //initializes digital pin 2 as an input  
    Serial.begin(9600); //This is the setup function where the serial port is initialised,  
attachInterrupt(0, rpm, RISING); //and the interrupt is attached  
    int i,k;  
  
    lcd.createChar(1,c_select);  
    lcd.createChar(2,c_up);  
    lcd.createChar(3,c_down);  
    lcd.begin(16, 2);  
    lcd.clear();
```

```
lcd.print("  welcome");
lcd.setCursor(0,1);
lcd.print("  please wait");
delay(3000);
for (k=0;k<3;k++)
{
  lcd.scrollDisplayLeft();
  delay(100);
}
for (i=0;i<3;i++)
{
  for (k=0;k<6;k++)
  {
    lcd.scrollDisplayRight();
    delay(100);
  }
  for (k=0;k<6;k++)
  {
    lcd.scrollDisplayLeft();
    delay(100);
  }
}
for (k=0;k<16;k++)
{
  lcd.scrollDisplayLeft();
  delay(100);
}
lcd.clear();
lcd.print("masukkan SP");
lcd.setCursor(0,1);
lcd.print("from ");
```

```
lcd.print(MINVAL,DEC);
lcd.print(" to ");
lcd.print(MAXVAL,DEC);
lcd.print(" ");
waitButton();
waitReleaseButton();
do
{
  lcd.clear();
  guess=bottom+(top-bottom)/4;
  trynumber=guess ;

  lcd.print("Is it ");
  lcd.print(guess,DEC);
  lcd.print("?");
  lcd.setCursor(0,1);
  lcd.write(1);
  lcd.write(' ');
  lcd.write(2);
  lcd.write(' ');
  lcd.write(3);
  lcd.write(' ');
  do
  {
    buttonPressed=waitButton();
  }
  while(!(buttonPressed==KEYPAD_SELECT || buttonPressed==KEYPAD_UP ||
buttonPressed==KEYPAD_DOWN));
  lcd.setCursor(0,1);
  lcd.write(buttonPressed==KEYPAD_SELECT?1:' ');
  lcd.write(' ');
```



```
lcd.write(buttonPressed==KEYPAD_UP?'2: ');
lcd.write(' ');
lcd.write(buttonPressed==KEYPAD_DOWN?'3: ');
delay(100);
waitReleaseButton();
if (buttonPressed==KEYPAD_UP)
{
  bottom=constrain(guess+1,MINVAL,top);
}
else if (buttonPressed==KEYPAD_DOWN)
{
  top=constrain(guess-1,bottom,MAXVAL);
}
}
while (buttonPressed!=KEYPAD_SELECT && top!=bottom);
lcd.clear();
if (top==bottom)
{
  lcd.print("to large ");
  waitReleaseButton();
}
else
{
  lcd.print("please wait...");
}
lcd.print(guess,DEC);
lcd.print("!");
lcd.setCursor(0,1);
lcd.print("Setpoint: ");
lcd.print(trynumber,DEC);
lcd.print(" ");
```

```
delay (2000);
lcd.clear();
waitButton();
waitReleaseButton();
}
void loop()
{

NbTopsFan = 0; //Set NbTops to 0 ready for calculations
sei(); //Enables interrupts
delay (1000); //Wait 1 second
cli(); //Disable interrupts
Calc = (NbTopsFan * 60 / 5.5); //(Pulse frequency x 60) / 5.5Q, = flow rate in L/hour
lcd.setCursor(8,1);
lcd.print("flow: "); //Co differensial
lcd.setCursor(13,1);
lcd.print(Calc);
lcd.print(" L/hour\r\n");
Serial.print (Calc, DEC); //Prints the number calculated above
Serial.print (" L/h\r\n"); //Prints "L/hour" and returns a new line

sensorkekeruhan2 = analogRead(Pinkekeruhan2);
nilaikekeruhan2 = (((sensorkekeruhan2 * 5.0 / 1023.0)/5)*100)+ 6;

lcd.setCursor(8,0);
lcd.print("Tout: ");
lcd.setCursor(13,0);
lcd.print(nilaikekeruhan2);
Serial.print("Tout:");
Serial.println(nilaikekeruhan2);
sensorkekeruhan1 = analogRead(Pinkekeruhan1);
```

$nilaikekeruhan1 = (((sensorkekeruhan1 * 5.0 / 1023.0)/5)*100)+ 9;$

```
lcd.setCursor(0,0);  
lcd.print("Tin: ");  
lcd.setCursor(5,0);  
lcd.print(nilaikekeruhan1);  
Serial.print("Tin:");  
Serial.println(nilaikekeruhan1);  
if (nilaikekeruhan1<=60)  
{kontrol_pompa=0;}  
fuzzifikasi();  
implikasi();  
defuzzifikasi();  
kontrol();  
}
```

```
void fuzzifikasi()  
{ Sp=trynumber ;  
aktual=nilaikekeruhan2;  
Serial.print("SP:");  
Serial.println(Sp);  
error = Sp - aktual ;  
Serial.print("error:");  
Serial.println(error);
```

```
a=-50;
```

```
b=-5;
```

```
c=-1;
```

```
d=0;
```

```
e=1;
```

f=5;

g=50;

```
if (error>=a && error<=b)
```

```
{
```

```
  uErrorNB = 1*1;
```

```
  uErrorN = 0;
```

```
  uErrorZ = 0;
```

```
  uErrorP = 0;
```

```
  uErrorPB = 0;
```

```
}
```

```
// jika error berada di daerah antara titik b dan c
```

```
if (error>=b && error<=c)
```

```
{
```

```
  uErrorNB = 1*(c - error)/(c - b);
```

```
  uErrorN = 1*(error - b)/(c - b);
```

```
  uErrorZ = 0;
```

```
  uErrorP = 0;
```

```
  uErrorPB = 0;
```

```
}
```

```
// jika error berada di daerah antara titik c dan d
```

```
if(error>=c && error<=d)
```

```
{
```

```
  uErrorNB = 0;
```

```
  uErrorN = 1*((d - error)/(d - c));
```

```
  uErrorZ = 1*((error - c)/(d - c));
```

```
  uErrorP = 0;
```

```
  uErrorPB = 0;
```

```
}
```

```
// jika error berada di daerah antara titik d dan e
if(error>=d && error <=e)
{
uErrorNB = 0;
uErrorN = 0;
uErrorZ = 1*(e - error)/(e - d);
uErrorP = 1*(error - d)/(e - d);
uErrorPB = 0;
}

// jika error berada di daerah antara titik e dan f
if(error>=e && error<=f)
{
uErrorNB = 0;
uErrorN = 0;
uErrorZ = 0;
uErrorP = 1*(f - error)/(f - e);
uErrorPB = 1*(error - e)/(f - e);
}

//jika error berada di daerah antara titik f dan g
if(error>=f && error<=g)
{
uErrorNB = 0;
uErrorN = 0;
uErrorZ = 0;
uErrorP = 0;
uErrorPB = 1*1; }

// FUZIFIKASI_PERUBAHAN_ERROR();
```

```
d_error = error - prev_error ;
```

```
da=-5;
```

```
db=-2;
```

```
dc=-1;
```

```
dd=0;
```

```
de=1;
```

```
df=2;
```

```
dg=5;
```

```
Serial.print("D_error:");
```

```
Serial.println(d_error);
```

```
// jika d_error berada di daerah antara titik da dan db
```

```
if (d_error>=da && d_error<=db)
```

```
{
```

```
uDErrorNB = 1*1;
```

```
uDErrorN = 0;
```

```
uDErrorZ = 0;
```

```
uDErrorP = 0;
```

```
uDErrorPB = 0;
```

```
}
```

```
// jika d_error berada di daerah antara titik db dan dc
```

```
if (d_error>=db && d_error<=dc)
```

```
{
```

```
uDErrorNB = 1*(dc - d_error)/(dc - db);
```

```
uDErrorN = 1*(d_error - db)/(dc - db);
```

```
uDErrorZ = 0;
```

```
uDErrorP = 0;
```

```
uDErrorPB = 0;
```

```
}
```

```
// jika d_error berada di daerah antara titik dc dan dd
```

```
if(d_error>=dc && d_error<=dd)
```

```
{
```

```
uDErrorNB = 0;
```

```
uDErrorN = 1*((dd - d_error)/(dd - dc));
```

```
uDErrorZ = 1*((d_error - dc)/(dd - dc));
```

```
uDErrorP = 0;
```

```
uDErrorPB = 0;
```

```
}
```

```
// jika d_error berada di daerah antara titik dd dan de
```

```
if(d_error>=dd && d_error <=de)
```

```
{
```

```
uDErrorNB = 0;
```

```
uDErrorN = 0;
```

```
uDErrorZ = 1*(de - d_error)/(de - dd);
```

```
uDErrorP = 1*(d_error - dd)/(de - dd);
```

```
uDErrorPB = 0;
```

```
}
```

```
// jika d_error berada di daerah antara titik de dan df
```

```
if(d_error>=de && d_error<=df)
```

```
{
```

```
uDErrorNB = 0;
```

```
uDErrorN = 0;
```

```
uDErrorZ = 0;
```

```
uDErrorP = 1*(df - d_error)/(df - de);
```

```
uDErrorPB = 1*(d_error - de)/(df - de);
```

```
}
```

```
//jika d_error berada di daerah antara titik df dan dg
```

```
if(d_error>=df && d_error<=dg)
{
  uDErrorNB = 0;
  uDErrorN = 0;
  uDErrorZ = 0;
  uDErrorP = 0;
  uDErrorPB = 1*1;
}
```

```
prev_error = error;
delay (100);
}
```

```
void implikasi()
{
```

```
// untuk kondisi dekat
```

```
w1 =min(uDErrorNB,uErrorNB);
```

```
Serial.print("w1:");
```

```
Serial.println(w1);
```

```
w2 =min(uDErrorNB,uErrorN);
```

```
Serial.print("w2:");
```

```
Serial.println(w2);
```

```
w3 =min(uDErrorNB,uErrorZ);
```

```
Serial.print("w3:");
```

```
Serial.println(w3);
```

```
w4 =min(uDErrorNB,uErrorP);
```

```
Serial.print("w4:");
```

```
Serial.println(w4);
```

```
w5 =min(uDErrorNB,uErrorPB);
```

```
Serial.print("w5:");
```

```
Serial.println(w5);
```



```
w6 =min(uDErrorN,uErrorNB);
Serial.print("w6:");
Serial.println(w6);
w7 =min(uDErrorN,uErrorN);
Serial.print("w7:");
Serial.println(w7);
w8 =min(uDErrorN,uErrorZ);
Serial.print("w8:");
Serial.println(w8);
w9 =min(uDErrorN,uErrorP);
Serial.print("w9:");
Serial.println(w9);
w10 =min(uDErrorN,uErrorPB);
Serial.print("w10:");
Serial.println(w10);
w11 =min(uDErrorZ,uErrorNB);
Serial.print("w11:");
Serial.println(w11);
w12 =min(uDErrorZ,uErrorN);
Serial.print("w12:");
Serial.println(w12);
w13 =min(uDErrorZ,uErrorZ);
Serial.print("w13:");
Serial.println(w13);
w14 =min(uDErrorZ,uErrorP);
Serial.print("w14:");
Serial.println(w14);
w15 =min(uDErrorZ,uErrorPB);
Serial.print("w15:");
Serial.println(w15);
w16 =min(uDErrorP,uErrorNB);
```

```
Serial.print("w16:");
Serial.println(w16);
w17 =min(uDErrorP,uErrorN);
Serial.print("w17:");
Serial.println(w17);
w18 =min(uDErrorP,uErrorZ);
Serial.print("w18:");
Serial.println(w18);
w19 =min(uDErrorP,uErrorP);
Serial.print("w19:");
Serial.println(w19);
w20 =min(uDErrorP,uErrorPB);
Serial.print("w20:");
Serial.println(w20);
w21 =min(uDErrorPB,uErrorNB);
Serial.print("w21:");
Serial.println(w21);
w22 =min(uDErrorPB,uErrorN);
Serial.print("w22:");
Serial.println(w22);
w23 =min(uDErrorPB,uErrorZ);
Serial.print("w23:");
Serial.println(w23);
w24 =min(uDErrorPB,uErrorP);
Serial.print("w24:");
Serial.println(w24);
w25 =min(uDErrorPB,uErrorPB);
Serial.print("w25:");
Serial.println(w25);
delay (100);
}
```

```
void defuzzifikasi()
```

```
{
```

```
    K1=0;
```

```
    S1=2;
```

```
    B=4;
```

```
    SB1=10;
```

```
    BS1=15;
```

```
    Kp
```

```
    =(w1*K1+w2*K1+w3*K1+w4*K1+w5*K1+w6*K1+w7*K1+w8*K1+w9*S1+w10*B+w11*K1+w12*K1+w13*K1+w14*B+w15*SB1+w16*K1+w17*K1+w18*K1+w19*B+w20*S1+w21*K1+w22*K1+w23*K1+w24*SB1+w25*BS1) /
```

```
    (w1+w2+w3+w4+w5+w6+w7+w8+w9+w10+w11+w12+w13+w14+w15+w16+w17+w18+w19+w20+w21+w22+w23+w24+w25);
```

```
    Serial.print("Kp:");
```

```
    Serial.println(Kp);
```

```
    K2= 0;
```

```
    S2= 5;
```

```
    B2= 10;
```

```
    SB2= 15;
```

```
    BS2= 20;
```

```
    Ti=
```

```
    (w1*K2+w2*K2+w3*K2+w4*S2+w5*S2+w6*K2+w7*K2+w8*K2+w9*S2+w10*S2+w11*K2+w12*K2+w13*K2+w14*S2+w15*S2+w16*K2+w17*K2+w18*K2+w19*S2+w20*B2+w21*K2+w22*K2+w23*K2+w24*B2+w25*B2) /
```

```
    (w1+w2+w3+w4+w5+w6+w7+w8+w9+w10+w11+w12+w13+w14+w15+w16+w17+w18+w19+w20+w21+w22+w23+w24+w25);
```

```
    Serial.print("Ti:");
```

```
Serial.println(Ti);
```

```
K3= 7;
```

```
S3= 5;
```

```
B3= 3;
```

```
SB3= 1;
```

```
BS3= 0;
```

```
Td =
```

```
(w1*BS3+w2*BS3+w3*S3+w4*K3+w5*K3+w6*BS3+w7*BS3+w8*S3+w9*K3+w10*K3  
+w11*BS3+w12*SB3+w13*K3+w14*K3+w15*K3+w16*BS3+w17*SB3+w18*K3+w19*  
K3+w20*K3+w21*BS3+w22*SB3+w23*S3+w24*K3+w25*K3) /
```

```
(w1+w2+w3+w4+w5+w6+w7+w8+w9+w10+w11+w12+w13+w14+w15+w16+w17+w18+  
w19+w20+w21+w22+w23+w24+w25);
```

```
Serial.print("Td:");
```

```
Serial.println(Td);
```

```
delay(100);
```

```
}
```

```
void kontrol()
```

```
{
```

```
out_P = Kp * error;
```

```
Serial.print("out_P:");
```

```
Serial.println(out_P);
```

```
//kontrol integral
```

```
if(Ti!=0)
```

```
{tmp=errorI_1 + error;
```

```
out_I = (tmp * Ts * Kp/Ti); } //Co integral}
```

```
errorI_1=error;
```

```
if (out_I>=maksimal)
{
    out_I=maksimal;
}

if(out_I<=minimal)
{
    out_I=minimal;
}
Serial.print("out_I:");
Serial.println(out_I);

//kontrol diferensial
tmpD = error - errorD_1;

out_D = (Kp*Td*tmpD)/Ts;
last=aktual;
Serial.print("out_D:");
Serial.println(out_D);
errorD_1 = error ;
//kontrol PID
out_PID= out_P + out_I + out_D;
if (out_PID>=255)
{out_PID = 255;}
else {out_PID = out_PID;}

lcd.setCursor(0,1);
lcd.print("PID: ");//Co proporsional
lcd.setCursor(4,1);
lcd.print(out_PID);
Serial.print ("PID;");
```

```
Serial.println(out_PID);

kontrol_pompa = 255 - out_PID ;

    analogWrite(pompapin, kontrol_pompa );
Serial.print("kontrol pompa:");
Serial.println(kontrol_pompa);
delay(100);
}
int waitButton()
{
    int buttonPressed;
    waitReleaseButton;
    lcd.blink();
    while((buttonPressed= lcd.button())==KEYPAD_NONE)
    {
    }
    lcd.print ( "please select" );
    delay(50);
    lcd.noBlink();
    return buttonPressed;
}

void waitReleaseButton()
{
    delay(50);
    while(lcd.button()!=KEYPAD_NONE)
    {
    }
    delay(50);
}
```

