



TUGAS AKHIR TF 141581

**PERANCANGAN SISTEM KONTROL SUDUT *PITCH*
BERBASIS INTERVAL LOGIKA FUZZY TIPE-2 PADA
PROTOTYPE TURBIN ANGIN SKALA KECIL**

**MARIO ARDHANY
NRP 2414 106 031**

Dosen Pembimbing
Dr. Ir. Ali Musyafa', M.Sc

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT TF-141581

**PITCH ANGLE CONTROL SYSTEM DESIGN BASED
ON INTERVAL TYPE 2 FUZZY LOGIC FOR SMALL
SCALE WIND TURBINE PROTOTYPE**

MARIO ARDHANY
NRP 2414 106 031

Supervisor
Dr. Ir. Ali Musyafa', M.Sc

ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

**PERANCANGAN SISTEM KONTROL SUDUT *PITCH*
BERBASIS INTERVAL LOGIKA FUZZY TIPE-2 PADA
PROTOTYPE TURBIN ANGIN SKALA KECIL**

TUGAS AKHIR

Oleh :

Mario Ardhany
NRP : 2414 106 031

Surabaya, 22 Januari 2017
Mengetahui/Menyetujui

Pembimbing



Dr. Ir. Ali Musyafa', M.Sc.
NIP. 19600901 198701 1 001



Ketua Departemen
Teknik Fisika FTI-ITS

Agus Muhamad Hatta, S.T., M.Si., Ph.D.
NIP. 19780902 200312 1 002

LEMBAR PENGESAHAN

PERANCANGAN SISTEM KONTROL SUDUT *PITCH* BERBASIS INTERVAL LOGIKA FUZZY TIPE-2 PADA PROTOTYPE TURBIN ANGIN SKALA KECIL





TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi
Program Studi S-1 Jurusan Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

Mario Ardhanay
NRP. 2414 106 031

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Ali Musyafa', M.Sc.  (Pembimbing I)
2. Dr. Ir. Purwadi Agus D., M.Sc.  (Ketua Penguji)
3. Dr. Imam Abadi, S.T., M.T.  (Penguji I)
4. Dyah Sawitri, S.T., M.T.  (Penguji II)

SURABAYA
22 Januari 2017

PERANCANGAN SISTEM KONTROL SUDUT PITCH BERBASIS INTERVAL LOGIKA FUZZY TIPE-2 PADA PROTOTYPE TURBIN ANGIN SKALA KECIL

Nama Mahasiswa : Mario Ardhany
NRP : 2414 106 031
Program Studi : S1 Teknik Fisika
Dosen Pembimbing : Dr. Ir. Ali Musyafa', M.Sc

Abstrak

Energi angin dapat dimanfaatkan untuk menghasilkan energi listrik, untuk memanfaatkan energi tersebut, salah satu alat yang dapat digunakan adalah turbin angin, dengan faktor kecepatan angin yang bervariasi akan menjadi kendala dalam desain turbin angin tersebut. Faktor tersebut akan mempengaruhi kecepatan putar turbin angin. Dalam tugas akhir ini telah dirancang sebuah sistem pengendalian sudut pitch pada prototipe turbin angin berbasis interval type II fuzzy logic dengan tujuan mendapatkan performansi turbin angin yang optimal. Bilah yang digunakan bertipe non-uniform blade untuk jenis kecepatan angin rendah dan turbin angin berukuran kecil dengan profil penampang airfoil NREL S83n. Pengujian kontrol dilakukan dengan setpoint 10, 15, 20 dan 25 pps. Berdasarkan hasil pengujian didapatkan data rise time dan settling time tercepat sebesar 38 detik dan 29 detik saat setpoint 10, sedangkan yang terlama 140 detik dan 134 detik saat setpoint 25. Maksimum overshoot tertinggi sebesar 10% saat setpoint 10 dan terendah sebesar 0% saat setpoint 25. Error steady state tertinggi sebesar 5,1% saat setpoint 10 dan terendah sebesar 2,06% saat setpoint 15. Nilai ITAE terendah 307 saat setpoint 10, ITAE tertinggi 1787 saat setpoint 25. Sistem pengendalian yang terbaik untuk respon sistem turbin angin ini yaitu saat setpoint 25, karena memiliki error steady yang kecil (dibawah toleransi 5%) dan tidak adanya maksimum overshoot.

Kata Kunci: turbin angin, sudut pitch, interval type II fuzzy logic, maksimum overshoot, error steady state.

PITCH ANGLE CONTROL SYSTEM DESIGN BASED ON INTERVAL TYPE 2 FUZZY LOGIC FOR SMALL SCALE WIND TURBINE PROTOTYPE

Name of Student : Mario Ardhany
Number Of Registration : 2414 106 031
Department : Engineering Physics FTI-ITS
Supervisor : Dr. Ir. Ali Musyafa', M.Sc

Abstract

Wind energy can be used to generate electrical energy. Wind turbine is one of the instrument that can be used to take benefit of this wind energy, speed variation of the wind will be a factor of consideration in designing wind turbine. That factor will affect the rotational speed of wind turbines. This final project has designed a blade pitch angle control system control for wind turbine prototype based on interval type II fuzzy logic. The aim is to obtain results in form of optimal rotation of wind turbine. Blades used non-uniform type for low wind speeds and small-sized wind turbines with NREL airfoil profile S83n. Setpoint control test carried out with 10, 15, 20 and 25 pps. From the results of the test, it is obtained that : the fastest rise time and settling time at 38 seconds and 29 seconds when setpoint 10, whereas the longest 140 seconds and 134 seconds when setpoint 25. Maximum overshoot peak 10% when setpoint 10 and lowest 0% when setpoint 25. The highest steady state error 5.1% when setpoint 10 and the lowest 2.06% when setpoint 15. The lowest ITAE value 307 when setpoint 10, while the highest value 1787 when setpoint 25. The best control system for wind turbine system response when setpoint 25 pps, because it has small error steady state (under a tolerance of 5 %) and the maximum overshoot is 0%.

Keywords : wind turbine, pitch angle, interval type 2 fuzzy logic, maximum overshoot, error steady state

KATA PENGANTAR

Alhamdulillah segala puji hanya bagi Allah SWT dzat yang menurunkan Al-Qur'an dan dengan-Nya Ia mengangkat derajat manusia, serta berkat limpahan rahmat, taufiq, karunia, pertolongan, nikmat, terutama atas hidayah-Nya, penulis dapat menyelesaikan penulisan laporan tugas akhir ini dengan judul "Perancangan Sistem Kontrol Sudut Pitch Berbasis Interval Logika Fuzzy Tipe-2 Pada Prototipe Turbin Angin Skala Kecil". Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan laporan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan laporan ini. Oleh karena itu, penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu diantaranya:

1. Bapak Agus Muhamad Hatta, S.T, M.Si, Ph.D selaku Ketua Jurusan Teknik Fisika FTI-ITS Surabaya dan bapak Dr. Ir. Purwadi Agus Darwito, M.Sc. selaku dosen wali penulis yang telah memberikan bimbingan dan ilmu yang sangat bermanfaat.
2. Bapak dan Ibu yang telah banyak memberikan motivasi, dukungan, bimbingan, do'a-do'a yang terus menerus mengalir, serta kakak (Rini N, Eni A, Didik M, Endang M) yang terus memberi dukungan moral dan do'a yang tak pernah berhenti.
3. Bapak Dr. Ir. Ali Musyafa', M.Sc selaku dosen pembimbing yang telah sabar untuk membimbing dan memotivasi dalam menyelesaikan tugas akhir ini, terima kasih sebesar-sebsarnya atas segala bantuan dan perhatiannya.
4. Bapak dan Ibu dosen pengajar di jurusan Teknik Fisika ITS.
5. Rekan seperjuangan Eko Mamo Prapitag Tinomo Putro dab Jauharotul Maknunah yang telah banyak membantu dan merasakan susah payah, jerih payah bersama, dalam membuat plant turbin angin, perjuangan membuat hardware, membeli kipas dan lain-lain, hingga perancangan kontrol, melewati susah senang bersama.

6. Semua rekan dari Laboratorium Pengukuran Fisis yang telah menemani, memberikan masukan dan menjadi tempat setiap pengujian berlangsung untuk pengambilan data.
7. Pak didik dan seluruh teknisi laboratorium Non Metal PPNS yang telah membantu dan memberikan banyak ilmu serta pengalamannya.
8. Teman-teman seperjuangan angkatan LJ Genap 2014 yang selalu memberikan dukungan dan motivasi dalam menyelesaikan tugas akhir ini.
9. Teman-teman seperjuangan tugas akhir , TA-wan dan TA-wati yang bersama-sama mengerjakan tugas akhir.
10. Pihak Jurusan Teknik Fisika Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember yang memfasilitasi dalam hal administrasi.
11. Serta seluruh pihak yang telah banyak membantu dan mendukung yang tidak dapat penulis sebutkan satu per satu.

Dengan segala kekurangan dan keterbatasan, penulis sadar bahwa penulisan laporan tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kami sangat mengharapkan kritik, saran dan koreksi yang membangun dari berbagai pihak yang ditujukan pada kebaikan laporan tugas akhir ini. Penulis juga berharap semoga laporan ini dapat memberikan informasi, wawasan dan wacana yang bermanfaat untuk perkembangan ilmu pengetahuan dan bagi dunia *instrument wind turbine*.

Surabaya, Januari 2016

Penulis

DAFTAR ISI

	Hal
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
ABSTRAK.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	3
BAB II DASAR TEORI	5
2.1 Prinsip Turbin Angin.....	5
2.2 Bilah Turbin Angin.....	8
2.2.1 Dimensi Rotor dan Jumlah Bilah.....	9
2.2.2 Sudut <i>Pitch</i>	9
2.2.3 Panjang <i>Chord</i> dan Dimensi <i>Taper</i>	10
2.2.4 Bahan Bilah.....	11
2.2.5 Jenis <i>Airfoil</i>	11
2.3 Pengaturan Sudut <i>Pitch</i> Bilah Turbin Angin.....	13
2.3.1 Arduino Uno.....	14
2.3.2 Motor Servo.....	16
2.3.3 <i>Rotary Encoder</i>	19
2.3.4 <i>Photointerrupter</i>	19
2.3.5 <i>Slip Ring</i>	20
2.4 Logika <i>Fuzzy</i> Tipe-2.....	20
2.4.1 Logika <i>Fuzzy</i>	21
2.4.2 Interval Logika <i>Fuzzy</i> Tipe-2.....	24
BAB III METODOLOGI PENELITIAN	29
3.1 Prosedur Umum Penelitian.....	29
3.1.1 Perancangan <i>Prototype</i> Turbin Angin.....	30

3.1.2 Perancangan Algoritma Kontrol Menggunakan Interval <i>Fuzzy Type 2</i>	30
3.1.3 Simulasi Algoritma Kontrol dengan Menggunakan Software Komputasi	30
3.1.4 Pemrograman Arduino	30
3.1.5 <i>Interface Hardware dan Software</i>	31
3.1.6 Pengambilan Data	31
3.1.7 Analisis Data dan Pembahasan	31
3.2 <i>Protoype Turbin Angin</i>	31
3.2.1 <i>Blade Turbin Angin</i>	31
3.2.2 <i>Aktuator</i>	32
3.2.3 Sensor	34
3.2.4 <i>Rotatial Connector</i> atau <i>Slip Ring</i>	37
3.3 Pengambilan Data Awal	39
3.4 Perancangan Algoritma Pengendalian Interval <i>Fuzzy Tipe 2</i>	45
3.5 Perancangan Fungsi Keanggotaan Interval <i>Fuzzy Tipe 2</i>	47
3.6 Perancangan <i>Rule Base</i>	51
BAB IV ANALISIS DATA DAN PEMBAHASAN	55
4.1 Analisis Kinerja Turbin Angin	55
4.1.1 Nilai RPM Maksimum	57
4.1.2 Koefisien Daya / <i>Coefficient Power</i>	58
4.2 Pengujian Akurasi	60
4.3 Analisis Respon Sistem Pengendalian	62
4.3.1 Analisis Pengujian <i>Setpoint</i>	62
4.3.2 Analisis Pengujian <i>Tracking Setpoint</i>	67
BAB V PENUTUP	71
5.1 Kesimpulan	71
5.2 Saran	72

DAFTAR PUSTAKA
LAMPIRAN

DAFTAR GAMBAR

	Hal
Gambar 2.1 Vektor gaya pada <i>airfoil</i> dengan sudut serang berbeda	5
Gambar 2.2 <i>Tube</i> angin yang melewati turbin angin	6
Gambar 2.3 Sudut <i>pitch</i> bilah turbin	9
Gambar 2.4 Penampang <i>airfoil</i> NREL S835 untuk pangkal bilah.....	13
Gambar 2.5 Penampang <i>airfoil</i> NREL S833 untuk tengah bilah.....	13
Gambar 2.6 Penampang <i>airfoil</i> NREL S834 untuk ujung bilah.....	13
Gambar 2.7 Papan Arduino Uno	14
Gambar 2.8 Sinyal kendali motor servo	17
Gambar 2.9 Bagian dalam motor servo.....	18
Gambar 2.10 Pemasangan <i>photointerrupter</i>	19
Gambar 2.11 <i>Slip ring</i>	20
Gambar 2.12 Himpunan <i>fuzzy</i> untuk variabel umur	22
Gambar 2.13 Himpunan <i>fuzzy</i> pada variabel temperatur	22
Gambar 2.14 <i>Footprint of Certainty</i>	25
Gambar 2.15 Operasi pada <i>membership function</i> tipe 2.....	25
Gambar 2.16 <i>Fuzzy Inference System</i> pada tipe-2.....	26
Gambar 2.17 Diagram blok sistem <i>fuzzy</i> tipe 2.....	26
Gambar 3.1 Diagram alir penelitian.....	29
Gambar 3.2 <i>Blade</i> turbin angin	31
Gambar 3.3 Penampang <i>airfoil</i> NREL S833, S834 dan S835 .	32
Gambar 3.4 Motor Servo Hitec HS-805BB	33
Gambar 3.5 <i>Rotary encoder</i> dan <i>optocoupler</i>	36
Gambar 3.6 <i>Slip ring</i>	37
Gambar 3.7 <i>Prototype</i> turbin angin keseluruhan.....	39
Gambar 3.8 Tiga buah blower.....	40
Gambar 3.9 Diagram blok sistem pengendalian.....	45
Gambar 3.10 Diagram alir algoritma interval tipe 2	46
Gambar 3.11 Grafik fungsi keanggotaan <i>input error</i>	48
Gambar 3.12 Grafik fungsi keanggotaan input <i>delta error</i>	48

Gambar 3.13	Fungsi keanggotaan output.....	49
Gambar 3.14	Tampilan <i>rule editor</i> Matlab 2014a.....	53
Gambar 3.15	Tampilan <i>rule viewer</i> Matlab 2014a.....	54
Gambar 3.16	<i>Surface view</i> pada Matlab 2014a.....	54
Gambar 4.1	Hubungan sudut pitch turbin angin dengan Kecepatan sudut.....	57
Gambar 4.2	Respon sistem saat <i>setpoint</i> = 10 pps	63
Gambar 4.3	Respon sistem saat <i>setpoint</i> = 15 pps	63
Gambar 4.4	Respon sistem saat <i>setpoint</i> = 20 pps	64
Gambar 4.5	Respon sistem saat <i>setpoint</i> = 25 pps	65
Gambar 4.6	Respon sistem saat pengujian <i>tracking setpoint</i> ..	68

DAFTAR TABEL

	Hal
Tabel 2.1 Perbandingan bahan bilah turbin angin.....	11
Tabel 2.2 Spesifikasi desain <i>airfoil</i> NREL S83n	12
Tabel 2.3 Alokasi pin arduino uno.....	15
Tabel 3.1 Berat penyeimbangan <i>blade</i> turbin angin	32
Tabel 3.2 Spesifikasi motor servo Hitec HS-805BB.....	33
Tabel 3.3 Pengujian sudut servo	34
Tabel 3.4 Pengujian sensor kecepatan	36
Tabel 3.5 Spesifikasi arduino uno.....	36
Tabel 3.6 Spesifikasi <i>slip ring</i>	37
Tabel 3.7 Spesifikasi <i>prototype</i> turbin angin	38
Tabel 3.8 Data pengaruh kecepatan angin , sudut <i>blade</i> dan jarak terhadap kecepatan putar turbin angin	40
Tabel 3.9 Mode pengujian turbin angin dengan kecepatan angin variatif	42
Tabel 3.10 Data pengaruh kecepatan angin, sudut <i>blade</i> dan kecepatan putar turbin (PPS) dengan jarak 3 meter .	42
Tabel 3.11 Perancangan <i>rule base</i>	51
Tabel 3.12 Kondisi <i>error</i> dan <i>delta error</i>	52
Tabel 4.1 Data pengaruh kecepatan angin, sudut <i>blade</i> dan kecepatan putar turbin (RPM) dengan jarak 3 meter	55
Tabel 4.2 Nilai CP maksimum untuk setiap sudut pitch dan kecepatan angin	59
Tabel 4.3 Nilai daya yang diperoleh dari turbin angin	59
Tabel 4.4 Hasil nilai <i>output fuzzy</i> pada matlab dan arduino	60
Tabel 4.5 Analisa respon sistem pengendalian interval <i>fuzzy</i> tipe 2 saat pengujian <i>setpoint</i>	65
Tabel 4.6 Nilai ITAE dengan $t=400s$	66
Tabel 4.7 Analisa respon sistem pengendalian interval <i>fuzzy</i> tipe 2 saat pengujian <i>tracking setpoint</i>	69

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Energi angin merupakan salah satu energi yang berpotensi dapat menggantikan energi konvensional atau sumber energi terbarukan. Energi angin dapat dimanfaatkan untuk menghasilkan energi listrik atau konversi ke energi konvensional untuk keperluan lain. Untuk memanfaatkan energi tersebut, salah satu alat yang dapat digunakan adalah turbin angin, dengan faktor kecepatan angin yang bervariasi atau berubah-ubah akan menjadi kendala dalam desain turbin angin tersebut. Faktor tersebut akan mempengaruhi kecepatan putar turbin angin yang bergantung pada kecepatan angin, perancangan bilah untuk turbin angin yang meliputi ukuran (jari-jari rotor), penampang *airfoil*, panjang *chord*, dan sudut *pitch* bilah (Piggot, 2001).

Berdasarkan data kecepatan angin di berbagai wilayah, sumberdaya energi angin Indonesia berkisar antara 2,5 – 5,5 m/detik pada ketinggian 24 meter di atas permukaan tanah. Dengan kecepatan tersebut sumberdaya energi angin Indonesia termasuk dalam kategori kecepatan angin kelas rendah hingga menengah, sehingga belum banyak yang memanfaatkan energi ini. Konfigurasi turbin angin yang dapat digunakan adalah tipe HAWT (*Horizontal Axis Wind Turbine*) dengan berjumlah 3 buah dan tipe *airfoil* bilah NREL S83n (Buhl, M. 2009). *Airfoil* NREL S83n adalah tipe yang sesuai digunakan untuk turbin angin skala kecil (diameter 1-3 m) (Buhl, M. 2009). Parameter lain yang perlu diperhatikan dalam merancang bilah turbin angin adalah penentuan sudut *pitch* bilah turbin angin yang mana akan mempengaruhi performansi dari turbin angin, yaitu kecepatan sudut rotor, kecepatan angin optimal, dan koefisien daya. Dengan penentuan sudut *pitch* tersebut diharapkan kinerja turbin angin dengan skala kecepatan angin yang rendah dapat ditingkatkan.

Dalam tugas akhir ini didesain sebuah sistem pengendalian *blade pitch angle* pada *prototype* turbin angin yang berbasis *Interval Fuzzy Logic Type 2*, dengan berapa komponen untuk

menyusun *prototype* turbin angin yang digunakan pada penelitian ini menggunakan komponen pada penelitian sebelumnya yang telah dibuat oleh Farid Ridha Muttaqin tahun 2011. Dari *prototype* yang telah ada dilakukan pengembangan penambahan dimensi pada *blade* turbin yang digunakan untuk mendapatkan kecepatan putar turbin dengan efisiensi yang lebih optimal. Panjang *blade* yang digunakan sebelumnya memiliki panjang 80 cm dikembangkan pada panjang *bladenya* dengan panjang 100 cm dan juga mikroprosesor yang digunakan sebelumnya yaitu mikrokontroler, untuk tugas akhir ini digunakan mikroprosesor yang sederhana dan efisien dengan arduino. Pengembangan ini dilakukan agar mendapatkan daya ekstrak yang lebih optimal yang dihasilkan pada turbin angin. Penelitian tugas akhir ini dibuat bertujuan untuk mendapatkan hasil berupa putaran optimal serta bilah turbin angin yang dapat diatur sudutnya untuk mendapatkan daya maksimal yang dihasilkan pada turbin angin.

1.2 Perumusan Masalah

Dari latar belakang tersebut, diangkat beberapa permasalahan, yaitu :

- a. Bagaimana merancang *prototype* turbin angin skala kecil yang dimensinya dapat ditingkatkan sehingga mampu menghasilkan daya yang lebih besar?
- b. Bagaimana mengetahui kinerja sistem *prototype* turbin angin dengan menerapkan strategi kontrol sudut *pitch* dengan Interval Fuzzy Logic Type 2 ?

1.3 Tujuan

Tujuan dari penelitian ini berdasarkan permasalahan yang diuraikan adalah sebagai berikut ini:

- a. Untuk mendesain dan membangun sebuah sistem turbin angin untuk angin dengan kecepatan rendah.
- b. Untuk mendesain dan membangun sebuah sistem kontrol sudut *pitch* pada turbin angin.
- c. Mengetahui performansi dari turbin angin dengan kontrol sudut *pitch* yang telah dibuat.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini antara lain :

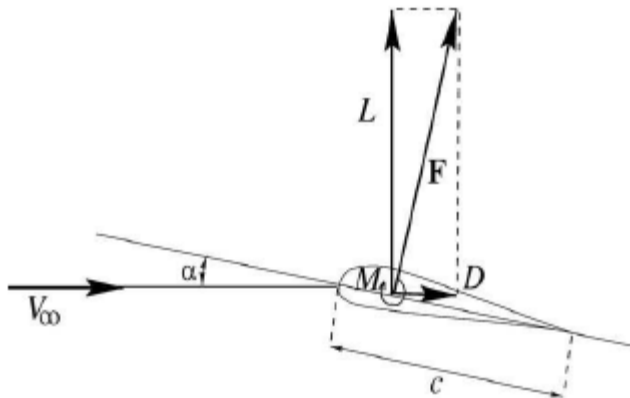
- a. Perancangan penampang bilah turbin angin mengacu pada literatur yang sudah ada dan diketahui performansinya, yaitu menggunakan *airfoil* NREL S835 untuk bagian pangkal bilah dan S833 untuk bagian tengah bilah, dan S834 untuk bagian ujung bilah.
- b. Sistem turbin angin yang digunakan adalah turbin angin berskala kecil dengan diameter bilah kurang lebih 2 meter.
- c. Variabel yang dimanipulasi adalah sudut *pitch* bilah turbin dan kecepatan angin
- d. Sistem kontroler berbasis kecerdasan buatan dengan metode Interval Fuzzy Logic Type 2.
- e. Mikroprosesor yang digunakan adalah jenis arduino.

Halaman ini sengaja dikosongkan

BAB II DASAR TEORI

2.1 Prinsip Turbin Angin

Prinsip dasar dari sebuah turbin angin bagaimana *blade* dapat berputar pada porosnya *blade* yang kena sapu angin sesuai dengan hukum kontinuitas akan menghasilkan vektor dari gaya *lift* dan gaya *drag* yang dihasilkan akibat bentuk *aerodinamis* dari penampang bilah turbin angin. Pada gambar 2.1 dijelaskan ketika sebuah penampang *airfoil* terkena angin dari arah depan, maka akan menghasilkan vektor gaya *lift* (L) dan *drag* (D) (Burton, et al., 2001). Gaya *lift* dan gaya *drag* ini perubahannya dipengaruhi oleh bentuk geometri bilah, kecepatan dan arah angin terhadap garis utama bilah (sudut *pitch*). Akibat perubahan gaya *lift* dan *drag*, maka kecepatan sudut dan torsi poros akan berubah pula. Perubahan sudut *pitch* bilah akan mempengaruhi kecepatan sudut (RPM) dari rotor karena adanya perubahan jumlah daya tiup angin yang diterima oleh bilah yang dikonversi menjadi kecepatan putar rotor (Adam. 2008).



Gambar 2.1 Vektor gaya pada *airfoil* dengan sudut serang berbeda (Burton, et al. 2001)

Daya dari angin yang dapat ditangkap oleh sebuah *horizontal axis wind turbine* (HAWT) dapat diturunkan dari persamaan energi kinetik angin yang bergerak dengan kecepatan tertentu ke arah x .

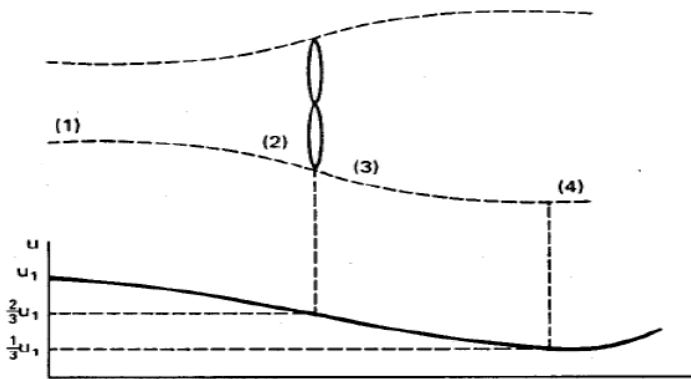
Adapun persamaan energi yang melewati turbin angin adalah sebagai berikut (Johnson, 2006) :

$$U = \frac{1}{2} m v_w^2 = \frac{1}{2} (\rho_{air} A_r x) v_w^2 \quad (2.1)$$

Diketahui bahwa daya adalah turunan dari energi terhadap waktu, maka:

$$P = \frac{dU}{dt} = \frac{1}{2} \rho_{air} A_r v_w^2 \frac{dx}{dt} = \frac{1}{2} \rho_{air} v_w^3 \quad (2.2)$$

Kemudian untuk mengetahui besarnya daya yang dapat diekstrak oleh turbin angin dapat dilakukan dengan menghitung selisih daya angin sebelum dan sesudah melewati turbin angin.



Gambar 2.2 Tube angin yang melewati turbin angin (Johnson, 2006)

Bila ada *tube* angin yang bergerak dengan kecepatan v akan melewati wind turbin, maka pastinya kecepatannya akan semakin berkurang sejalan dengan semakin dekatnya jaraknya terhadap turbin angin, karena tekanannya naik akibat ruang gerak yang semakin sempit (mampat). Ketika melewati turbin angin, maka energi kinetik angin tersebut diubah oleh turbin angin menjadi energi rotasional. Dan setelah jarak tertentu, maka kecepatan angin

akan kembali seperti semula akibat ruang geraknya telah melebar dan mendapat energi dari udara disekitarnya. Persamaan untuk muka tube angin yang melewati turbin angin ideal adalah sebagai berikut :

$$\begin{aligned}
 v_2 &= v_3 = \frac{2}{3} v_1 \\
 v_4 &= \frac{1}{3} v_1 \\
 A_2 &= A_3 = \frac{3}{2} A_1 \\
 A_4 &= 3A_1 \\
 P_{wind} &= \frac{1}{2} \rho_{air} A v^3 \quad (2.3)
 \end{aligned}$$

Jadi daya angin yang diekstrak adalah :

$$P = P_1 - P_4 = \frac{1}{2} \rho_{air} (A_1 v_1^3 - A_4 v_4^3) = \frac{1}{2} \rho_{air} \left(\frac{8}{9} A_1 v_1^3 \right) \quad (2.4)$$

Untuk mempermudah, maka persamaan kemudian dibawa ke bentuk lain yang menggunakan A_2 (sesuai luas area *wind turbin*).

$$P = \frac{1}{2} \rho_{air} \left[\frac{8}{9} \left(\frac{2}{3} A_2 \right) v_1^3 \right] = \frac{1}{2} \rho_{air} \left(\frac{16}{27} A_2 v_1^3 \right) \quad (2.5)$$

Dimana :

P = Daya turbin angin (watt)

ρ_{air} = Massa jenis udara (kg/m^3)

V = Kecepatan angin (m/s)

A_r = Luas sapuan penampang bilah (m^2)

C_p = *Power coefficient*

Pada persamaan tersebut terdapat angka $16/27$ yang merupakan bilangan Betz Limit. Yaitu daya maksimal yang bisa diekstrak oleh turbin angin adalah sebesar 59% untuk turbin angin ideal. Bilangan Betz Limit pada turbin angin *real* digantikan dengan koefisien daya (C_p) yang pasti memiliki nilai kurang dari 0.59. Semakin besar nilai C_p maka akan semakin besar power yang dapat ditangkap oleh turbin angin. C_p sendiri dipengaruhi dari

dari fungsi λ (*tip speed ratio*) dan θ (*pitch angle*) (Burton, et al, 2001). Jadi persamaan 2.4 dapat ditulis kembali menjadi:

$$P = \frac{1}{2} \rho_{air} C_p(\lambda, \theta) A_r v_w^3 \quad (2.6)$$

Sedangkan λ sendiri dirumuskan sbagai berikut :

$$\lambda = \frac{\omega R}{v} \quad (2.7)$$

Dimana :

λ = *tip speed ratio*

ω = kecepatan sudut (rps)

v = kecepatan angin (m/s)

R = jari-jari rotor bilah (m)

Jika diasumsikan ω adalah konstan sesuai *setpoint* yang diinginkan dan R bilah adalah konstan, maka C_p hanya akan bergantung pada v (kecepatan angin) dan θ (*pitch angle*), dari sinilah kemudian θ dijadikan variabel yang dimanipulasi sebagai kompensasi perubahan kecepatan angin (v) untuk mendapatkan power yang diinginkan. Sedangkan untuk mendapatkan θ sesuai dengan yang dibutuhkan dilakukan pengambilan data dengan menggunakan kecepatan angin (v) yang ditentukan.

Hubungan antara power dengan kecepatan sudut (ω) *shaft* adalah sebagai berikut (Burton, et al, 2001):

$$P = \tau_{aero} \omega \quad (2.8)$$

Dimana :

P = daya angin (watt)

τ_{aero} = Torsi *aero* dinamis dari rotor *blade* (Nm)

ω = kecepatan sudut rotor *blade* (rad/s)

Jadi daya (P) berbanding lurus dengan kecepatan sudut rotor turbin.

2.2 Bilah Turbin Angin

Pada turbin angin, desain bilah merupakan hal yang paling penting karena bilah merupakan komponen utama yang menangkap angin untuk kemudian dikonversikan menjadi gerakan mekanik. Perkembangan teknologi telah menciptakan bilah turbin angin dengan berbagai variasi bahan, ukuran, jenis *airfoil*, jumlah

bilah, dan lain sebagainya. Pada turbin angin *horizontal axis*, hal yang penting untuk diperhatikan adalah jari-jari bilah, jumlah bilah, sudut *pitch*, panjang *chord*, jenis *airfoil*, dan bahan bilah.

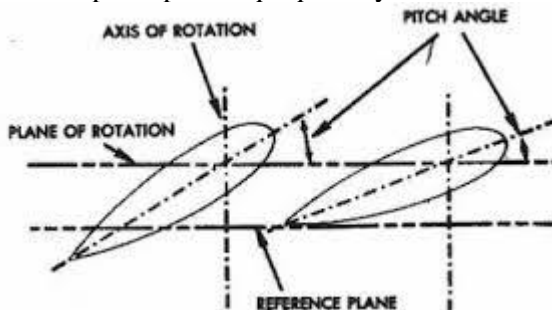
2.2.1. Dimensi Rotor dan Jumlah Bilah

Dimensi rotor turbin berpengaruh terhadap kemampuan turbin untuk menangkap angin yang melewati turbin. Semakin besar diameter rotor, maka semakin besar pula area sapuan angin yang dapat dimanfaatkan. Namun hal ini akan berpengaruh terhadap kecepatan rotor turbin tersebut, semakin besar rotor, maka koefisien daya akan makin besar dan kecepatan putaran turbin akan makin rendah

Jumlah bilah pada turbin angin tidak memiliki batasan khusus. Pada umumnya turbin angin *horizontal axis* memiliki tiga bilah atau banyak bilah. Pemilihan jumlah bilah pada turbin berdasarkan pada rancangan kecepatan turbin, adanya *noise*, dan estetika turbin angin (Rand, Joseph. 2010).

2.2.2. Sudut *Pitch*

Sudut *pitch* adalah sudut antara garis *chord* dengan bidang putar turbin angin. Sudut *pitch* pada bilah mengacu pada sudut serang dari bilah terhadap aliran fluida yang melewati. Pada pesawat terbang atau kapal, sudut *pitch propeler* dapat diatur sedemikian rupa untuk mengendalikan daya yang dihasilkan sehingga kecepatan pesawat atau kapal dapat berubah tanpa mengubah kecepatan putaran *propelernya*.



Gambar 2.3 Sudut *pitch* bilah turbin (Piggot, Hugh. 2001).

Pada turbin angin, pengaturan sudut *pitch* dilakukan untuk menyesuaikan daya yang dihasilkan terhadap kecepatan angin yang bervariasi.

2.2.3. Panjang *Chord* dan Dimensi *Taper*

Panjang *chord* pada turbin angin tidak memiliki pengaruh terlalu besar pada performa turbin angin. Rotor dengan bilah yang memiliki panjang *chord* seragam di sepanjang *span*-nya, masih bisa berputar dengan rugi efisiensi yang kecil. Namun terdapat sebab lain yang membuat penentuan panjang *chord* pada bilah harus dipertimbangkan. Bagian pangkal bilah merupakan daerah dengan tangkapan angin yang kecil, sedangkan bagian ujung bilah merupakan daerah penghasil torsi yang paling besar. Dengan membuat bagian pangkal lebih lebar daripada bagian ujung, maka turbin akan lebih mudah untuk mulai berputar pada kecepatan angin yang rendah (Piggot, Hugh. 2001).

Bagian ujung merupakan daerah penghasil torsi paling besar, sehingga dimensi *chord* harus diperhitungkan dengan cermat. Untuk mendekati batasan Betz, maka persamaan berikut dapat digunakan sebagai pendekatan.

$$C = \frac{16\mu R (R/r)}{9 \lambda^2 B} \quad (2.9)$$

Dimana :

C = Panjang *chord* (m)

R = Radius total rotor (m)

r = radius pada segmen *chord* yang dihitung (m)

λ = *tip speed ratio*

B = Jumlah bilah pada rotor

Bilah dengan panjang *chord* yang semakin kecil pada ujungnya (*taper*) memiliki keuntungan sebagai berikut (Piggot, Hugh. 2001):

- Efisiensi sedikit meningkat
- Ketahanan terhadap *stress* dan *fatigue* meningkat. Beban lengkung terbesar adalah pada bagian pangkal sehingga pangkal yang lebar lebih kuat menahan beban mekanis.
- Pangkal yang lebar membantu torsi awal putar

2.2.4. Bahan Bilah

Bahan untuk bilah turbin bervariasi dengan berbagai kelebihan dan kelemahan. Hal yang dipertimbangkan dalam pemilihan bahan adalah harga, kekuatan dan massa. Untuk skala kecil dengan harga yang murah, maka kayu balsa dapat menjadi pilihan. Untuk skala menengah dengan kekuatan yang lebih, bahan logam baja atau aluminium dapat dipakai apabila harga tidak diperhatikan. Untuk skala sangat besar dengan teknologi yang tinggi dan massa yang ringan, bahan komposit serat kaca atau serat karbon dapat digunakan. Tabel berikut menunjukkan perbandingan berbagai bahan bilah (Piggot, Hugh. 2001).

Tabel 2.1 Perbandingan bahan bilah turbin (Piggot, Hugh. 2001).

Bahan	Keterangan
Kayu	- Murah
	- Ringan
	- Mudah di fabrikasi dengan alat sederhana
Logam Baja	- Mahal
	- Berat
Aluminium	- Ringan
	- Mudah dibentuk
	- Tidak tahan <i>fatigue</i>
Fiber Carbon	- Sangat ringan
	- Kuat
	- Sangat Mahal
	- Harus di fabrikasi di pabrik dengan teknologi tinggi
Fiber Glass	- Ringan
	- Mudah dibuat dengan proses cetak
	- Lebih murah dibandingkan logam dan serat karbon

2.2.5. Jenis Airfoil

Turbin angin sederhana dengan skala kecil tidak terlalu memperhatikan jenis *airfoil* yang dipakai. Untuk turbin angin horizontal axis biasanya menggunakan bentuk bentuk foil yang sederhana mulai dari pelat datar atau *airfoil* tipis berstandar NACA

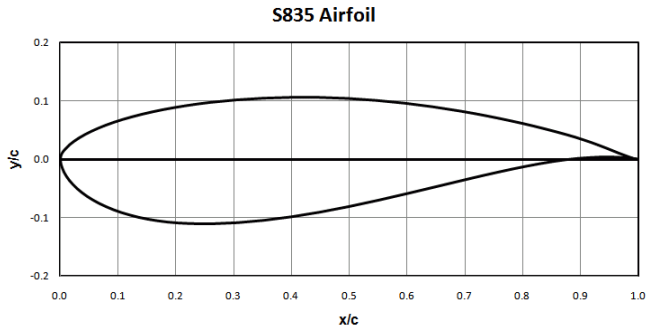
(National Advisory Committee for Aeronautics) seri 4 atau 5 digit. *Airfoil* sederhana tersebut bisa saja digunakan dengan rugi performa yang sedikit (Piggot, Hugh. 2001).

Badan laboratorium nasional NREL (*National Renewable Energy Laboratory*) telah melakukan penelitian dan merilis beberapa *airfoil* yang cocok untuk diterapkan pada turbin angin *horizontal axis* dengan skala kecil hingga besar. *Airfoil* tersebut diprediksi memiliki peningkatan energi sebesar 23% - 33% untuk *regulated* turbin, 8% - 20% untuk *variable pitch* turbin, dan 8% - 10% untuk *variable RPM* turbin (Buhl, M. 2009). Untuk turbin angin dengan diameter rotor 1 – 3 meter, NREL mengeluarkan seri S822 dan S823 yang kemudian digantikan oleh seri S83n (S833, S834, S835). *Airfoil* seri S83n merupakan *airfoil* tebal yang cocok digunakan untuk turbin angin dengan variabel speed dan variabel *pitch* dengan tipikal foil rendah noise dan koefisien *lift* yang tinggi (Buhl, M. 2009). Tabel 2.2 berikut menunjukkan spesifikasi *airfoil* NREL S83n.

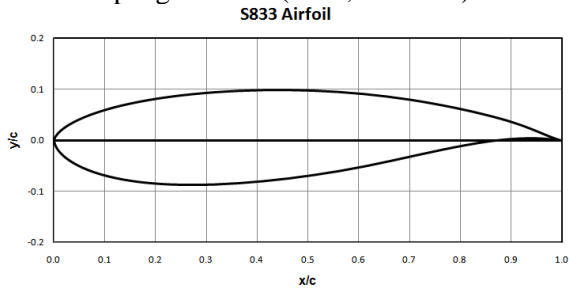
Tabel 2.2 Spesifikasi desain *airfoil* NREL S83n (Buhl, M. 2009)

	S833	S834	S835
Blade radial station	0,75	0,95	0,40
Parameter	<i>Objective / constraint</i>		
Reynolds number, R	$0,4 \times 10^6$	$0,4 \times 10^6$	$0,25 \times 10^6$
Max lift coef. $C_{l,max}$	1,10	1,00	1,20
Lower limit of low drag, lift coef. $C_{l,u}$	0,30	0,20	0,40
Upper limit of low drag, lift coef. $C_{l,u1}$	0,90	0,80	1,00
Zero lift <i>pitching</i> moment coef. $C_{m,0}$	$\geq -0,15$		
<i>Airfoil</i> thickness, t/c	18 %	15 %	21 %

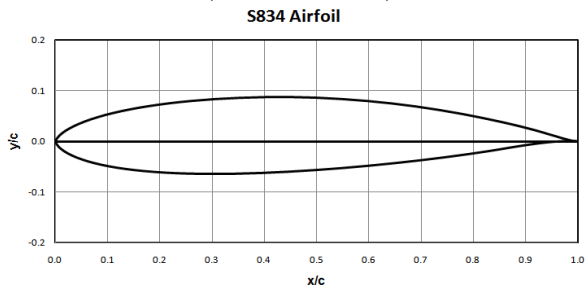
Berdasarkan desain spesifikasi tersebut, maka ketiga *airfoil* tersebut dapat digunakan pada turbin angin dengan diameter 2 m yang dilengkapi dengan variabel *pitch* control. Untuk bagian pangkal digunakan tipe S835, bagian tengah tipe S833, dan bagian ujung bilah tipe S834.



Gambar 2.4 Penampang *airfoil* NREL S835 untuk pangkal bilah (Buhl, M. 2009)



Gambar 2.5 Penampang *airfoil* NREL S833 untuk tengah bilah (Buhl, M. 2009)



Gambar 2.6 Penampang *airfoil* NREL S834 untuk ujung bilah (Buhl, M. 2009)

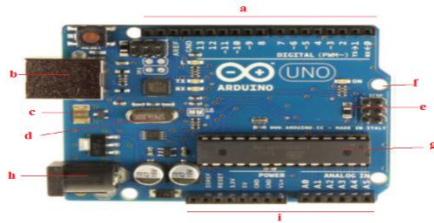
2.3 Pengaturan Sudut *Pitch* Bilah Turbin Angin

Untuk mengatur sudut *pitch* dari bilah turbin angin, maka diperlukan suatu mekanisme elektronik untuk memudahkan

pengaturan sudut sekaligus untuk mencatat kecepatan putar turbin angin dan kecepatan angin pada kecepatan angin tertentu. Pada tugas akhir ini digunakan Arduino sebagai mikroprosesor untuk mengintegrasikan sensor-sensor yang terdapat pada *prototype* turbin angin.

2.3.1 Arduino Uno

Arduino Uno adalah pengendali *mikro single-board* yang bersifat *open-source*, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Kata “Uno” pada arduino uno berasal dari bahasa Italia yang berarti satu dan keluaran produk arduino dinamai Arduino 1.0. Arduino Uno dan tipe 1.0 menjadi referensi dari produk arduino selanjutnya. *Hardware* nya memiliki prosesor Atmel AVR (*mikrokontroler* AVR 8 bit dan ARM 32 bit.) dan *software* nya memiliki bahasa pemrograman sendiri. Open source IDE yang digunakan untuk membuat aplikasi *mikrokontroler* yang berbasis *platform* arduino. *Mikrokontroler* sendiri adalah komputer berskala kecil yang terdiri atas *mikroprosesor*, memori, dan beberapa fitur terintegrasi dalam satu *integrated circuit* (IC). Salah satu keunggulan dari arduino adalah tidak perlu perangkat *chip programmer* karena di dalamnya sudah terdapat *bootloader* yang akan menangani *upload* program dari komputer dan sudah memiliki sarana komunikasi USB dengan suplai tegangan, sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya. Suplai eksternal selain USB juga dapat digunakan dengan menghubungkan sebuah adaptor AC ke DC. Dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan seperti pada gambar 2.7 berikut.



Gambar 2.7 Papan Arduino Uno

Fungsi dari masing-masing pin pada Arduino dapat dilihat pada tabel berikut.

Tabel 2.3 Alokasi pin Arduino uno

Kode	Keterangan	Fungsi
a	14 pin <i>input</i> / <i>output</i> digital (0-13)	sebagai <i>input</i> atau <i>output</i> , dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog <i>output</i> dimana tegangan <i>output</i> -nya dapat diatur. Nilai sebuah pin <i>output</i> analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V
b	USB	untuk memuat program dari komputer ke dalam papan, komunikasi serial antara papan dan komputer, memberi daya listrik kepada papan
c	Sambungan SV1	Sambungan atau jumper untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.
d	Q1 = Kristal (quartz crystal oscillator)	Jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

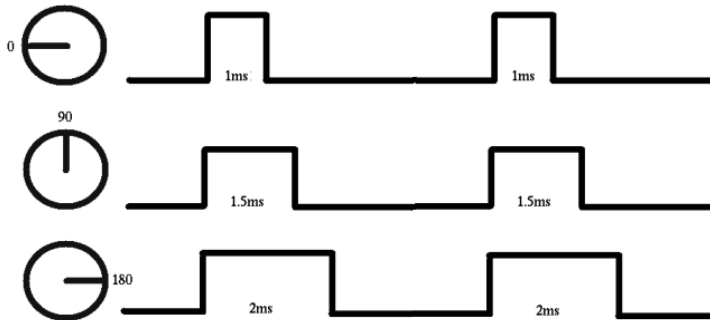
Tabel 2.3 Lanjutan

Kode	Keterangan	Fungsi
e	Tombol Reset S1	Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.
f	In = Circuit Serial Programming (ICSP)	Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.
g	IC 1 = Mikrokontroler Atmega	Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.
h	X1 = Sumber Daya External	Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan dc antara 9-12V.
i	6 Pin <i>Input</i> analog (0-5)	Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin <i>input</i> antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

2.3.2 Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah (*clockwise* (CW) / searah jarum jam dan *counter clockwise* (CCW) / berlawanan arah jarum jam) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Mode sinyal kendali motor servo adalah berdasarkan sinyal *step*

PWM. Dimana arah dan besar perubahan sudutnya bergantung pada *duty cycle* yang diberikan melalui PWM (Farid. 2011).



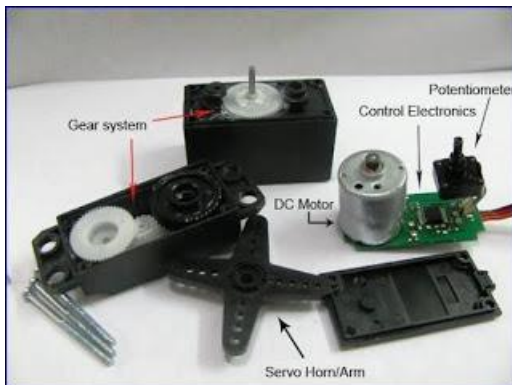
Gambar 2.8 Sinyal Kendali Motor Servo (Farid. 2011)

Karakteristik sinyal kendali motor servo standar (beroperasi pada sudut 0 – 180 derajat) adalah dimana bila diberikan pulsa sebesar 1,5 ms mencapai gerakan 90 derajat. Bila diberikan data kurang dari 1,5 ms maka posisi mendekati 0 derajat dan bila diberikan data lebih dari 1,5 ms maka posisi mendekati 180 derajat. Pensinyalan ini dapat dilihat pada gambar 2.8. Beberapa ketentuan pemberian sinyal kendali pada motor servo adalah sebagai berikut:

- Motor servo akan bekerja secara baik jika pada bagian pin sinyal diberikan sinyal PWM dengan frekuensi 50 Hz.
- Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *duty cycle* 1,5 ms, maka rotor dari motor akan berhenti tepat ditengah-tengah (sudut 90° / netral).
- Pada saat *duty cycle* dari sinyal yang diberikan kurang dari 1,5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) atau berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle*, dan akan bertahan diposisi tersebut
- Sebaliknya, jika *duty cycle* dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier

pula terhadap besarnya *duty cycle*, dan bertahan diposisi tersebut. (EG, 2010).

Motor servo dilengkapi dengan tiga kabel. Dua digunakan sebagai jalur catu daya positif dan ground. Kabel lain digunakan sebagai jalur sinyal kontrol. Kabel-kabel tersebut pada umumnya memiliki warna merah sebagai catu daya positif, kuning sebagai sinyal kontrol atau hitam sebagai ground. Tidak seperti motor DC, membalik koneksi kabel positif dan ground tidak akan membalik putaran motor servo/ hal ini akan merusak rangkaian feedback dari motor servo, oleh sebab itu sangat penting untuk memperhatikan pengkodean warna pada kabel motor.



Gambar 2.9 Bagian dalam motor servo (Farid. 2011)

Motor servo umumnya tersusun atas motor DC, sistem gear, sensor posisi (potensiometer) dan rangkaian kontrol. Motor DC terhubung dengan mekanisme *gear* yang memberikan umpan balik terhadap sensor posisi. Dari *gearbox*, *output* dari motor terhubung keluar dengan poros. Potensiometer mengubah posisinya berdasarkan putaran motor dan menghasilkan perubahan tegangan yang ekuivalen dengan sudutnya. Tegangan ini kemudian dibandingkan dengan sinyal kendali PWM yang masuk ke rangkaian kontrol. Dari perbedaan tersebut motor akan berputar sehingga posisi *error* adalah sama dengan nol, sehingga motor akan bertahan di posisi tersebut (Farid. 2011).

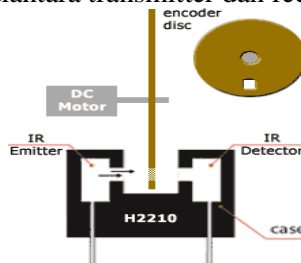
2.3.3 Rotary Encoder

Rotary encoder, atau bisa juga disebut dengan *shaft encoder* adalah suatu komponen elektro mekanis yang digunakan untuk mengubah posisi angular dari gerakan poros menjadi kode analog atau digital. Keluaran dari *encoder* dapat diolah lebih lanjut menjadi informasi kecepatan, jarak, RPM ataupun posisi. *Rotary encoder* digunakan pada aplikasi-aplikasi yang membutuhkan kepresisian dalam menentukan kondisi gerakan poros yang berputar terus menerus. Terdapat dua jenis *rotary encoder*, yaitu tipe *absolute* dan *incremental*. Kedua jenis *encoder* ini memiliki fisik yang berbeda. *Rotary encoder* tipe *absolute* memiliki dua atau lebih pasangan sensor dan dapat memberikan informasi berupa posisi sudut absolut dari suatu poros. Sedangkan *rotary encoder* tipe *incremental* hanya memiliki satu pasang sensor untuk mendeteksi jarak pergerakan relative dari suatu poros (Farid. 2011).

Sensor yang umum digunakan untuk mendeteksi perubahan putaran dari *rotary encoder* adalah sensor cahaya atau sensor magnetic. Apabila menggunakan sensor magnetic, maka titik-titik *bit* pada *encoder* diwakili dengan perbedaan warna hitam-putih atau lubang-lubang. Pada umumnya aplikasi robotic *rotary encoder* yang digunakan adalah yang bertipe *relative* dengan dikombinasikan dengan sensor *optocoupler* (*photointerrupter*).

2.3.4 Photointerrupter

Photointerrupter merupakan komponen *switching* yang berbasis cahaya. Komponen ini terdiri dari LED inframerah dan *phototransistor*. *Photointerrupter* berfungsi untuk mendeteksi keberadaan benda diantara transmitter dan receiver.

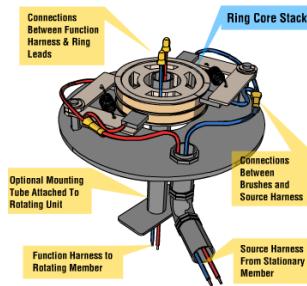


Gambar 2.10 Pemasangan *photointerrupter* (Farid. 2011)

Bila photointerrupter dikombinasikan dengan *rotary encoder* relative, maka rangkaian ini dapat digunakan untuk mendeteksi putaran dari poros yang berputar. Saat poros berputar, maka celah-celah pada *encoder* akan bergerak dan *photointerrupter* akan menghasilkan pulsa-pulsa kotak. Dengan menghitung jumlah pulsa kotak, maka akan diketahui kecepatan rotasi dari poros tersebut.

2.3.5 Slip Ring

Slip ring merupakan metode untuk membuat sambungan elektrik melalui benda yang berputar. *Slip ring* biasa disebut dengan *rotary connector* atau *rotary joints* juga bisa ditemukan pada motor elektrik, generator, alternator dan turbin angin.



Gambar 2.11 *Slip ring* (Farid. 2011)

Slip ring adalah hubungan berputar yang digunakan untuk menghantarkan arus listrik dari benda yang diam ke benda yang berputar, hal ini dapat dilakukan dengan cara menahan inti sambungan dengan *brush* yang berputar atau menahan brush dengan intik sambungan yang berputar.

2.4 Logika Fuzzy Tipe-2

Dalam Logika *Fuzzy*, basis pengetahuan yang digunakan untuk membangun aturan (*rules*) memiliki ketidakpastian yang dikarenakan perbedaan pengertian dalam mengartikan kata-kata dalam kaidah *fuzzy*, perbedaan dalam menentukan himpunan *consequent* setiap kaidah, serta adanya bising yang menyisipi data. Fungsi keanggotaan Logika *Fuzzy* tipe 1 tidak bisa secara langsung

menangani ketidakpastian *rule*. Fungsi keanggotaan *antecedent* atau *consequent* Logika Fuzzy tipe 2 mampu menangani ketidakpastian aturan. Konsep ketidakpastian dari *fuzzy* tipe 2 pertama kali dikenalkan oleh Zadeh, sebagai pengembang dari konsep ordinary *fuzzy set*. Tingkatan pada *fuzzy set* tipe 2 bisa berada pada *primary membership subset*, dan pada *secondary membership subset*. Sama dengan Logika Fuzzy tipe 1, Logika Fuzzy tipe 2 juga terdiri dari ; membership function, *fuzzy inference system* dan *defuzzifikasi*.

2.4.1 Logika Fuzzy

Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Inteligent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Algoritma ini digunakan dalam berbagai aplikasi pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Logika *fuzzy* menginterpretasikan statemen yang samar menjadi sebuah pengertian yang logis. Logika Fuzzy pertama kali diperkenalkan oleh Prof. Lotfi Zadeh seorang kebangsaan Iran yang menjadi guru besar di University of California di Berkeley pada tahun 1965 dalam papernya yang monumental. Dalam paper tersebut dipaparkan ide dasar *fuzzy set* yang meliputi *inclusion, union, intersection, complement, relation* dan *convexity*. Pelopor aplikasi *fuzzy set* dalam bidang kontrol, yang merupakan aplikasi pertama dan utama dari *fuzzy set* adalah Prof. Ebrahim Mamdani dan kawan-kawan dari Queen Mary College London. Penerapan kontrol *fuzzy* secara nyata di industri banyak dipelopori para ahli dari Jepang, misalnya Prof. Sugeno dari Tokyo Institute of Technology, Prof. Yamakawa dari Kyusu Institute of Technology, Togay dan Watanabe dari Bell Telephone Labs.

1. Himpunan Fuzzy

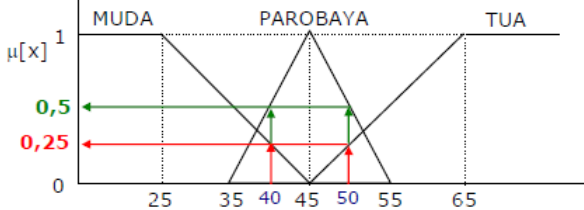
a. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperatur, permintaan, dsb.

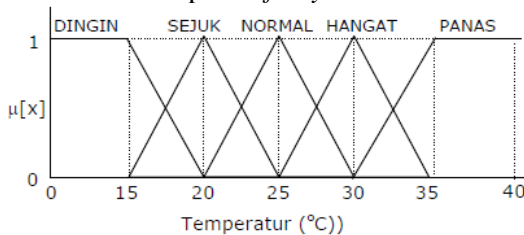
b. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Contoh:

- Variabel umur, terbagi menjadi 3 himpunan *fuzzy*, yaitu: MUDA, PAROBAYA, dan TUA.
- Variabel temperatur, terbagi menjadi 5 himpunan *fuzzy*, yaitu: DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS.



Gambar 2.12 Himpunan *fuzzy* untuk variabel umur



Gambar 2.13 Himpunan *fuzzy* pada variabel temperature

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh:

- Semesta pembicaraan untuk variabel umur: $[0 +\infty]$
- Semesta pembicaraan untuk variabel temperatur: $[0 40]$

d. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam

suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan *fuzzy*:

- MUDA = [0 45]
- PABOBAYA = [35 55]
- TUA = [45 +∞]
- DINGIN = [0 20]
- SEJUK = [15 25]
- NORMAL = [20 30]
- HANGAT = [25 35]
- PANAS = [30 40]

2. Operasi Himpunan *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama fire strength atau α -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu:

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \quad (2.10)$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y]) \quad (2.11)$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A^c} = 1 - \mu_A[x] \quad (2.12)$$

Fungsi dari logika *fuzzy* adalah sebagai berikut:

- a) *Fuzzifikasi* berfungsi untuk mentransformasi sinyal masukan yang bersifat *crisp* ke himpunan *fuzzy* dengan menggunakan operator fuzzifikasi.
- b) Basis pengetahuan berisi basis data dan aturan dasar yang mendefinisikan himpunan *fuzzy* atas daerah-daerah masukan dan keluaran yang disusun dalam perangkat aturan kontrol.
- c) Logika pengambilan keputusan merupakan inti dari logika *fuzzy* yang memiliki kemampuan seperti manusia dalam pengambilan keputusan. Aksi pengaturan *fuzzy* disimpulkan dengan menggunakan implikasi *fuzzy* dan mekanisme interferensi *fuzzy*.
- d) *Defuzzifikasi* berfungsi untuk mentransformasi kesimpulan tentang aksi atur yang bersifat *fuzzy* menjadi sinyal sebenarnya yang bersifat *crisp* dengan menggunakan operator *defuzzifikasi*.

2.4.2 Interval Logika Fuzzy Tipe-2

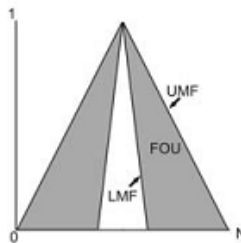
Logika *fuzzy* tipe 2 merupakan pengembangan dari Logika *fuzzy* tipe 1. Logika *fuzzy* tipe 2 memiliki fungsi keanggotaan interval ; *membership function*, *Fuzzy Inference System* dan *Defuzzifikasi*.

A. Membership Function

Logika *fuzzy* Interval tipe 2 memiliki kesamaan dengan logika *fuzzy* tipe 1 yaitu proses *fuzzifier*, *rule base*, *interference engine* dan *output processor*. *Output processor* meliputi tipe *reducer* dan *defuzzifier* menghasilkan suatu *output fuzzy set tipe 1* atau sebuah

bilangan. Namun perbedaannya terletak dalam proses pencarian *centroid*, pada interval tipe 2 *fuzzy set* dilakukan dengan dengan *Upper Membership Function* (UMF) dan *Lower Membership Function* (LMF).

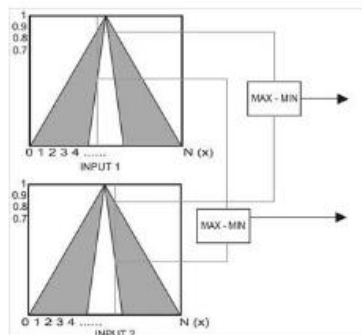
Fungsi keanggotaan dalam interval tipe 2 *fuzzy logic* diatur sebagai daerah yang disebut dengan *Footprint Uncertainty* dimana dibatasi dengan 2 tipe 1 fungsi keanggotaan *fuzzy logic* UMF dan LMF. Fungsi keanggotaan interval tipe 2 *fuzzy logic* dapat dilihat pada gambar 5 berikut ini :



Gambar 2.14 *Footprint of Certainty* (Aftika, 2012)

B. Operasi pada *membership function* tipe-2

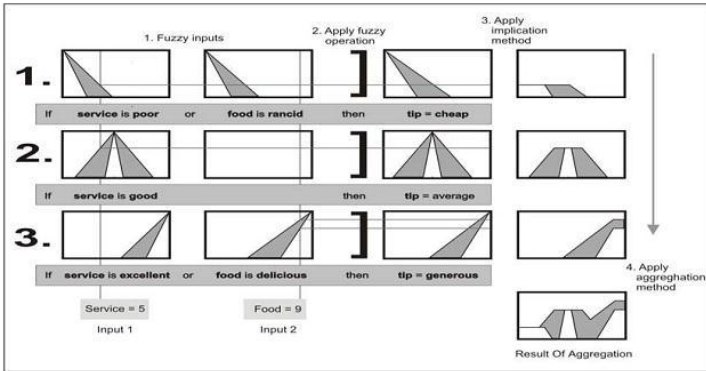
Operasi pada interval *fuzzy* tipe 2 hampir sama dengan *fuzzy* tipe 1, hanya saja pada *fuzzy* tipe 2, operasi dilakukan pada dua interval, atas (UMF) dan bawah (LMF) sekaligus. Operasi pada *membership function* tipe 2 dapat dilihat pada Gambar 2.15 sebagai berikut :



Gambar 2.15 Operasi pada *membership function fuzzy* tipe 2 (Aftika, 2012)

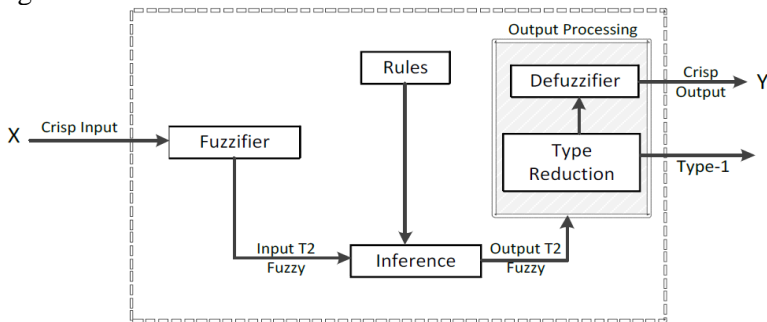
C. Fuzzy Inference System

Fuzzy inference system pada tipe-2 hampir sama dengan fuzzy inference system pada tipe-1, dengan menggunakan tahapan yang sama. Operasi Fuzzy inference system dapat dilihat pada gambar sebagai berikut.



Gambar 2.16. Fuzzy inference system pada tipe-2 (Aftika, 2012)

Diagram blok pada logika fuzzy tipe 2 hampir sama dengan logika fuzzy biasa atau tipe 1, Perbedaan logika fuzzy tipe 2 dengan logika fuzzy tipe 1 hanya pada bagian proses keluaran. Pada logika fuzzy tipe 2 dari himpunan fuzzy melalui dua tahapan proses sebelum menjadi himpunan tegas. Proses pengurangan tipe untuk merubah keluaran fuzzy tipe-2 menjadi tipe-1 kemudian dilanjutkan ke proses defuzzifikasi untuk mendapatkan himpunan tegas.



Gambar 2.17 Diagram blok sistem fuzzy tipe 2 (Aria, 2011)

Sebelum mendapatkan hasil akhir berupa nilai *crisp*, ada dua tahapan yang dilakukan yaitu *type reduction* (tipe reduksi), merupakan langkah pertama pengolahan *output* dengan menghitung centroid dari logika *fuzzy* tipe 2 dan Defuzzifikasi, untuk memperoleh nilai *output* (*crisp*). Tipe reduksi dapat diselesaikan dengan beberapa metode antara lain:

- Reduksi tipe *centroid*(*center-of-sums*)

Centroid dihitung dari kombinasi area *output* yang sebelumnya didapatkan dari aktivasi setiap kaidah berdasarkan *firing strength*.

- Reduksi tipe pusat himpunan(*center-of-sets*)

Metode ini menghitung *centroid* setiap konsekuen dari kaidah yang aktif .

- Reduksi tipe ketinggian (*height*)

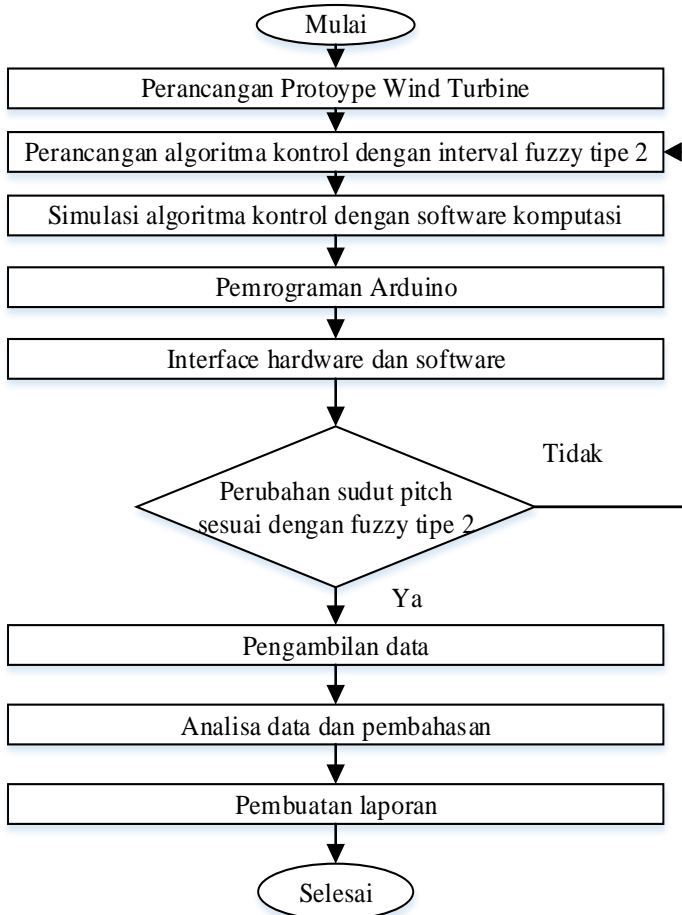
Keluaran setiap kaidah digantikan dengan *singleton* pada titik maksimum keanggotaan lalu dihitung *centroid* nya.

Halaman ini sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

3.1 Prosedur Umum Penelitian

Pada sub bab ini menjelaskan terkait prosedur penelitian tugas akhir yang dilakukan guna mencapai tujuan penelitian. Berikut ini diagram alir penelitian :



Gambar 3.1 Diagram alir penelitian

Dari diagram alir tersebut, penjelasan proses pengerjaan penelitian ini adalah sebagai berikut :

3.1.1 Perancangan *Prototype* Turbin Angin

Perancangan *prototype* turbin angin merupakan pembuatan turbin angin dengan diameter bilah sepanjang 2 meter dan memiliki 3 bilah, dengan tipe turbin angin *Horizontal Axis Wind Turbine* (HAWT). Turbin angin memiliki berat sekitar 990 gram untuk tiap *blade*, dengan bahan *blade* terbuat dari *fiber glass*. Turbin angin dilengkapi dengan sensor kecepatan yang terhubung dengan Arduino sebagai indikator kecepatan turbin angin. Penggerak *aktuator blade* menggunakan motor servo dengan torsi maksimum 20 kg.cm

3.1.2 Perancangan Algoritma Kontrol Menggunakan *Interval Fuzzy Tipe 2*

Perancangan algoritma kontrol ini dibuat berdasarkan karakteristik turbin angin yang telah dibuat dan disusun dengan menggunakan *interval fuzzy* tipe 2. Logika *fuzzy* yang ada akan disusun dalam software komputasi Matlab.

3.1.3 Simulasi Algoritma Kontrol dengan Menggunakan Software Komputasi

Simulasi menggunakan software komputasi yaitu Matlab 2014a. Simulasi ini bertujuan untuk mendapatkan kesesuaian logika kontrol yang ada dengan karakteristik turbin angin yang sudah dibuat. Kesalahan pada *plant* dapat diminimalisir dengan adanya simulasi ini. Simulasi *fuzzy* pada Matlab dapat dilakukan dengan melihat hasil perancangan logika *fuzzy* pada *rule viewer*.

3.1.4 Pemrograman Arduino

Hasil dari perancangan logika *fuzzy* tipe 2 yang sudah disimulasikan di Matlab kemudian diprogram dengan bahasa C atau bahasa pemrograman arduino agar dapat diupload ke dalam arduino. Sensor-sensor dan *aktuator* yang digunakan dalam *plant* turbin angin juga diprogram dalam arduino.

3.1.5 *Interface Hardware dan Software*

Arduino yang telah disusun pemrogramannya kemudian disambungkan dengan *plant* turbin angin untuk dilakukan pengujian kontrol logika *fuzzy* tipe 2 terhadap turbin angin berupa respon perubahan sudut *pitch* turbin angin terhadap kecepatan angin. Bila pada *plant* turbin angin berhasil merubah sudut *pitch* sesuai dengan kontrol logika *fuzzy* tipe 2, maka akan dilakukan pengambilan data, bila *plant* turbin angin tidak berhasil merubah sudut *pitch*, maka kembali lagi pada perancangan algoritma kontrol menggunakan interval *fuzzy* tipe 2.

3.1.6 Pengambilan Data

Pengambilan data yaitu mendapatkan data dari *plant* dengan parameternya adalah besar *pitch* angle pada turbin berdasarkan kecepatan angin yang diterima turbin, putaran turbin yang dihasilkan dan respon *plant* terhadap kontrol yang diberikan. Respon tersebut berupa perubahan PPS (*pulse per second*) turbin angin dengan kontrol yang diberikan ketika berputar.

3.1.7 Analisa Data dan Pembahasan

Analisa data meliputi kecepatan angin yang diterima, putaran turbin angin yang dihasilkan dan respon *plant* terhadap kontrol yang diberikan. Selanjutnya pembahasan dari analisa yang telah dibuat.

3.2 *Prototype Turbin Angin*

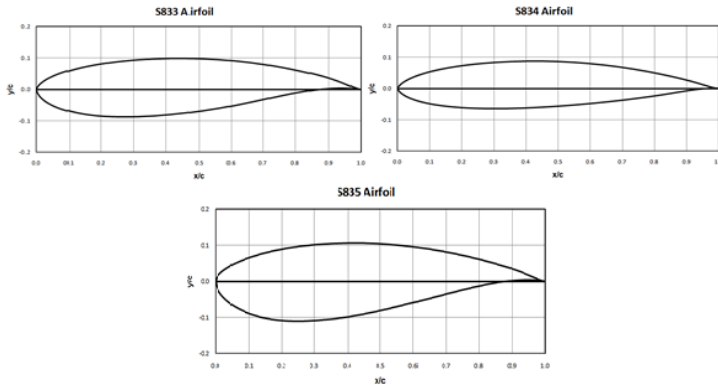
3.2.1 *Blade Turbin Angin*

Turbin angin yang dirancang merupakan turbin angin tipe HAWT dengan tiga buah *blade*. *Blade* yang dibuat menggunakan standar NREL S835 (bagian pangkal 25 cm), S833 (bagian tengah 35 cm), S834 (bagian ujung 40 cm). Adapun panjang tiap *blade* nya adalah 100 cm



Gambar 3.2 *Blade* turbin angin

Pembuatan *blade* dilakukan di laboratorium non metal Politeknik Perkapalan Negeri Surabaya (PPNS) dengan bahan dasar fiberglass dan standar NREL seperti pada gambar dibawah ini.



Gambar 3.3 Penampang *airfoil* NREL S833, S834 dan S835

Proses awal pembuatan *blade* / bilah, yang dibuat berjumlah 3 bilah, terdapat perbedaan berat antara 3 bilah tersebut sehingga perlu dilakukan penyeimbangan dengan cara menambahkan berat atau mendempul pada *blade* yang lebih ringan dengan bahan yang sama. Berikut tabel berat *blade* sebelum dan sesudah dilakukan penyeimbangan

Tabel 3.1 Berat penyeimbangan *blade* turbin angin

<i>Blade</i>	Berat Sebelum Penyeimbangan (gram)	Berat Setelah Penyeimbangan (gram)
A	940	984
B	910	987
C	990	990

3.2.2 *Aktuator*

Aktuator adalah alat peralatan mekanis untuk menggerakkan sistem atau *plant* yang digunakan. *Aktuator* yang digunakan pada

perancangan *prototype* ini menggunakan motor servo. Merek dan tipe motor servo yang digunakan adalah Hitec HS-805BB, motor servo ini dapat menggerakkan *blade* sampai sudut maksimum 180° . Motor servo ini dilengkapi dengan *gear box* dan dapat menghasilkan torsi 19,80 kg.cm ketika diberi tegangan 4,8 V dan torsi 24,70 kg.cm ketika diberi tegangan 6 V



Gambar 3.4 Motor servo Hitec HS-805BB

Tabel 3.2 Spesifikasi motor servo Hitec HS-805BB

Spesifikasi Motor Servo	
Merek / Tipe	Hitec HS-805BB
Ukuran	66 x 30 x 58mm
Berat	152g
Kecepatan	4.8V: 0.19 sec/60° 6.0V: 0.14 sec/60°
Torsi	4.8V: 275.00 oz-in (19.80 kgcm) 6.0V: 343.00 oz-in (24.70 kg-cm)

Setiap *blade* akan dipasang 1 motor servo untuk penggerakannya. Motor servo berfungsi sebagai untuk mengatur putaran sudut *blade* atau *pitch angle*. Sudut pada tiap motor servo perlu diuji antara *inputan* sudut yang diberikan dan sudut yang dihasilkan oleh motor servo. Adapun pengujian sudut motor servo seperti pada tabel 3.3 berikut.

Tabel 3.3 Pengujian Sudut Servo

Sudut Program	Sudut servo A	Sudut servo B	Sudut servo C
0	0	0	0
5	4	5	5
10	10	10	8
15	15	15.5	13
20	21	22	18
25	26	28	22
30	32	34	28
35	37	38	33
40	42	43	38
45	47	48	43
50	52	53	48
55	58	58	55
60	62	63	60
65	67	67	65
70	73	72	70
75	77	76	76
80	81	82	80
85	86	87	86
90	94	93	91

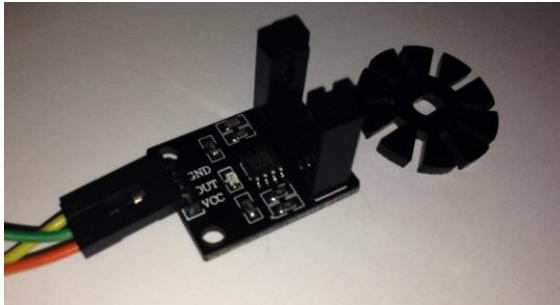
3.2.3 Sensor

Untuk mendapatkan kecepatan putar turbin angin, digunakan sensor berupa *rotary encoder* dengan 16 lubang sebagai pembangkit pulsa dan *optocoupler/photointerrupter* yang berfungsi untuk menghitung jumlah lubang. Cara kerja *optocoupler* adalah menghitung banyaknya pulsa yang dihasilkan dengan adanya lubang yang berputar. Efek gelap terang yang timbul dari piringan akan terbaca oleh *optocoupler* sebagai kondisi *high* dan *low*. Kondisi *high* saat *optocoupler* mengalami kondisi terang artinya tidak terhalang oleh piringan dan kondisi *low* terjadi saat *optocoupler* mengalami kondisi gelap atau saat terhalang piringan, pada kondisi *low* rangkaian sensor akan mengeluarkan

tegangan sebesar 0 V dan pada kondisi *high* rangkaian sensor akan mengeluarkan tegangan 4,5 Volt. Jumlah pulsa yang dikeluarkan rangkaian sensor selama satu detik atau *pulse per second* (PPS) akan diterima arduino dan dikonversi untuk menghitung banyaknya putaran tiap detik / *rotary per second* maupun banyaknya putaran tiap menit / *rotary per minute* seperti persamaan 3.1 dan 3.2 berikut.

$$RPS = \frac{PPS}{16} \quad (3.1)$$

$$RPM = \frac{PPS}{16} \times 60 \quad (3.2)$$



Gambar 3.5 Rotary encoder dan optocoupler

Pada tabel 3.4 merupakan pengujian dari sensor kecepatan dari sebuah motor DC 12 Volt dan arus 11 Ampere dengan beberapa variasi tegangan *supply* sehingga kecepatan motor akan bervariasi berbanding lurus dengan besar tegangan yang diberikan. Setiap tegangan *input* diambil tiga data untuk mengantisipasi data *error* akibat adanya faktor lain seperti tegangan dari DC power supply yang tidak stabil. Data yang terbaca pada sensor kecepatan akan dibandingkan dengan data hasil pembacaan alat ukur *Tachometer*.

Tabel 3.4 Pengujian sensor kecepatan

Tegangan Input Motor DC	Pembacaan Rotary Encoder (RPM)	Pembacaan Tachometer HT-3200 ONOSOKKI (RPM)
1.7 volt	390	390
	384	384
	396	396
2.5 Volt	560	560
	561	561
	588	588
3.7 Volt	1044	1059
	1049	1058
	1056	1048
6 Volt	1878	1830
	1832	1847
	1860	1830
8 Volt	2622	2684
	2664	2616
	2646	2674

Controller pada system ini menggunakan arduino uno. Arduino Uno adalah *board* atau sistem yang menggunakan ATmega328P sebagai otak yang mana mempunyai 14 *pin output/input* mempunyai *clock speed* 16 MHz. Berikut tabel spesifikasi dari arduino uno yang digunakan :

Tabel 3.5 Spesifikasi arduino uno

Spesifikasi Arduino Uno	
Microcontroller	: ATmega328P
Operating Voltage	: 5V
Input Voltage (recommended)	: 7-12V
Digital I/O	: Pins 14 (of which 6 provide PWM <i>output</i>)
PWM Digital I/O Pins	: 6

Tabel 3.5 Lanjutan

Spesifikasi Arduino Uno	
Analog <i>Input</i> Pins	: 6
DC Current per I/O Pin	: 20 mA
DC Current for 3.3V Pin	: 50 mA
Flash Memory	: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
Clock Speed	: 16 MHz
Length x Width	: 68.6 mm x 53.4 mm
Weight	: 25 g

3.2.4 Rotatial Connector atau Slip Ring

Rotatial connector atau *slip ring* diperlukan untuk mentransmisikan sinyal kendali untuk motor servo yang ada pada penopang *blade* yang berputar. Jalur yang diperlukan berjumlah lima buah yaitu untuk catudaya positif, *ground* dan tiga buah sinyal kendali. Dengan menggunakan *slip ring* kabel akan berputar didalam *shaft* sehingga kabel tidak akan mudah putus dan tidak akan ikut berputar.



Gambar 3.6 Slip Ring

Tabel 3.6 Spesifikasi slip ring

Spesifikasi Slip Ring	
Tegangan	240VAC/DC
Kecepatan kerja	0-250RPM
Arus	2A per circuit
Suhu Operasi	-20~60°C

Secara garis keseluruhan spesifikasi *prototype* turbin angin yang telah dibuat sebagai berikut :

Tabel 3.7 Spesifikasi *prototype* turbin angin

SPEKIFIKASI	
<i>Blade</i>	
Panjang	: 100 cm
Tipe	: NREL N83N
Bahan	: Fiberglass
Berat	: 0,99 gram
Motor Servo	
Tipe	: Hitec HS-805BB
Dimensi	: 66 x 30 x 58mm
Vin	: 4.8V: 0.19 sec/60° : 6.0V: 0.14 sec/60°
Torsi	: 4.8V: 275.00 oz-in (19.80 kgcm) : 6.0V: 343.00 oz-in (24.70 kg-cm)
Sensor	
Tipe	Rotary Encoder dan Photointerruptor
Rotational Connector	
Jumlah	: 1 buah
Tipe	: 6 Wires 240V 250RPM Conductive
Rotor Shaft	
Bahan	: Stainless Steel
Dimensi	: d = 1,5 cm, t = 0,2 cm
Penopang Blade	
Bahan	: PVC
Dimensi	: d = 28 cm, t = 1,5 cm
Main Plate	
Bahan	: Plat besi
Dimensi	: d = 40 cm, t = 0,2 cm
Tiang Penyangga	
Bahan	: Plat besi
Dimensi	: d = 8 cm, t = 0,2 cm



Gambar 3.7 *Prototype* turbin angin keseluruhan

3.3 Pengambilan Data Awal

Pengambilan data dilakukan dengan cara memberikan angin yang bervariasi pada turbin angin dan merubah sudut *pitch* secara manual. Pada pengambilan data awal, alat yang digunakan yaitu 3 buah blower yang mana setiap blower diatur untuk kecepatan yang sama. Ketiga blower tersebut mempunyai tiga buah variasi kecepatan yaitu lambat (*low*), sedang (*medium*) dan cepat (*high*) dimana ketiga blower tersebut memiliki spesifikasi kecepatan angin yang sama pada pada tiap variasinya. Pengambilan data dilakukan di Laboratorium Fisis Teknik Fisika. Data-data yang diambil berupa sudut *pitch*, kecepatan sudut atau kecepatan putar dan kecepatan angin. Data tersebut diambil dari nilai rata-rata saat mencapai kondisi *steady* maksimalnya kurang lebih 600 detik.



Gambar 3.8 Tiga buah blower

Pengambilan data dilakukan dengan variasi kenaikan sudut yang diberikan pada servo sebesar 5° dimulai dari 0° sampai 90° . Pengambilan data pada awal pengujian dilakukan dengan jarak yang bervariasi terhadap kecepatan putar turbin angin. Jarak yang bervariasi tersebut berpengaruh terhadap kecepatan angin yang dihasilkan oleh blower dan kecepatan putar turbin angin.

Tabel 3.8 Data pengaruh kecepatan angin, sudut *blade* dan jarak terhadap kecepatan putar turbin angin

Sudut <i>Pitch</i>	RPM					
	V ₁ = 2 m/s	V ₂ = 2,9 m/s	V ₃ = 3,5 m/s	V ₄ = 3,9 m/s	V ₅ = 4,5 m/s	V ₆ = 5 m/s
	Jarak = 300 cm	Jarak = 250 cm	Jarak = 250 cm	Jarak = 180 cm	Jarak = 130 cm	Jarak = 100 cm
0	13	0	27	32	18	30
5	97	27	32	18	30	27
10	79	40	40	33	48	42
15	53	44	45	48	47	42
20	45	38	41	44	42	40
25	36	34	36	39	39	36
30	34	30	32	35	35	33
35	31	29	30	33	31	30
40	29	26	27	30	29	27
45	25	24	25	27	28	25

Tabel 3.8 Lanjutan

Sudut <i>Pitch</i>	RPM					
	$V_1 =$ 2 m/s	$V_2 =$ 2,9 m/s	$V_3 =$ 3,5 m/s	$V_4 =$ 3,9 m/s	$V_5 =$ 4,5 m/s	$V_6 =$ 5 m/s
	Jarak = 300 cm	Jarak = 250 cm	Jarak = 250 cm	Jarak = 180 cm	Jarak = 130 cm	Jarak = 100 cm
50	23	23	23	25	24	23
55	21	21	22	24	23	21
60	19	19	21	22	20	20
65	17	17	18	20	18	19
70	14	15	15	18	16	18
75	12	13	14	16	15	15
80	11	10	11	15	14	13
85	9	10	10	12	12	13
90	0	8	7	10	10	11

Dari tabel 3.8 dapat diketahui bahwa putaran turbin angin maksimum berada pada saat jarak antara turbin angin dengan blower sejauh 300 cm atau 3 meter yang berarti bahwa putaran turbin akan meningkat bila jarak sumber kecepatan angin dengan turbin angin semakin menjauh. Hal ini ditunjukkan dengan putaran turbin angin yang mencapai nilai 97 RPM. Pada saat sumber kecepatan angin yaitu blower berjarak 100 hingga 250 cm, putaran turbin angin tidak dapat mencapai batas maksimumnya. Hal ini ditunjukkan dengan nilai putaran turbin yang tidak mencapai atau tidak melebihi dari nilai 50 RPM. Salah satu faktornya disebabkan kurang meratanya distribusi angin terhadap panjang dan lebar pada *pitch* turbin angin, sehingga hanya sebagian kecil atau sebagian besar bilah turbin angin yang terkena angin, meskipun dengan kecepatan angin yang lebih cepat yang dapat dilihat dari kecepatan V_2 hingga V_6 memiliki nilai kecepatan angin lebih besar dari pada V_1 , akan tetapi putaran turbin angin pada kecepatan V_1 lebih besar. Maka hasil pengujian pada tabel 3.8 pengaruh kecepatan angin terhadap kecepatan putar turbin angin kurang dapat dianalisa karena hasilnya yang kurang optimal. Sehingga dilakukan

pengujian kedua yaitu dengan jarak yang tetap atau konstan agar performansi turbin angin dapat dianalisa secara optimal.

Jarak yang digunakan yaitu antara blower dengan turbin angin sejauh 3 meter, hal ini dilakukan agar angin yang dihasilkan dapat mengenai seluruh panjang *blade*, Sedangkan untuk kecepatan anginnya dibuat beberapa variasi kecepatan angin dari pengaturan yang tersedia pada blower yaitu *low*, *medium* dan *high*. Berikut tabel 3.9 mode pengujian turbin angin dengan kecepatan angin yang bervariasi dari pengaturan kipas angin dengan jarak 300 cm.

Tabel 3.9 Mode pengujian turbin angin dengan kecepatan angin variatif

Mode	Kipas 1	Kipas 2	Kipas 3	Kecepatan Angin
1	High	High	High	2,6 m/s
2	Medium	Medium	Medium	1,9 m/s
3	Low	Low	Low	1,6 m/s
4	High	Low	High	2,3 m/s
5	Medium	Low	Medium	1,8 m/s
6	Low	High	Low	1,7 m/s

Berdasarkan tabel 3.9, kecepatan angin tertinggi yaitu 2,6 m/s dan kecepatan angin terendah yaitu 1,7 m/s. Kecepatan putar turbin diambil lebih rinci dari sebelumnya yaitu berupa *pulse per second* dan variabel sudut *pitch* yang digunakan juga lebih rinci dari sebelumnya yaitu dengan kelipatan 2° dimulai dari 0° hingga 80° . Berikut tabel 3.10 data kecepatan putar turbin angin yang dihasilkan dari berbagai kecepatan angin dan sudut *pitch* yang digunakan.

Tabel 3.10 Data pengaruh kecepatan angin, sudut *blade* dan kecepatan putar turbin (PPS) dengan jarak 3 meter

Sudut Pitch	Pulse per Second (PPS)					
	V ₁ = 2,6 m/s	V ₂ = 1,9 m/s	V ₃ = 1,6 m/s	V ₄ = 2,3 m/s	V ₅ = 1,8 m/s	V ₆ = 1,7 m/s
0	4	3	3	3	4	3
2	7	6	5	5	4	4

Tabel 3.10 lanjutan

Sudut Pitch	<i>Pulse per Second (PPS)</i>					
	V ₁ = 2,6 m/s	V ₂ = 1,9 m/s	V ₃ = 1,6 m/s	V ₄ = 2,3 m/s	V ₅ = 1,8 m/s	V ₆ = 1,7 m/s
4	23	23	22	23	23	24
6	25	22	21	25	19	22
8	21	21	19	22	21	21
10	18	20	17	19	19	19
12	15	15	16	16	15	16
14	15	15	13	15	14	14
16	13	15	13	14	13	13
18	12	15	12	12	12	12
20	11	15	10	12	11	11
22	10	11	9	10	11	10
24	10	10	8	10	10	9
26	9	10	8	9	9	9
28	9	9	8	8	9	9
30	8	8	6	8	8	8
32	8	8	6	8	8	8
34	8	7	6	7	7	8
36	7	7	6	7	7	7
38	7	7	5	7	7	6
40	7	6	5	7	7	6
42	7	6	5	6	6	6
44	6	6	4	6	6	6
46	6	6	5	5	5	5
48	6	5	5	5	5	5
50	6	5	5	5	5	5
52	5	5	5	5	5	5

Tabel 3.10 lanjutan

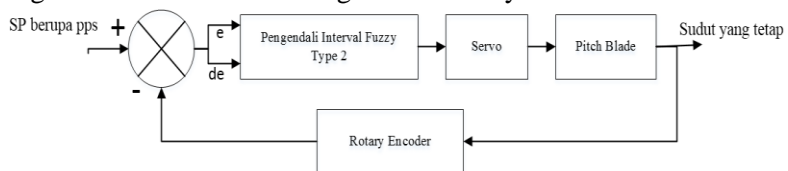
Sudut Pitch	Pulse per Second (PPS)					
	V ₁ = 2,6 m/s	V ₂ = 1,9 m/s	V ₃ = 1,6 m/s	V ₄ = 2,3 m/s	V ₅ = 1,8 m/s	V ₆ = 1,7 m/s
54	5	5	4	5	5	5
56	5	5	4	5	5	5
58	5	4	4	4	5	4
60	5	4	4	4	4	4
62	5	4	4	4	4	4
64	5	4	4	4	4	4
66	4	4	4	4	4	4
68	4	3	3	4	4	3
70	4	3	3	4	3	3
72	4	4	3	3	3	3
74	3	3	3	3	3	3
76	3	3	0	2	3	2
78	2	2	0	2	3	2
80	1	0	0	1	2	0

Berdasarkan data yang didapat yaitu pada tabel 3.10, turbin angin memiliki kecepatan putar maksimum sebesar 25 pps saat diberi kecepatan angin 2,6 m/s dan juga pada kecepatan angin 2,3 m/s. Sedangkan pada kecepatan angin 1,9 m/s didapat kecepatan putar maksimumnya adalah 23 pps, saat kecepatan angin 1,6 m/s, kecepatan putar maksimumnya 22 pps, saat kecepatan angin 1,8 m/s kecepatan putar maksimumnya 23 pps dan saat kecepatan angin 1,7 m/s kecepatan putar maksimumnya 24 pps. Dari enam variasi kecepatan yang diberikan terlihat bahwa empat diantaranya mencapai kecepatan putaran maksimumnya pada sudut 4° (kecepatan 1,9 m/s, 1,6 m/s, 1,8 m/s dan 1,7 m/s) dan sudut 6° saat kecepatan 2,6 m/s dan 2,3 m/s. Maka dapat diketahui bahwa turbin

angin mengalami peningkatan kecepatan putar turbin dari sudut 0° hingga 6° , dengan titik puncaknya berada pada range 4° hingga 6° .

3.4 Perancangan Algoritma Pengendalian Interval Fuzzy Tipe 2

Sistem pengendalian sudut *pitch* pada turbin angin merupakan sistem pengendalian loop tertutup. *Input* pengendaliannya berupa *error* dan *delta error*. *Error* merupakan selisih antara *setpoint* yang diberikan dengan variabel kontrol (putaran turbin yang terukur /pps). *Delta error* merupakan selisih antara *error* yang sekarang dengan *error* yang sebelumnya. Kedua *input*an ini akan digunakan sebagai referensi dalam merancang algoritma pengendalian yang akan dibuat. Pengendali dibuat untuk memperkecil *error*, namun pada logika *fuzzy*, *input* minimalnya adalah dua, maka ditambahkan *delta error* untuk memenuhi kondisi tersebut. *Delta error* berfungsi untuk mengetahui kondisi kerja sistem pengendali, sedang dalam melakukan peningkatan atau penurunan kecepatan putar dan untuk mengendalikan sudut *pitch blade*, digunakan motor servo sebagai *aktuator*-nya.

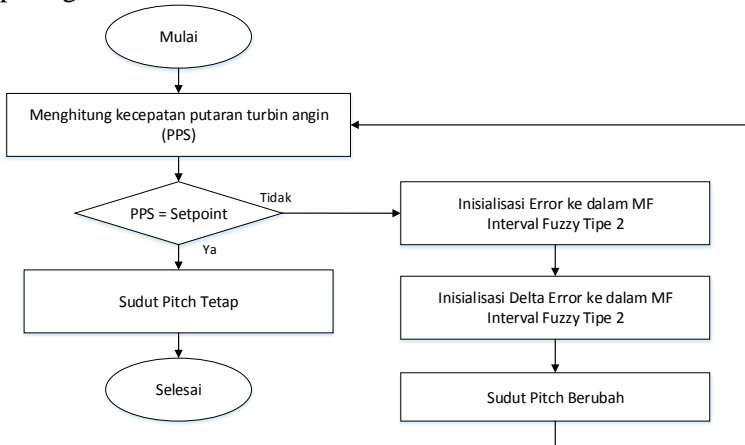


Gambar 3.9 Diagram blok sistem pengendalian

Setpoint pada pengendalian adalah kecepatan yang terukur yaitu berupa *pulse per second* (pps), lalu *input* pengendalian pada diagram blok tersebut adalah *error* dan *delta error*. Sedangkan *output*nya adalah perubahan sudut pada turbin angin dengan motor servo sebagai penggeraknya. *Error* didapatkan dari selisih *setpoint* dengan kecepatan yang terukur, dimana kecepatan yang terukur didapat dari *rotary encoder*. *Delta error* didapatkan dari selisih antara nilai *error* yang terbaca dengan nilai *error* yang sebelumnya. *Input error* dan *delta error* kemudian diproses

dengan pengendali interval logika *fuzzy* tipe 2. Hasil pengolahan dari interval logika *fuzzy* tipe 2 akan dilanjutkan ke motor servo, untuk merubah sudut pada *pitch* turbin angin. *Pitch* turbin akan berubah dan turbin angin akan berputar, yang mana bila putaran turbin angin sudah sesuai dengan pengaturan interval logika *fuzzy* tipe 2 yang telah dibuat, maka sudut *pitch* akan tetap, akan tetapi bila sudut *pitch* tidak sesuai dengan pengaturan interval logika *fuzzy* tipe 2, maka akan dikembalikan pada *input error* dan *delta error* dimana putaran turbin angin akan dibaca oleh sensor *rotary encoder*. Kecepatan angin yang digunakan yaitu kecepatan angin maksimum sebesar 2,6 m/s.

Dari gambar 3.9 diatas, maka dapat disusun diagram alir dari proses perancangan algoritma pengendalian interval *fuzzy* tipe 2 pada gambar 3.10 berikut :



Gambar 3.10 Diagram alir algoritma interval tipe 2

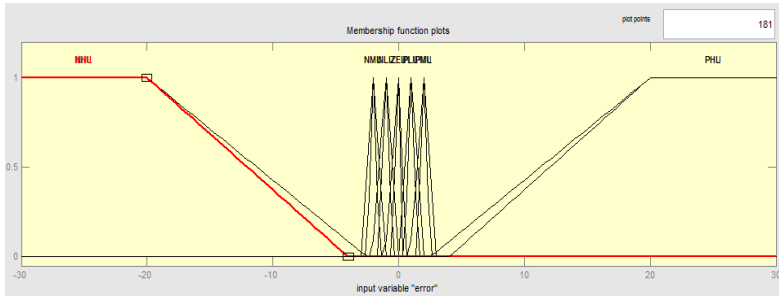
Langkah awal adalah inisialisasi kondisi *plant* dengan mendeteksi nilai *setpoint* dan nilai *pps*. Jika nilai *pps* sama besarnya dengan nilai *setpoint*, maka sudut tidak dirubah. Jika nilai *pps* berbeda dengan nilai *setpoint*, maka *error*nya dihitung dan mengklasifikasikannya kedalam fungsi keanggotaan *error* interval *fuzzy* tipe 2. Berikutnya langkah yang sama untuk nilai *delta error*

dengan cara menghitung *delta error* dan mengklasifikasikannya kedalam fungsi keanggotaan *error*. Kemudian setelah fungsi keanggotaan *error* dan *delta error* telah ditentukan maka dapat ditentukan aktuasi yang tepat yaitu berupa perubahan sudut *pitch* pada *blade*. Perubahan sudut dapat berupa penurunan atau peningkatan kecepatan sudut.

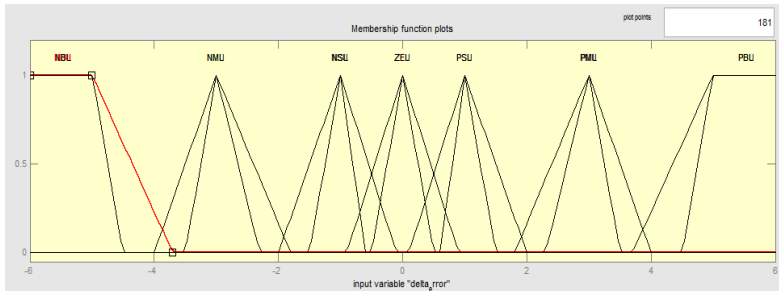
3.5 Perancangan Fungsi Keanggotaan Interval Fuzzy Tipe 2

Perancangan fungsi keanggotaan interval *fuzzy* tipe 2 hampir sama seperti *fuzzy* biasa, yaitu dengan 2 *input* yaitu *error* dan *delta error* dan 1 *output* berupa perubahan sudut *pitch blade* turbin angin, namun ada penambahan *lower membership function* dan *upper membership function*. Satu fungsi keanggotaan pada interval tipe 2 memiliki 2 fungsi keanggotaan *fuzzy* tipe 1 yaitu berupa UMF dan LMF. Perancangan tersebut dilakukan dengan beberapa tahapan dan dibentuk berdasarkan penelitian yang dilakukan berulang-ulang hingga mencapai nilai yang optimal. Rentang minimum dan maksimum fungsi keanggotaan *error* yang digunakan adalah -30 hingga 30. Nilai ini diambil dari kecepatan sudut maksimum yaitu 25 pps. Sedangkan rentang fungsi keanggotaan *delta error* didapatkan berdasarkan penelitian. Berdasarkan penelitian diketahui bahwa nilainya tidak terlalu besar yaitu dibawah 10 pps. Maka untuk mengantisipasi terjadi lonjakan, maka digunakan rentang -20 hingga 20. Metode *fuzzy* yang digunakan adalah metode takagi sugeno karena metode ini sudah cukup untuk kontrol kendali sudut turbin angin.

Adapun fungsi keanggotaan *input error* dan *delta error* yang telah dirancang pada software komputasi Matlab seperti pada gambar 3.11 dan gambar 3.12 berikut ini.



Gambar 3.11 Grafik fungsi keanggotaan *input error*



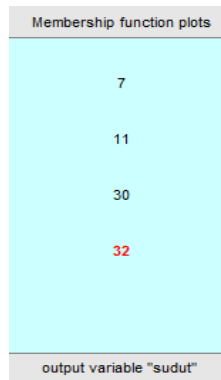
Gambar 3.12 Grafik fungsi keanggotaan *input delta error*

Gambar 3.11 dan gambar 3.12 merupakan grafik fungsi keanggotaan *input error* dan *delta error* yang memiliki bentuk segitiga dan bentuk trapesium. Fungsi keanggotaan ini menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya yang memiliki interval 0 sampai 1. Fungsi keanggotaan yang berbentuk trapesium pada umumnya diletakkan pada bagian ujung dari keseluruhan seri fungsi keanggotaan yang dibuat, karena dapat mempresentasikan nilai derajat keanggotaan satu mutlak pada bagian ujung tersebut, sedangkan fungsi keanggotaan yang berbentuk segitiga merupakan gabungan antara dua garis lurus yang linear, dimana nilai pada kedua ujungnya mempunyai derajat keanggotaan nol dan pada bagian tengah atau puncak mempunyai derajat keanggotaan satu.

Fungsi keanggotaan *input error* yang dibuat dengan range -90 hingga 90 sedangkan fungsi keanggotaan *input delta error*

dibuat dengan *range* -10,5 hingga 10,5. Nilai ini didapat dari hasil pengujian performa turbin angin dengan berbagai sudut yang telah dilakukan. Adapun *input error* dan *delta* memiliki 7 fungsi keanggotaan. Fungsi keanggotaan *input error* adalah Negatif High (NH), Negatif Medium (NM), Negatif Low (NL), Zero (ZE), Positif Low (PL), Positif Medium (PM) dan Positif High (PH). Sedangkan fungsi keanggotaan *delta error* adalah Negatif Big (NB), Negatif Medium (NM), Negatif Small (NS), Zero (ZE), Positif Small (PS), Positif Medium (PM), Positif Big (PB).

Masing-masing satu fungsi keanggotaan pada logika *fuzzy* tipe 2 memiliki 2 fungsi keanggotaan yaitu atas (*upper*) dan bawah (*lower*), sehingga *input error* dan *delta error* memiliki 14 fungsi keanggotaan. Adapun fungsi keanggotaan *output* dari kontrol logika *fuzzy* tipe 2 yang telah dirancang di Matlab dengan metode *fuzzy* yang digunakan yaitu metode Takagi-Sugeno seperti pada gambar 3.13 berikut :



Gambar 3.13 Fungsi keanggotaan *output*

Gambar 3.13 merupakan fungsi keanggotaan *output*. Nilai *output* pada metode ini berupa konstan yang tidak memperhatikan bentuk fungsi keanggotaan *input*, namun hanya memperhatikan batasan-batasan yang digunakan. Sistem perbandingan dua *input* yang digunakan adalah minimum, jika data *input* terletak pada bagian perpotongan, maka nilai yang diambil adalah nilai yang minimum. Nilai pada *output* yang digunakan

yaitu 7, 11, 30 dan 32. Adapun batas-batas pada setiap fungsi keanggotaan *input error* dan *delta error*, serta nilai konstanta pada fungsi keanggotaan *output* adalah sebagai berikut :

Batas <i>input error</i>	Batas <i>input delta error</i>
NHU [-88 -60 -20 -4]	NBU [-10.5 -6 -5 -3.7]
NHL [-90 -60 -20 -2.5]	NBL [-9 -6 -5 -4.5]
NMU [-3 -2 -1]	NMU [-4 -3 -1.8]
NML [-2.5 -2 -1.5]	NML [-3.5 -3 -2.3]
NLU [-2.1 -1 0]	NSU [-2 -1 -0.1]
NLL [-1.6 -1 -0.5]	NSL [-1.5 -1 -0.6]
ZEU [-0.9 0 0.1]	ZEU [-0.9 0 0.9]
ZEL [-0.6 0 0.6]	ZEL [-0.5 0 0.5]
PLU [0 1 2]	PSU [0.1 1 2]
PLL [0.5 1 1.5]	PSL [0.6 1 1.5]
PMU [0.9 2 3]	PMU [1.8 3 4]
PML [1.4 2 2.5]	PML [2.3 3 3.5]
PHU [2.5 20 60 90]	PBU [3.7 5 6 10.5]
PHL [4 20 60 90]	PBL [4.5 5 6 9]

Nilai *output* sudut

Nilai *output* sudut *pitch* adalah 7, 11, 30, 32.

Agar nilai *output* semakin teliti, maka batasan fungsi keanggotaan dibuat lebih sempit, yaitu dengan resolusi terkecil sebesar 2° pada servo. Selain itu dengan pengujian pada *plant* yang diulang-ulang, didapatkan hasil yang optimal pada batasan-batasan fungsi keanggotaan. Nilai *output* tersebut dipilih berdasarkan pengambilan data awal respon turbin angin. Saat rentang sudut *pitch* 0°-2° turbin angin berputar dengan kecepatan rendah, sedangkan pada sudut *pitch* 4°-6° turbin angin berputar pada kecepatan maksimumnya. Saat sudut *pitch* 8°-32°, kecepatan turbin angin mulai cenderung melambat, hal ini dapat digunakan sebagai acuan nilai *output* sudut untuk menurunkan kecepatan putaran dari turbin angin. Besarnya nilai sudut *pitch* berbanding lurus dengan nilai penurunan kecepatan sudut turbin angin. Hal ini dikarenakan sudut *pitch* yang semakin besar menyebabkan resultan gaya angkat

(*lift*) dan gaya geser (*drag*) menjadi tidak seimbang karena gaya angkat yang lebih besar.

3.6 Perancangan *Rule Base*

Rule base menunjukkan hubungan antara *input* – *output*. *Rule base* dibangun berdasarkan prinsip sistem pengendalian untuk mempertahankan variabel proses tetap berada dalam *setpoint*. Logika ini memiliki fungsi untuk memberikan hubungan antara nilai *error* dan *delta error* sebagai *input* dan nilai sudut *pitch* sebagai *output* yang mana sudut *pitch* dapat berubah merubah mengikuti dua *input* tersebut. Adapun perancangan *rule base* yang telah dibuat seperti pada tabel 3.11 sebagai berikut :

Tabel 3.11 Perancangan *rule base*

DE / E	NH	NM	NL	ZE	PL	PM	PH
NB	32	32	30	11	11	11	11
NM	32	32	30	11	7	7	7
NS	32	32	30	7	7	7	7
ZE	32	30	30	7	7	7	7
PS	32	30	30	7	7	7	7
PM	30	30	30	11	11	11	7
PB	30	11	11	11	11	11	11

$$\text{Error} = \text{setpoint} - \text{pps (variabel proses)} \quad 3.3$$

$$\text{Delta error} = e_n - e_{n-1} \quad 3.4$$

Dimana :

e_n = *error* yang terukur sekarang.

e_{n-1} = *error* yang terukur sebelumnya.

Pada *rule base* yang telah dibuat terdapat *error* dan *delta error* pada sistem pengendalian. *Error* menyatakan status variabel proses, yang merupakan selisih nilai antara *setpoint* yang diberikan dengan variabel yang terukur (pps) pada *plant* turbin. Sedangkan *delta error* menyatakan kondisi aktual *plant* turbin yang sedang berjalan. Berdasarkan persamaan *error* ini, maka status proses dapat diketahui. Status proses tersebut bisa bernilai negatif atau positif, dalam pengendalian ini berarti turbin angin butuh

dipercepat atau diperlambat. Status proses yang bernilai negatif, berarti kecepatan sudut turbin angin terlalu besar dibanding *setpoint*, maka turbin angin butuh diperlambat. Begitupun sebaliknya Status proses yang bernilai positif, berarti kecepatan sudut turbin angin terlalu kecil dibanding *setpoint*, maka turbin angin butuh dipercepat. Sedangkan *delta error* merupakan selisih antara nilai *error* yang sekarang dengan *error* yang sebelumnya. Nilai pada *delta error* ini menunjukkan keadaan yang terjadi pada turbin angin, sedang melakukan perlambatan atau peningkatan. Penjelasan tentang *error* dan *delta error* yang bertambah atau berkurang dapat dilihat pada tabel 3.12 berikut.

Tabel 3.12 Kondisi *error* dan *delta error*

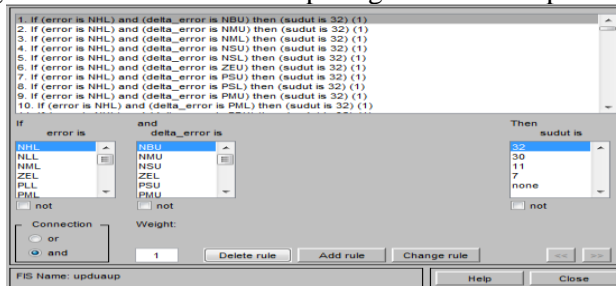
Error	Delta error	Keterangan
+	+	PPS turun dibawah <i>setpoint</i>
+	-	PPS sedang menaik menuju <i>setpoint</i>
-	+	PPS sedang menurun menuju <i>setpoint</i>
-	-	PPS naik diatas <i>setpoint</i>

Jika *error* bernilai positif (+) dan *delta error* bernilai positif (+), kondisi yang terjadi adalah pps menurun dibawah *setpoint* yang diberikan, yang berarti *plant* turbin angin sedang berputar melambat atau menjauhi dari *setpoint* yang diberikan, oleh karena itu perlu diberikan perubahan sudut *pitch* untuk meningkatkan kecepatan putar turbin. Jika *error* bernilai positif (+) dan *delta error* bernilai negatif (-), kondisi yang terjadi adalah pps sedang menaik menuju *setpoint*, yang berarti turbin angin sedang berputar dalam kondisi peningkatan kecepatan putarnya dan sedang menuju *setpoint*. Jika *error* bernilai negatif (-) dan *delta error* bernilai positif (+), kondisi yang terjadi adalah pps sedang menurun menuju *setpoint*, yang berarti kecepatan putar turbin angin melebihi atau melewati *setpoint* yang diberikan dan sedang menurunkan kecepatan putar turbin angin untuk mendekati *setpoint*. Jika *error* bernilai negatif (-) dan *delta error* bernilai negatif (-), kondisi yang terjadi adalah pps naik diatas *setpoint* yang diberikan, yang berarti turbin angin berputar melewati *setpoint* yang diberikan dan terus

bertambah kencang putarannya, maka diperlukan perubahan sudut *pitch* untuk menurunkan kecepatan putar turbin.

Pada tabel *rule base*, dapat diketahui jika *error* bernilai negatif *high* (NH) dan *delta error* bernilai negatif *big* (NB), maka kecepatan sudut lebih besar dari *setpoint*, maka perubahan yang dilakukan adalah dengan menurunkan kecepatan. Kecepatan yang menurun berbanding lurus dengan besarnya sudut *pitch* yang diubah. Jika *error* bernilai positif *high* (PH) dan *delta error* bernilai positif *big* (PB) berarti kecepatan putar turbin angin lebih kecil dari *setpoint*, maka aktuasi yang dilakukan adalah meningkatkan kecepatan. Besarnya peningkatan juga berbanding lurus dengan sudut *pitch* yaitu dengan sudut 7° , karena sudut 7 juga masih dapat digunakan untuk melakukan peningkatan kecepatan putar. Bila nilai *input* bernilai positif *high* (PH) dan negatif *small* (NS) maka peningkatan kecepatan sudut diperbesar hingga nilai variabel proses berada pada *setpoint*. Namun saat sampai *input delta error* negatif *medium* (NM) dan negatif *big* (NB), turbin sedang berada pada kondisi peningkatan kecepatan yang sangat besar sehingga dibutuhkan sedikit pengurangan kecepatan agar tidak terjadi *overshoot* yang terlalu besar.

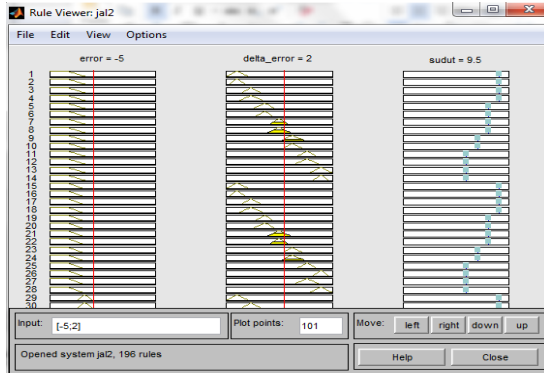
Dari data *rule base* tersebut, selanjutnya dapat dibuat pada *rule editor* dan disimulasikan secara rumus oleh *rule view* yang telah disediakan oleh Matlab 2014a. berikut tampilan *rule editor* pada gambar 3.14 dan *rule view* pada gambar 3.15 seperti berikut :



Gambar 3.14 Tampilan *rule editor* Matlab 2014a

Rule editor pada Matlab berfungsi untuk membuat perumusan dari fungsi keanggotaan *input* dan *output*. Perumusan yang dibuat

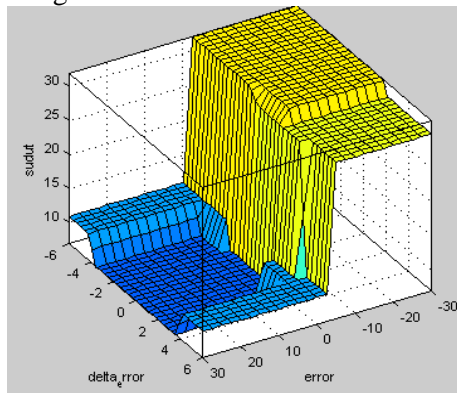
mencakup kebutuhan yang digunakan pada *plant*. *Rule editor* juga sebagai realisasi dalam perancangan *fuzzy* pada Matlab dari perancangan *rule base* yang telah dibuat.



Gambar 3.15 Tampilan *rule viewer* Matlab 2014a

Rule viewer berfungsi untuk melihat hasil perancangan editor *rule base* dari logika *fuzzy*. *Rule viewer* juga dapat disebut sebagai simulasi untuk logika *fuzzy* yang telah dibuat. Besaran nilai pada dua *inputan fuzzy* dapat diberikan secara manual, hingga nilai *output* akan keluar berupa angka.

Adapun untuk *surface view* dari *rule base* seperti pada gambar 3.16 sebagai berikut :



Gambar 3.16 *Surface view* pada Matlab 2014a

BAB IV ANALISIS DATA DAN PEMBAHASAN

4.1 Analisis Kinerja Turbin Angin

Pengujian kinerja turbin angin harus dilakukan sebelum penerapan algoritma kontrol sistem pengendalian turbin angin menggunakan interval *fuzzy* tipe 2. Beberapa data diambil untuk mengetahui kinerja turbin angin yang telah dibangun. Beberapa variabel dapat dijadikan acuan kinerja turbin angin adalah perubahan RPM turbin angin terhadap sudut *pitch* dengan variasi kecepatan angin, koefisien daya terhadap kecepatan angin yang bervariasi. Berikut merupakan tabel dari data RPM terhadap setiap variasi sudut *pitch* turbin angin untuk mencari RPM maksimum pada turbin angin.

Tabel 4.1 Data pengaruh kecepatan angin, sudut *blade* dan kecepatan putar turbin (RPM) dengan jarak 3 meter

Sudut <i>Pitch</i>	<i>Rotation per minute (RPM)</i>					
	V ₁ = 2,6 m/s	V ₂ = 1,9 m/s	V ₃ = 1,6 m/s	V ₄ = 2,3 m/s	V ₅ = 1,8 m/s	V ₆ = 1,7 m/s
0	15	12	12	10	17	11
2	25	22	20	18	16	14
4	86	86	83	87	88	91
6	94	84	78	92	71	84
8	79	77	71	82	79	78
10	68	74	63	72	73	71
12	58	58	60	60	58	60
14	55	54	51	56	53	54
16	50	57	47	51	48	50
18	44	55	44	47	45	46
20	43	54	39	44	43	42
22	39	40	34	39	39	38
24	37	39	32	37	37	36
26	35	36	31	34	35	35

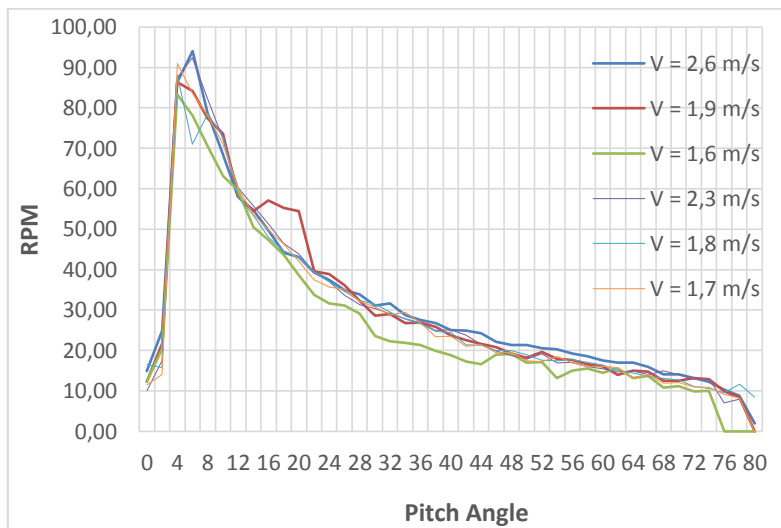
Tabel 4.1 lanjutan

Sudut Pitch	Rotation per minute (RPM)					
	V ₁ = 2,6 m/s	V ₂ = 1,9 m/s	V ₃ = 1,6 m/s	V ₄ = 2,3 m/s	V ₅ = 1,8 m/s	V ₆ = 1,7 m/s
28	34	32	29	31	32	32
30	31	29	24	30	31	31
32	32	29	22	29	29	29
34	29	27	22	28	28	29
36	28	27	21	27	27	27
38	27	26	20	25	25	23
40	25	24	19	25	24	24
42	25	23	17	24	21	21
44	24	22	17	21	21	21
46	22	21	19	20	20	19
48	21	19	19	19	20	20
50	21	18	17	18	19	17
52	21	20	17	19	18	17
54	20	18	13	17	18	19
56	19	18	15	17	18	17
58	19	17	16	16	17	16
60	18	16	15	16	16	16
62	17	14	16	15	15	16
64	17	15	13	15	14	13
66	16	15	14	14	14	14
68	14	13	11	15	13	12
70	14	13	11	14	13	12
72	13	13	10	13	11	11
74	12	13	10	13	11	11
76	10	10	0	7	10	9
78	9	8	0	8	12	8
80	2	0	0	2	8	0

Tabel 4.1 ini merupakan konversi dari nilai pps (*pulse per second*) yang dihasilkan dirubah menjadi RPM (*rotation per minute*) dan jarak antara sumber angin dan turbin angin sebesar 300 cm atau 3 meter. Berdasarkan data dari tabel 4.1 ini, maka dapat dianalisis RPM maksimum dan nilai *coefficient power* yang dihasilkan oleh turbin angin.

4.1.1 Nilai RPM Maksimum

Dari hasil data tabel 4.1, maka hubungan antara kecepatan maksimum dengan sudut *pitch* dapat dicari. Berikut grafik yang menunjukkan hubungan antara RPM dengan sudut *pitch*



Gambar 4.1 Hubungan sudut *pitch* turbin angin dengan kecepatan sudut

Dari gambar 4.1 tersebut, dapat diketahui bahwa turbin angin masih mampu berputar dengan sudut yang besar bahkan hingga sudut 80° meskipun kecepatan RPM nya semakin menurun, hal ini menunjukkan bahwa semakin besar sudut *pitch* pada turbin angin berbanding lurus dengan penurunan pada kecepatan sudut pada turbin angin. Faktor lain yang menyebabkan turbin masih mampu berputar dengan sudut yang besar adalah jarak antara

blower dengan turbin yang mencakup 3 meter, sehingga angin yang mengalir dapat mengenai seluruh panjang *blade*. Berdasarkan grafik, nilai RPM maksimum turbin angin didapat saat kondisi sudut *pitch* antara 4°-6° dengan nilai RPM tertinggi adalah 94,00.

4.1.2 Koefisien Daya / *Coefficient Power*

Koefisien daya dapat diperoleh melalui perhitungan yang berhubungan dengan data kecepatan angin. Koefisien daya pada turbin angin didapat dari kecepatan angin sebelum melewati turbin dan kecepatan angin setelah melewati turbin. Koefisien daya atau *Coeffisient Power* adalah perbandingan antara daya yang dihasilkan secara mekanik pada sudu akrobat gaya angin terhadap daya yang dihasilkan oleh gaya *lift* pada aliran udara. Dengan mengambil kecepatan putar maksimum pada tiap variasi kecepatan angin, maka diambil kecepatan angin setelah melewati turbin dan didapatkan nilai koefisien daya. Pengukuran kecepatan angin ditentukan pada jarak dua meter sebelum dan sesudah *blade* pada turbin.

$$C_p = \frac{P}{P_o} = \frac{\frac{1}{4} \rho_{udara} \cdot A \cdot (v_1^2 - v_2^2) \cdot (v_1 + v_2)}{\frac{1}{2} \cdot \rho_{udara} \cdot A \cdot v_1^3} \quad 4.1$$

$$C_p = \frac{P}{P_o} = \frac{1}{2} \left[\left[1 - \left(\frac{v_2}{v_1} \right)^2 \right] \left[1 + \left(\frac{v_2}{v_1} \right) \right] \right] \quad 4.2$$

Dimana :

C_p = koefisien daya

P = daya mekanik yang dihasilkan rotor (watt)

P_o = daya mekanik total yang terkandung dalam angin yang melalui sudu (watt)

ρ_{udara} = massa jenis udara (kg/m³)

A = luas penampang bidang putar sudu (m²)

v_1 = kecepatan aliran udara sebelum melewati sudu rotor (m/s)

v_2 = kecepatan aliran udara setelah melewati sudu rotor (m/s)

Massa jenis udara diketahui 1,225, rumus untuk mencari luas penampang bidang putar adalah $\pi.r^2$, dan diketahui kecepatan angin sebelum dan sesudah turbin, maka dapat dihitung nilai CP pada turbin angin. Berikut nilai CP tertinggi pada masing-masing sudut *pitch*

Tabel 4.2 Nilai CP maksimum untuk setiap sudut *pitch* dan kecepatan angin

Sudut	Kecepatan Angin sebelum turbin	Kecepatan Angin setelah turbin	CP Maksimum
6°	2,6 m/s	1,2 m/s	0,57510
4°	1,9 m/s	0,9 m/s	0,57151
4°	1,6 m/s	0,8 m/s	0,56250
6°	2,3 m/s	1,3 m/s	0,53259
4°	1,8 m/s	1,0 m/s	0,53772
4°	1,7 m/s	0,9 m/s	0,55038

Dari tabel diatas dapat diketahui nilai CP tertinggi pada *prototype* turbin angin adalah 0,57510. Maka secara keseluruhan *prototype* turbin angin memiliki CP yang relative baik dan bernilai besar atau mendekati nilai Bet'z (sekitar 16/27 atau 0,593). Dari nilai CP tersebut, dapat dihitung daya yang bisa dihasilkan oleh turbin angin dengan menggunakan persamaan 2.5, yaitu

$$P = \frac{1}{2} \cdot \rho_{udara} \cdot Cp \cdot A \cdot v_1^3 \quad 4.3$$

Berikut nilai daya yang didapat dari turbin angin :

Tabel 4.3 Nilai daya yang diperoleh dari turbin angin

Sudut	CP Maksimum	Kecepatan Angin sebelum turbin	Daya (watt)
6°	0,57510	2,6 m/s	25,70957
4°	0,57151	1,9 m/s	9,97048
4°	0,56250	1,6 m/s	5,86022
6°	0,53259	2,3 m/s	16,48187
4°	0,53772	1,8 m/s	7,97641
4°	0,55038	1,7 m/s	6,877619

Berdasarkan hasil perhitungan, maka didapat daya tertinggi yang diperoleh oleh turbin angin yaitu saat CP tertinggi 0,57510 yaitu sebesar 25,70957 watt.

4.2 Pengujian Akurasi

Pengujian akurasi adalah pengujian yang ditujukan untuk mengetahui seberapa besar kinerja yang sesuai dari hasil perancangan kontrol logika *fuzzy* tipe 2 pada simulasi yaitu matlab dengan hasil perancangan program pada arduino. Nilai yang akan dibandingkan adalah nilai *output* pada simulasi matlab dan arduino sehingga dapat diketahui seberapa besar akurasi pada program arduino yang akan disambungkan pada *plant* turbin angin. Berikut tabel 4.4 hasil nilai perbandingan pada *output fuzzy* pada simulasi dan program arduino.

Tabel 4.4 Hasil nilai *output fuzzy* pada matlab dan arduino

E	DE	Mat lab	Ardu ino	Toler ansi	E	DE	Mat lab	Ardu ino	Toler ansi
0	0	7	7	0%	2	0	7	7	0%
0	1	7	7	0%	2	1	7	7	0%
0	2	11	11	0%	2	2	11	11	0%
0	3	11	11	0%	2	3	11	11	0%
0	4	11	11	0%	2	4	11	11	0%
0	5	11	11	0%	2	5	11	11	0%
0	-1	7	7	0%	2	-1	7	7	0%
0	-2	11	11	0%	2	-2	7	6	14%
0	-3	11	11	0%	2	-3	7	7	0%
0	-4	11	11	0%	2	-4	11	11	0%
0	-5	11	11	0%	2	-5	11	11	0%
1	0	7	7	0%	3	0	7	7	0%
1	1	7	7	0%	3	1	7	7	0%
1	2	11	11	0%	3	2	7	7	0%
1	3	11	11	0%	3	3	7	7	0%
1	4	11	11	0%	3	4	11	11	0%
1	5	11	11	0%	3	5	11	11	0%
1	-1	10,8	10	7,4%	3	-1	7	7	0%
1	-2	10,8	10	7,4%	3	-2	7	7	0%

Tabel 4.4 lanjutan

E	DE	Mat lab	Ardu ino	Toler ansi	E	DE	Mat lab	Ardu ino	Toler ansi
1	-3	10,8	10	7,4%	3	-3	7	7	0%
1	-4	11	11	0%	3	-4	11	11	0%
1	-5	11	11	0%	3	-5	11	11	0%
-1	0	30	30	0%	-3	0	32	32	0%
-1	1	30	30	0%	-3	1	32	32	0%
-1	2	30	30	0%	-3	2	30	30	0%
-1	3	30	30	0%	-3	3	30	30	0%
-1	4	11	11	0%	-3	4	30	30	0%
-1	5	11	11	0%	-3	5	30	30	0%
-1	-1	30	30	0%	-3	-1	32	32	0%
-1	-2	30	30	0%	-3	-2	32	32	0%
-1	-3	30	30	0%	-3	-3	32	32	0%
-1	-4	30	30	0%	-3	-4	32	32	0%
-1	-5	30	30	0%	-3	-5	32	32	0%
-2	0	30	30	0%	-4	0	32	32	0%
-2	1	30	30	0%	-4	1	32	32	0%
-2	2	30	29	3%	-4	2	30	30	0%
-2	3	30	30	0%	-4	3	30	30	0%
-2	4	11	10	9%	-4	4	30	29	3%
-2	5	11	11	0%	-4	5	30	30	0%
-2	-1	31,9	31	2,8%	-4	-1	32	32	0%
-2	-2	31,9	31	2,8%	-4	-2	32	32	0%
-2	-3	31,9	31	2,8%	-4	-3	32	32	0%
-2	-4	31,9	31	2,8%	-4	-4	32	32	0%
-2	-5	31,9	31	2,8%	-4	-5	32	32	0%

Berdasarkan tabel 4.4 diatas dapat diketahui bahwa toleransi terbesar yaitu hanya saat satu titik dengan besar 14% dan toleransi terkecil yaitu 0%. Ketidakakurasian sistem ini disebabkan karena beberapa kemungkinan yang salah satunya adalah kurang sesuainya nilai alternative keputusan pada sistem di matlab maupun arduino. Bila semua hasil toleransi dirata-ratakan, maka

akan mendapatkan total rata-rata sebesar 0,39%, hal ini menunjukkan bahwa perancangan kontrol dapat berfungsi dengan baik sesuai hasil dari simulasi pada matlab.

4.3 Analisis Respon Sistem Pengendalian

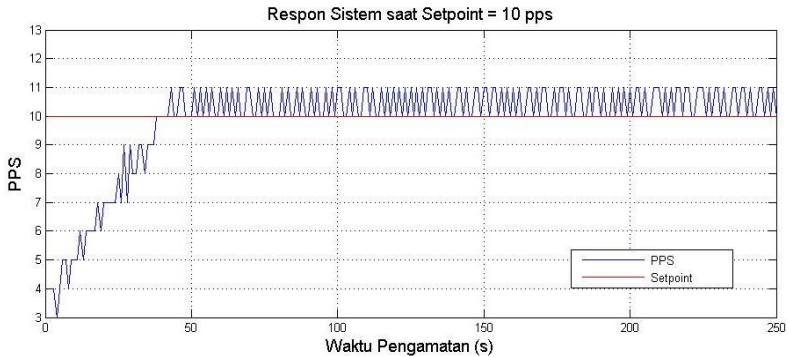
Analisis sistem pengendalian pada kontrol *fuzzy* tipe 2 dapat dilakukan dengan melihat hasil respon yang ditunjukkan *plant* sesuai dengan parameter respon berupa *setpoint* yang diberikan. Tujuan dari analisis ini untuk melihat kemampuan sistem untuk mengatasi gangguan, melihat respon sistem dengan *setpoint* yang diberikan, mempertahankan *setpoint* dan parameter lain seperti maksimum *overshoot*, *rise time*, *settling time*, *error steady state* dan ITAE. Maksimum *overshoot* adalah adalah nilai puncak maksimum dari respon *plant* yang diukur dari nilai akhir respon dan pada umumnya dirumuskan dalam persentase, yang berarti maksimum *overshoot* merupakan selisih nilai maksimum respon sistem dengan *setpoint*. *Rise time* merupakan waktu respon dimulai dari nol detik yang dibutuhkan *plant* untuk mencapai *setpoint* yang diberikan. *Settling time* adalah ukuran waktu yang menyatakan respon telah masuk dalam keadaan stabil atau *steady state*, dan *error steady state* adalah rata-rata *error* yang muncul saat sistem mencapai kondisi stabil atau *steady state*. ITAE atau *Integral Time Absolute Error* merupakan jumlah nilai absolut *error* yang terjadi selama proses pengamatan. ITAE dapat dirumuskan secara matematis dengan menggunakan persamaan berikut :

$$ITAE = \sum_{t=0}^n t|e_t| \quad 4.4$$

4.3.1 Analisis Pengujian *Setpoint*

Proses pengambilan data respon sistem pada analisis ini dilakukan dengan cara memberikan *setpoint* pada turbin angin. Kecepatan angin pada sumber angin diberikan nilai maksimum yaitu sebesar 2,6 m/s. Kondisi *setpoint* yang diberikan yaitu sebesar 10 pps, 15 pps, 20 pps dan 25 pps. Pemilihan nilai pps tersebut dikaitkan dengan kemampuan blower dalam

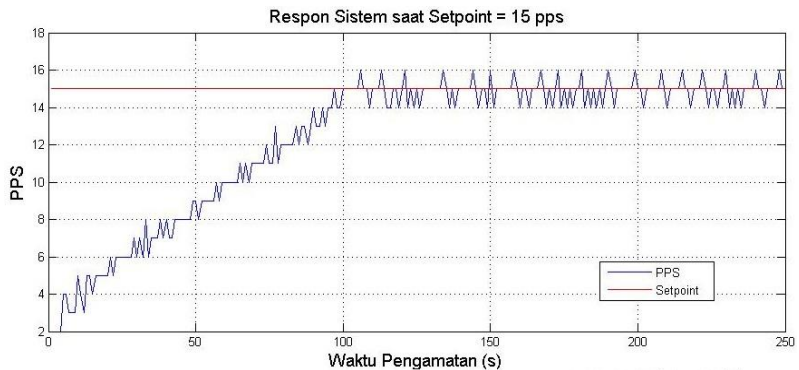
menghasilkan kecepatan angin. Gambar 4.2 menunjukkan respon sistem turbin angin saat diberikan *setpoint* sebesar 10 pps.



Gambar 4.2 Respon sistem saat *setpoint* = 10 pps

Ketika diberikan *setpoint* sebesar 10, respon sistem beresiliasi dengan rentang nilai dari 10 pps hingga 11 pps. Maksimum *overshoot* yang didapat sebesar 11 pps atau persentase maksimum *overshoot* sekitar 10%. Nilai *rise time* yang didapat yaitu saat detik ke 38 dan *settling time* atau waktu respon *plant* dalam keadaan *steady* setelah 29 detik. Nilai rata-rata respon sistem dalam keadaan *steady* untuk *setpoint* 10 yaitu sebesar 10,47 pps dan memiliki *error steady state* sebesar 0,51.

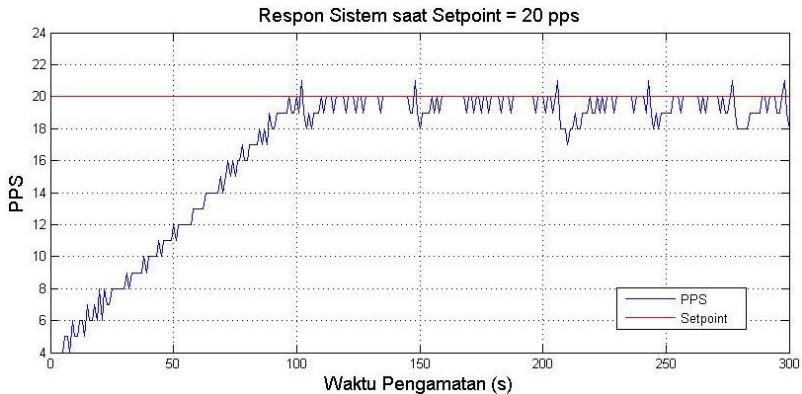
Berikut gambar 4.3 yang menunjukkan respon sistem turbin angin saat diberikan *setpoint* sebesar 15 pps.



Gambar 4.3 Respon sistem saat *setpoint* = 15 pps

Ketika diberikan *setpoint* sebesar 15, respon sistem beresilasi dengan rentang nilai dari 14 pps hingga 16 pps. Maksimum *overshoot* yang didapat sebesar 16 pps atau persentase maksimum *overshoot* sekitar 6,67%. Nilai *rise time* yang didapat yaitu saat detik ke 97 dan *settling time* atau waktu respon *plant* dalam keadaan *steady* setelah 100 detik. Nilai rata-rata respon sistem dalam keadaan *steady* untuk *setpoint* 15 yaitu sebesar 14,95 pps dan memiliki *error steady state* sebesar 0,31.

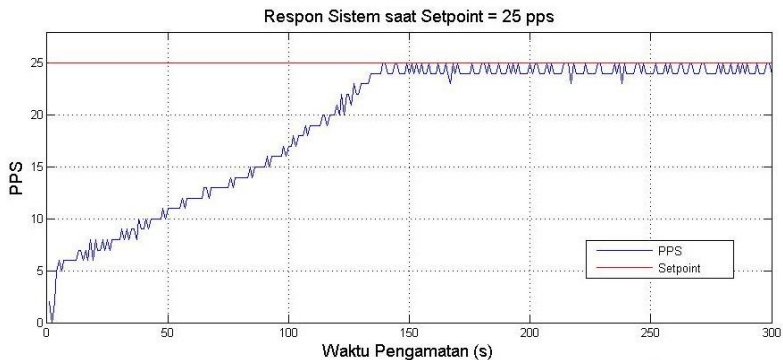
Berikut gambar 4.4 yang menunjukkan respon sistem turbin angin saat diberikan *setpoint* sebesar 20 pps.



Gambar 4.4 Respon sistem saat *setpoint* = 20 pps

Ketika diberikan *setpoint* sebesar 20, respon sistem beresilasi dengan rentang nilai dari 17 pps hingga 21 pps. Maksimum *overshoot* yang didapat sebesar 21 pps atau persentase maksimum *overshoot* sekitar 5%. Nilai *rise time* yang didapat yaitu saat detik ke 97 dan *settling time* atau waktu respon *plant* dalam keadaan *steady* setelah 97 detik. Nilai rata-rata respon sistem dalam keadaan *steady* untuk *setpoint* 20 yaitu sebesar 19,36 pps dan memiliki *error steady state* sebesar 0,59.

Berikut gambar 4.5 yang menunjukkan respon sistem turbin angin saat diberikan *setpoint* sebesar 25 pps.



Gambar 4.5 Respon sistem saat *setpoint* = 25

Ketika diberikan *setpoint* sebesar 25, respon sistem berhasil dengan rentang nilai dari 23 pps hingga 25 pps. Maksimum *overshoot* yang didapat sebesar 25 pps atau persentase maksimum *overshoot* sebesar 0%. Nilai *rise time* yang didapat yaitu saat detik ke 140 dan *settling time* atau waktu respon *plant* dalam keadaan *steady* setelah 134 detik. Nilai rata-rata respon sistem dalam keadaan *steady* untuk *setpoint* 25 yaitu sebesar 24,32 pps dan memiliki *error steady state* sebesar 0,68. Dari 4 *setpoint* yang telah diujikan, respon sistem dari interval logika *fuzzy* tipe 2 terhadap pengendalian sudut *pitch* dapat dilihat pada tabel 4.5 berikut ini.

Tabel 4.5 Analisis respon sistem pengendalian interval *fuzzy* tipe 2 saat pengujian *setpoint*

<i>Setpoint</i> (PPS)	<i>Rise time</i> (s)	<i>Settling time</i> (s)	Maksimum <i>Overshoot</i> (%)	Rata-rata Kondisi <i>Steady</i>	<i>Error Steady State</i> (%)
10	38	29	10%	10,47	5,1%
15	97	100	6,67%	14,95	2,06%
20	97	97	5%	19,36	2,95%
25	140	134	0%	24,32	2,72%

Dari tabel 4.5 diatas, dapat diketahui bahwa sistem pengendalian sudut *pitch* pada turbin angin menggunakan interval logika *fuzzy* tipe 2 dapat stabil terhadap semua *setpoint* yang

diberikan. Respon terbaik yang didapat pada pengujian yaitu saat *setpoint* 25, dengan tidak adanya maksimum *overshoot* atau sama dengan nol dan *error steady* yang kecil yaitu 2,72%. Jika ditinjau dari besarnya *error steady state* yang dihasilkan, tentu memenuhi toleransi *error* yang diberikan (5% dari *setpoint*) yaitu ketika *setpoint* 15, 20 dan 25. Saat *setpoint* 10 juga dapat dikatakan baik, karena hampir memenuhi kesalahan toleransi 5% dari *setpoint* (sebesar 5,1%). Secara keseluruhan respon sistem dapat dikatakan stabil. Maka kontroler yang dirancang menggunakan interval logika *fuzzy* tipe 2 pada turbin angin ini memiliki kinerja yang cukup baik untuk diterapkan karena nilai *error* yang masih dibawah toleransi. *Rise time* dan *settling time* tercepat yaitu ketika *setpoint* diberikan nilai 10, karena respon *plant* mampu mencapainya dalam waktu 38 detik dan 29 detik, sedangkan *rise time* dan *settling time* terlama yaitu ketika *setpoint* diberikan nilai 25, karena respon *plant* mampu mencapainya dalam waktu 140 detik dan 134 detik.

Adapun nilai ITAE dari empat *setpoint* dapat dilihat pada tabel 4.6 sebagai berikut

Tabel 4.6 Nilai ITAE dengan $t=400s$

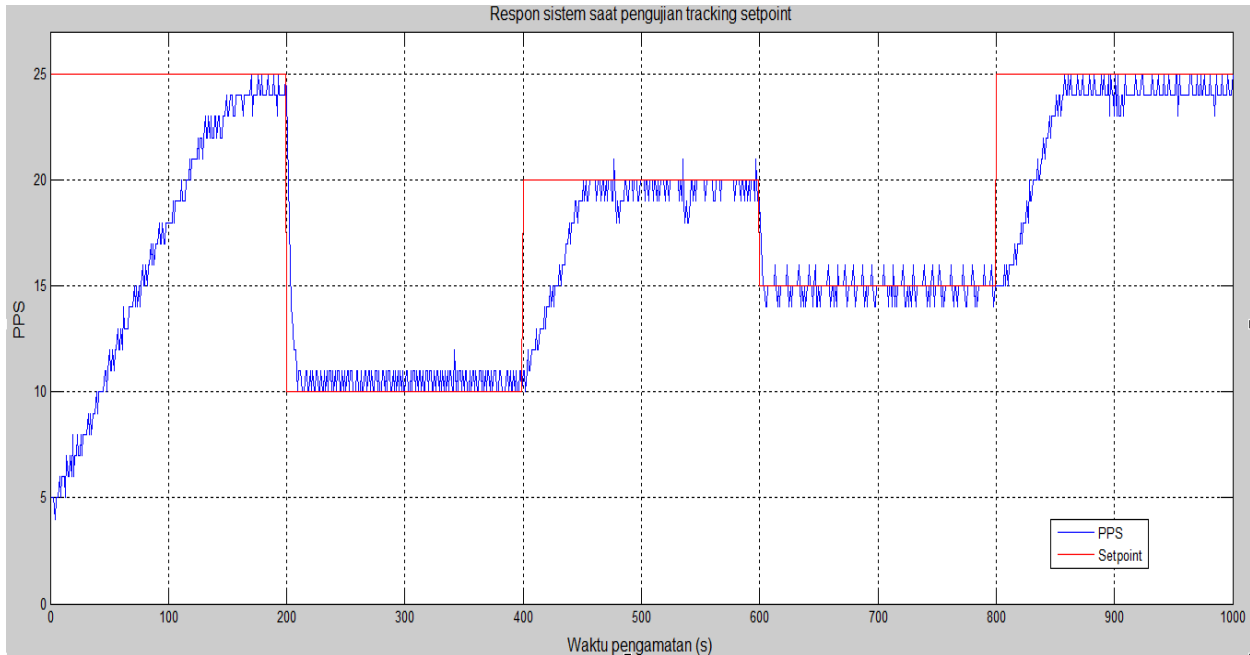
No	<i>Setpoint</i> (pps)	ITAE
1	10	307
2	15	731
3	20	1016
4	25	1787

ITAE merupakan jumlah nilai absolut *error* yang terjadi selama proses pengamatan. Nilai dari ITAE dihitung selama waktu proses pengamatan. Pada pengujian respon sistem pada turbin angin, rata-rata dilakukan selama 400 detik, maka ITAE dihitung dalam rentang waktu 400 detik. ITAE dihitung dengan cara menjumlahkan semua nilai absolut *error* yang dihasilkan dari respon sistem pengendalian. Dengan menghitung nilai ITAE maka dapat diketahui jumlah *error* absolut respon sistem terhadap *setpoint* yang diberikan. ITAE dihitung berdasarkan nilai *setpoint*

yang diberikan, dengan 4 variabel *setpoint* yang berbeda, maka terdapat 4 ITAE yang berbeda.. Nilai ITAE semakin besar atau bertambah ketika *setpoint* nya juga semakin besar, ini dikarenakan terdapat *error* yang besar untuk *setpoint* yang besar pada saat waktu respon mencapai kondisi *steady*. Begitupun sebaliknya, bila *setpoint* yang berikan kecil maka nilai ITAE pun juga semakin kecil, karena *error* yang dihasilkan juga kecil atau berada pada nilai sekitar *setpoint* yang diberikan. Maka dapat diambil kesimpulan bahwa semakin besar *setpoint* yang diberikan, maka semakin besar pula nilai ITAE nya,

4.3.2 Analisis Pengujian *Tracking Setpoint*

Pengujian *tracking setpoint* pada sistem turbin angin merupakan bagian dari pengujian kontrol logika *fuzzy* tipe secara menyeluruh. Pengujian dilakukan dengan mengubah nilai *setpoint* ketika sistem telah mampu mencapai keadaan mantap. *Setpoint* yang diberikan yaitu kesemua *setpoint* yang diuji sebelumnya yaitu 10, 15, 20, 25 akan tetapi diberikan *setpoint* secara acak, atau dari tinggi menuju rendah maupun sebaliknya. Pengujian yang dilakukan yaitu dengan memberikan nilai awal *setpoint* tertinggi yaitu 25 - 10 - 20 - 15 - 25. Pada pengujian ini, diharapkan turbin angin mampu mengikuti nilai *setpoint* yang berubah-ubah. Kecepatan angin yang diberikan pun juga konstan seperti pengujian *setpoint* sebelumnya yaitu kecepatan angin paling tinggi sebesar 2,6 m/s. Perubahan *setpoint* menuju *setpoint* berikutnya diberikan dengan waktu 200 detik. Berikut grafik respon pengujian *tracking setpoint* pada turbin angin.



Gambar 4.6 Grafik respon sistem saat pengujian *tracking setpoint*

Tabel 4.7 Analisis respon sistem pengendalian interval *fuzzy* tipe 2 saat pengujian *tracking setpoint*

<i>Setpoint</i> (PPS)	<i>Rise time</i> (s)	<i>Settling time</i> (s)	Maksimum <i>Overshoot</i>	Rata-rata <i>Kondisi Steady</i>	<i>Error Steady State</i>
0-25	170	158	0%	24,09	3,61%
25-10	10	10	14%	10,51	2,05%
10-20	52	45	5%	19,6	1,71%
20-15	4	3	26,6%	14,9	1,35%
15-25	59	57	0%	24,14	3,40%

Dari tabel 4.7 diatas, maka dapat diketahui bahwa respon *tracking setpoint* pada turbin angin dapat mengikuti *setpoint* yang diberikan. *Error steady state* tertinggi yaitu saat *setpoint* 0 dirubah ke 25, dan keseluruhan nilai *error steady state* pada tiap *setpoint* masih dibawah toleransi 5% yang berarti kontrol logika *fuzzy* tipe 2 dapat berjalan stabil dan mantap mengikuti *setpoint*. Nilai rata-rata kondisi *steady* pada tiap perubahan *setpoint* uga cukup baik, yaitu tidak terlampau jauh dari perubahan *setpoint* yang diberikan. Nilai maksimum *overshoot* tertinggi saat *setpoint* 20 menuju 15, hal ini dikarenakan penurunan *setpoint* yang lebih tinggi menuju *setpoint* yang rendah, sehingga menghasilkan oveshoot yang tinggi. Waktu untuk mencapai puncak / *rise time* paling cepat yaitu saat perubahan *setpoint* dari 20 menuju 15 dan waktu *settling* atau waktu mencapai kondisi *steady* juga saat perubahan *setpoint* dari 20 menuju 15, hal ini dikarenakan respon dalam keadaan penurunan *setpoint* yang tidak terlalu besar yaitu dari 20 menuju 15. Ketika waktu penurunan dari *setpoint* tertinggi yaitu 25 menuju *setpoint* paling rendah yaitu 10, membutuhkan waktu untuk mencapai puncak selama 10 detik, begitupun waktu untuk mencapai *steadynya* juga 10 detik, hal ini dikarenakan jarak perubahan *setpoint* yang besar membutuhkan waktu yang cukup lama untuk mengikuti *setpoint* yang diberikan. Nilai *rise time* yang paling lama yaitu saat *setpoint* 0 dirubah menuju *setpoint* 25, membutuhkan waktu 170 detik dan waktu untuk *steadynya* yaitu 158 detik, hal ini berhubungan dengan performa turbin angin saat

diberikan sudut *pitch* untuk menaikkan kecepatan putar turbin angin. Permasalahan ini berlaku untuk semua perubahan *setpoint* yang membutuhkan kenaikan kecepatan putar turbin angin. Secara keseluruhan respon sistem dapat dikatakan stabil. Maka kontroler yang dirancang menggunakan interval logika *fuzzy* tipe 2 pada pengujian tracking *setpoint* angin ini memiliki kinerja yang cukup baik untuk diterapkan karena nilai *error* yang masih dibawah toleransi pada semua perubahan *setpoint*.

BAB V PENUTUP

5.1. Kesimpulan

Dari hasil analisis terkait penelitian tugas akhir yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut :

- a. Telah berhasil dirancang *prototype* turbin angin dengan bilah standar NREL S83N dengan turbin angin dapat menghasilkan RPM tertinggi sebesar 94 pada kecepatan angin 2,6 m/s, RPM sebesar 86 pada kecepatan angin 1,9 m/s, RPM sebesar 83 pada kecepatan angin 1,6 m/s, RPM sebesar 92 pada kecepatan angin 2,3 m/s, RPM sebesar 88 pada kecepatan angin 1,8 m/s dan RPM sebesar 91 pada kecepatan angin 1,7 m/s. Saat rentang sudut pitch 0° - 2° turbin angin berputar dengan kecepatan rendah, sedangkan pada sudut pitch 4° - 6° turbin angin berputar pada kecepatan maksimumnya. Saat sudut pitch 8° - 20° , kecepatan turbin angin cenderung melambat. Nilai *Coefficient Power* terbesar didapatkan sebesar 0,57510 pada sudut pitch 6° saat kecepatan angin terbesar yaitu 2,6 m/s, dan daya terbesar yang dihasilkan sebesar 25,7095 watt.
- b. Berdasarkan analisa yang telah dilakukan, *rise time* dan *settling time* terbaik atau tercepat untuk mencapai *setpoint* yaitu saat *setpoint* 10 dengan *rise time* 38 detik dan *settling time* 29 detik, sedangkan yang terlama untuk mencapai *setpoint* saat *setpoint* 25 dengan *rise time* 140 detik dan *settling time* 134 detik, dikarenakan turbin membutuhkan waktu untuk mencapai puncak putaran maksimum. Maksimum *overshoot* tertinggi yaitu saat *setpoint* 10 dengan 10% dan terendah yaitu saat *setpoint* 25 dengan 0%. *Error steady state* tertinggi yaitu saat *setpoint* 10 dengan 5,1% dan *error steady state* terendah adalah *setpoint* 15 dengan 2,06%. Nilai ITAE terkecil dengan nilai 307 saat *setpoint* 10, ITAE terbesar sebesar 1787 saat *setpoint* 25. Sistem pengendalian yang baik dan cocok untuk respon sistem turbin angin ini yaitu saat *setpoint* 25 pps, karena memiliki *error steady* yang kecil (dibawah toleransi 5%) dan tidak adanya maksimum *overshoot*.

5.2. Saran

Beberapa saran untuk penelitian ini selanjutnya adalah sebagai berikut :

- a. Pengaplikasian performa dari turbin angin dengan generator listrik sehingga mampu menghasilkan daya (*watt*) dan dibandingkan hasil daya yang terukur dengan hasil daya perhitungan,
- b. Perancangan kontrol logika *fuzzy* dibuat dengan lebih teliti lagi agar mencapai respon pengendalian yang lebih baik.
- c. Penambahan sumber kecepatan angin yang jauh lebih besar.

DAFTAR PUSTAKA

- Musyafa', Ali. 2016. Turbin Angin & Sistem Kontrol. Surabaya: PT Revka Petra Media
- Burton, Tony, et al. 2001. Wind Energy Handbook. New York : John Wiley & Sons, LTD.
- Johnson, Gary L. 2006. Wind Energy System. Manhattan : KS.
- Piggot, Hugh. 2001. Wind Power Workshop. s.l. : Centre for Alternative Technology Publication.
- Rand, Joseph. 2010. Wind Project: Wind Turbine Blade Design. s.l. : National Wind Technology Center.
- S. V. Devika, Sk. Khamuruddeen, Sk. Khamurunnisa, Jayanth Thota, Khalesha Shaik. 2014. Arduino Based Automatic Plant Watering System. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4 Issue 10.
- Buhl, M. 2009. Wind Turbine Airfoils. National Renewable Energy Laboratory. Online]. [Cited: August 29, 2016.] [Http://Wind.Nrel.Gov/Airfoils/](http://Wind.Nrel.Gov/Airfoils/).
- Mendel dan Robert I. Bob John. 2002 "Type-2 Fuzzy Sets Made Simple Jerry", IEEE Transactions On Fuzzy Systems, Vol. 10, No. 2.
- Ali Musyafa'. 2012. Rancang Bangun Kontrol Logika Fuzzy Pada Sudut Angguk Turbin Angin Untuk Optimisasi Daya Listrik Di Ladang Angin Jawa Timur – Indonesia. Surabaya : Institut Teknologi Sepuluh Nopember.
- Ridha, Farid. 2011. Pemilihan Sudut Pitch Optimal Untuk Prototipe Turbin Angin Skala Kecil Dengan Tipe Bilah on-Uniform Airfoil NREL S83n. Surabaya : Institut Teknologi Sepuluh Nopember.
- Dan M. Somers Airfoils Incorporated. 2001. The S833, S834, and S835 Airfoils NREL National Renewable Energy Laboratory. Port Matilda, Pennsylvania November 2001 – November 2002.
- Adi, Adri. 2011. Rancang Bangun Sistem Pengendalian Kemiringan Sudut Blade Pada Prototype Turbin Angin

- Berbasis Logika Fuzzy. Teknik Fisika-FTI-ITS. Surabaya,.
- Aftika, Dwi. 2012. Rancang Bangun Blade Pitch Angle Control Prototype Turbin Angin untuk skala kecil Berbasis Interval Fuzzy Logic Type II. Teknik Fisika-FTI-ITS. Surabaya.
- Sigit, Rianto. dkk. 2007. Motor Servo. Surabaya: Politeknik Elektronika Negeri Surabaya.
- da Rosa, Aldo Vieira. 2009. Fundamentals Of Renewable Energy Process (second edition). USA : Stanford University.
- Harika, Adam. 2008. Rancang Bangun Blade Pitch Angle Control System Berbasis Classic-Fuzzy Pada Prototipe Wind Turbine. Teknik Fisika-FTI-ITS. Surabaya.
- Aria, Muhammad. 2011. Perancangan Pengendali Fuzzy Bertipe-2 Untuk Masalah Backer-Upper Truk . Teknik Elektro. Universitas Komputer Indonesia.
- Humaira, Mira. 2014. Perbandingan Algoritma Reduksi Tipe Pada Fuzzy Tipe-2. Teknologi Informasi. Politeknik Negeri Padang Indonesia.

LAMPIRAN A

Program Arduino

```
int sp = 25; //sesuai setpoint yang diinginkan
int waktu=0;
int encoder0PinA = 2;
int encoder0Pos = 0;
int encoder0PinALast = LOW;
int n = LOW;
int pos;
long errorsebelum=0;
double error [14], deltaerror [14], rules [14][14];
int pps = 0;
float turbinerror, turbindeltaerror, defuzzy, firing, temp, rule0, rule1,
rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10, rule11, rule12,
rule13, rule14, rule15, rule16, rule17, rule18, rule19, rule20, rule21,
rule22, rule23, rule24, rule25, rule26, rule27, rule28, rule29, rule30,
rule31, rule32, rule33, rule34, rule35, rule36, rule37, rule38, rule39,
rule40, rule41, rule42, rule43, rule44, rule45, rule46, rule47, rule48,
rule49, rule50, rule51, rule52, rule53, rule54, rule55, rule56, rule57,
rule58, rule59, rule60, rule61, rule62, rule63, rule64, rule65, rule66,
rule67, rule68, rule69, rule70, rule71, rule72, rule73, rule74, rule75,
rule76, rule77, rule78, rule79, rule80, rule81, rule82, rule83, rule84,
rule85, rule86, rule87, rule88, rule89, rule90, rule91, rule92, rule93,
rule94, rule95, rule96, rule97, rule98, rule99, rule100, rule101,
rule102, rule103, rule104, rule105, rule106, rule107, rule108, rule109,
rule110, rule111, rule112, rule113, rule114, rule115, rule116, rule117,
rule118, rule119, rule120, rule121, rule122, rule123, rule124, rule125,
rule126, rule127, rule128, rule129, rule130, rule131, rule132, rule133,
rule134, rule135, rule136, rule137, rule138, rule139, rule140, rule141,
rule142, rule143, rule144, rule145, rule146, rule147, rule148, rule149,
rule150, rule151, rule152, rule153, rule154, rule155, rule156, rule157,
rule158, rule159, rule160, rule161, rule162, rule163, rule164, rule165,
rule166, rule167, rule168, rule169, rule170, rule171, rule172, rule173,
rule174, rule175, rule176, rule177, rule178, rule179, rule180, rule181,
rule182, rule183, rule184, rule185, rule186, rule187, rule188, rule189,
rule190, rule191, rule192, rule193, rule194, rule195;
unsigned long lastmillis = 0;

#include <Servo.h>
```

A-2

```
Servo myservo;  
Servo myservo1;  
Servo myservo2;
```

```
void setup()  
{  
  myservo.attach(8);  
  myservo1.attach(9);  
  myservo2.attach(10);  
  pinMode (encoder0PinA,INPUT);  
  Serial.begin (9600);  
}
```

```
void FuzzyError(){  
  //untuk NHU  
  if (turbinerror <= -20)  
    { error [0] = 1;}  
  else if (turbinerror > -20 && turbinerror <= -4)  
    { error [0] = (-4-turbinerror)/(-4+20);}  
  else  
    { error [0] = 0;}  
  
  // untuk NMU  
  if (turbinerror <= -3)  
    { error [1] = 0;}  
  else if (turbinerror > -3 && turbinerror <= -2)  
    { error [1] = (turbinerror+3)/(-2+3);}  
  else if (turbinerror > -2 && turbinerror <= -1)  
    { error [1] = (-1-turbinerror)/(-1+2);}  
  else  
    { error [1] = 0;}  
  
  // untuk NLU  
  if (turbinerror <= -2.1)  
    { error [2] = 0;}  
  else if (turbinerror > -2.1 && turbinerror <= -1)  
    { error [2] = (turbinerror+2.1)/(-1+2.1);}  
  else if (turbinerror > -1 && turbinerror <= 0)  
    { error [2] = (0-turbinerror)/(0+1);}  
  else
```

```

{ error [2] = 0;}

// untuk ZEU
if (turbinerror <= -0.9)
{ error [3] = 0;}
else if (turbinerror > -0.9 && turbinerror <= 0)
{ error [3] = (turbinerror+0.9)/(0+0.9);}
else if (turbinerror > 0 && turbinerror <= 0.1)
{ error [3] = (0.1-turbinerror)/(0.1-0);}
else
{ error [3] = 0;}

// untuk PLU
if (turbinerror <= -0)
{ error [4] = 0;}
else if (turbinerror > -0 && turbinerror <= 1.1)
{ error [4] = (turbinerror+0)/(1.1+0);}
else if (turbinerror > 1.1 && turbinerror <= 2)
{ error [4] = (2-turbinerror)/(2-1.1);}
else
{ error [4] = 0;}

// untuk PMU
if (turbinerror <= 0.9)
{ error [5] = 0;}
else if (turbinerror > 0.9 && turbinerror <= 2)
{ error [5] = (turbinerror-0.9)/(2-0.9);}
else if (turbinerror > 2 && turbinerror <= 3)
{ error [5] = (3-turbinerror)/(3-2);}
else
{ error [5] = 0;}

// untuk PHU
if (turbinerror <= 4)
{ error [6] = 0;}
else if (turbinerror > 4 && turbinerror <= 20)
{ error [6] = (turbinerror-4)/(20-4);}
else
{ error [6] = 1;}

```

A-4

```
// untuk NHL
if (turbinerror <= -20)
{ error [7] = 1;}
else if (turbinerror > -20 && turbinerror <= -2.5)
{ error [7] = (-2.5-turbinerror)/(-2.5+20);}
else
{ error [7] = 0;}
```

```
// untuk NML
if (turbinerror <= -2.5)
{ error [8] = 0;}
else if (turbinerror > -2.5 && turbinerror <= -2)
{ error [8] = (turbinerror+2.5)/(-2+2.5);}
else if (turbinerror > -2 && turbinerror <= -1.5)
{ error [8] = (-1.5-turbinerror)/(-1.5+2);}
else
{ error [8] = 0;}
```

```
// untuk NLL
if (turbinerror <= -1.6)
{ error [9] = 0;}
else if (turbinerror > -1.6 && turbinerror <= -1)
{ error [9] = (turbinerror+1.6)/(-1+1.6);}
else if (turbinerror > -1 && turbinerror <= -0.5)
{ error [9] = (-0.5-turbinerror)/(-0.5+1);}
else
{ error [9] = 0;}
```

```
// untuk ZEL
if (turbinerror <= -0.6)
{ error [10] = 0;}
else if (turbinerror > -0.6 && turbinerror <= 0)
{ error [10] = (turbinerror+0.6)/(0+0.6);}
else if (turbinerror > 0 && turbinerror <= 0.6)
{ error [10] = (0.6-turbinerror)/(0.6-0);}
else
{ error [10] = 0;}
```

```
// untuk PLL
if (turbinerror <= 0.5)
```

```

{ error [11] = 0;}
  else if (turbinerror > 0.5 && turbinerror <= 1)
{ error [11] = (turbinerror-0.5)/(1-0.5);}
  else if (turbinerror > 1 && turbinerror <= 1.5)
{ error [11] = (1.5-turbinerror)/(1.5-1);}
  else
{ error [11] = 0;}

// untuk PML
if (turbinerror <= 1.4)
{ error [12] = 0;}
  else if (turbinerror > 1.4 && turbinerror <= 2)
{ error [12] = (turbinerror-1.4)/(2-1.4);}
  else if (turbinerror > 2 && turbinerror <= 2.5)
{ error [12] = (2.5-turbinerror)/(2.5-2);}
  else
{ error [12] = 0;}

// untuk PHL
if (turbinerror <= 2.5)
{ error [13] = 0;}
  else if (turbinerror > 2.5 && turbinerror <= 20)
{ error [13] = (turbinerror-2.5)/(20-2.5);}
  else
{ error [13] = 1;}

}
void FuzzyDeltaerror(){
// untuk NBU
  if (turbindeltaerror <= -5)
{ deltaerror [0] = 1;}
  else if (turbindeltaerror > -5 && turbindeltaerror <= -3.7)
{ deltaerror [0] = (-3.7-turbindeltaerror)/(-3.7+5);}
  else
{ deltaerror [0] = 0;}

// untuk NMU
  if (turbindeltaerror <= -4)
{ deltaerror [1] = 0;}
  else if (turbindeltaerror > -4 && turbindeltaerror <= -3)

```

A-6

```
{ deltaerror [1] = (turbindeltaerror+4)/(-3+4);}
  else if (turbindeltaerror > -3 && turbindeltaerror <= -1.8)
  { deltaerror [1] = (-1.8-turbindeltaerror)/(-1.8+3);}
  else
  { deltaerror [1] = 0;}

// untuk NSU
if (turbindeltaerror <= -2)
{ deltaerror [2] = 0;}
  else if (turbindeltaerror > -2 && turbindeltaerror <= -1)
  { deltaerror [2] = (turbindeltaerror+2)/(-1+2);}
  else if (turbindeltaerror > -1 && turbindeltaerror <= -0.1)
  { deltaerror [2] = (-0.1-turbindeltaerror)/(-0.1+1);}
  else
  { deltaerror [2] = 0;}

// untuk ZEU
if (turbindeltaerror <= -0.9)
{ deltaerror [3] = 0;}
  else if (turbindeltaerror > -0.9 && turbindeltaerror <= 0)
  { deltaerror [3] = (turbindeltaerror+0.9)/(0+0.9);}
  else if (turbindeltaerror > 0 && turbindeltaerror <= 0.9)
  { deltaerror [3] = (0.9-turbindeltaerror)/(0.9-0);}
  else
  { deltaerror [3] = 0;}

// untuk PSU
if (turbindeltaerror <= 0.1)
{ deltaerror [4] = 0;}
  else if (turbindeltaerror > 0.1 && turbindeltaerror <= 1)
  { deltaerror [4] = (turbindeltaerror-0.1)/(1-0.1);}
  else if (turbindeltaerror > 1 && turbindeltaerror <= 2)
  { deltaerror [4] = (2-turbindeltaerror)/(2-1);}
  else
  { deltaerror [4] = 0;}

// untuk PMU
if (turbindeltaerror <= 1.8)
{ deltaerror [5] = 0;}
  else if (turbindeltaerror > 1.8 && turbindeltaerror <= 3)
```

```

{ deltaerror [5] = (turbindeltaerror-1.8)/(3-1.8);}
  else if (turbindeltaerror > 3 && turbindeltaerror <= 4)
{ deltaerror [5] = (4-turbindeltaerror)/(4-3);}
  else
{ deltaerror [5] = 0;}

// untuk PBU
if (turbindeltaerror <= 3.7)
{ deltaerror [6] = 0;}
  else if (turbindeltaerror > 3.7 && turbindeltaerror <= 5)
{ deltaerror [6] = (turbindeltaerror-3.7)/(5-3.7);}
  else
{ deltaerror [6] = 1;}

// untuk NBL
if (turbindeltaerror <= -5)
{ deltaerror [7] = 1;}
  else if (turbindeltaerror > -5 && turbindeltaerror <= -4.5)
{ deltaerror [7] = (-4.5-turbindeltaerror)/(-4.5+5);}
  else
{ deltaerror [7] = 0;}

// untuk NML
if (turbindeltaerror <= -3.5)
{ deltaerror [8] = 0;}
  else if (turbindeltaerror > -3.5 && turbindeltaerror <= -3)
{ deltaerror [8] = (turbindeltaerror+3.5)/(-3+3.5);}
  else if (turbindeltaerror > -3 && turbindeltaerror <= -2.3)
{ deltaerror [8] = (-2.3-turbindeltaerror)/(-2.3+3);}
  else
{ deltaerror [8] = 0;}

// untuk NSL
if (turbindeltaerror <= -1.5)
{ deltaerror [9] = 0;}
  else if (turbindeltaerror > -1.5 && turbindeltaerror <= -1)
{ deltaerror [9] = (turbindeltaerror+1)/(-1+1.5);}
  else if (turbindeltaerror > -1 && turbindeltaerror <= -0.6)
{ deltaerror [9] = (-0.6-turbindeltaerror)/(-0.6+1);}
  else

```

A-8

```
{ deltaerror [9] = 0;}

// untuk ZEL
if (turbindeltaerror <= -0.5)
{ deltaerror [10] = 0;}
else if (turbindeltaerror > -0.5 && turbindeltaerror <= 0)
{ deltaerror [10] = (turbindeltaerror+0.5)/(0+0.5);}
else if (turbindeltaerror > 0 && turbindeltaerror <= 0.5)
{ deltaerror [10] = (0.5-turbindeltaerror)/(0.5-0);}
else
{ deltaerror [10] = 0;}

// untuk PSL
if (turbindeltaerror <= 0.6)
{ deltaerror [11] = 0;}
else if (turbindeltaerror > 0.6 && turbindeltaerror <= 1)
{ deltaerror [11] = (turbindeltaerror-0.6)/(1-0.6);}
else if (turbindeltaerror > 1 && turbindeltaerror <= 1.5)
{ deltaerror [11] = (1.5-turbindeltaerror)/(1.5-1);}
else
{ deltaerror [11] = 0;}

// untuk PML
if (turbindeltaerror <= 2.3)
{ deltaerror [12] = 0;}
else if (turbindeltaerror > 2.3 && turbindeltaerror <= 3)
{ deltaerror [12] = (turbindeltaerror-2.3)/(3-2.3);}
else if (turbindeltaerror > 3 && turbindeltaerror <= 3.5)
{ deltaerror [12] = (3.5-turbindeltaerror)/(3.5-3);}
else
{ deltaerror [12] = 0;}

// untuk PBL
if (turbindeltaerror <= 4.5)
{ deltaerror [13] = 0;}
else if (turbindeltaerror > 4.5 && turbindeltaerror <= 5)
{ deltaerror [13] = (turbindeltaerror-4.5)/(5-4.5);}
else
{ deltaerror [13] = 1;}
}
```



```
void RuleEvaluation(){
  int i, j;
  for ( i=0; i<=13; i=i+1)
  {
    for ( j=0; j<=13; j=j+1)
    { temp = min (error[i], deltaerror[j]);
      rules [i][j] = temp; }
    }
  rule0 = rules [0][0];
  rule1 = rules [0][1];
  rule2 = rules [0][2];
  rule3 = rules [0][3];
  rule4 = rules [0][4];
  rule5 = rules [0][5];
  rule6 = rules [0][6];
  rule7 = rules [0][7];
  rule8 = rules [0][8];
  rule9 = rules [0][9];
  rule10 = rules [0][10];
  rule11 = rules [0][11];
  rule12 = rules [0][12];
  rule13 = rules [0][13];
  rule14 = rules [1][0];
  rule15 = rules [1][1];
  rule16 = rules [1][2];
  rule17 = rules [1][3];
  rule18 = rules [1][4];
  rule19 = rules [1][5];
  rule20 = rules [1][6];
  rule21 = rules [1][7];
  rule22 = rules [1][8];
  rule23 = rules [1][9];
  rule24 = rules [1][10];
  rule25 = rules [1][11];
  rule26 = rules [1][12];
  rule27 = rules [1][13];
  rule28 = rules [2][0];
  rule29 = rules [2][1];
  rule30 = rules [2][2];
  rule31 = rules [2][3];
```

A-10

rule32 = rules [2][4];
rule33 = rules [2][5];
rule34 = rules [2][6];
rule35 = rules [2][7];
rule36 = rules [2][8];
rule37 = rules [2][9];
rule38 = rules [2][10];
rule39 = rules [2][11];
rule40 = rules [2][12];
rule41 = rules [2][13];
rule42 = rules [3][0];
rule43 = rules [3][1];
rule44 = rules [3][2];
rule45 = rules [3][3];
rule46 = rules [3][4];
rule47 = rules [3][5];
rule48 = rules [3][6];
rule49 = rules [3][7];
rule50 = rules [3][8];
rule51 = rules [3][9];
rule52 = rules [3][10];
rule53 = rules [3][11];
rule54 = rules [3][12];
rule55 = rules [3][13];
rule56 = rules [4][0];
rule57 = rules [4][1];
rule58 = rules [4][2];
rule59 = rules [4][3];
rule60 = rules [4][4];
rule61 = rules [4][5];
rule62 = rules [4][6];
rule63 = rules [4][7];
rule64 = rules [4][8];
rule65 = rules [4][9];
rule66 = rules [4][10];
rule67 = rules [4][11];
rule68 = rules [4][12];
rule69 = rules [4][13];
rule70 = rules [5][0];
rule71 = rules [5][1];

```
rule72 = rules [5][2];
rule73 = rules [5][3];
rule74 = rules [5][4];
rule75 = rules [5][5];
rule76 = rules [5][6];
rule77 = rules [5][7];
rule78 = rules [5][8];
rule79 = rules [5][9];
rule80 = rules [5][10];
rule81 = rules [5][11];
rule82 = rules [5][12];
rule83 = rules [5][13];
rule84 = rules [6][0];
rule85 = rules [6][1];
rule86 = rules [6][2];
rule87 = rules [6][3];
rule88 = rules [6][4];
rule89 = rules [6][5];
rule90 = rules [6][6];
rule91 = rules [6][7];
rule92 = rules [6][8];
rule93 = rules [6][9];
rule94 = rules [6][10];
rule95 = rules [6][11];
rule96 = rules [6][12];
rule97 = rules [6][13];
rule98 = rules [7][0];
rule99 = rules [7][1];
rule100 = rules [7][2];
rule101 = rules [7][3];
rule102 = rules [7][4];
rule103 = rules [7][5];
rule104 = rules [7][6];
rule105 = rules [7][7];
rule106 = rules [7][8];
rule107 = rules [7][9];
rule108 = rules [7][10];
rule109 = rules [7][11];
rule110 = rules [7][12];
rule111 = rules [7][13];
```

A-12

rule112 = rules [8][0];
rule113 = rules [8][1];
rule114 = rules [8][2];
rule115 = rules [8][3];
rule116 = rules [8][4];
rule117 = rules [8][5];
rule118 = rules [8][6];
rule119 = rules [8][7];
rule120 = rules [8][8];
rule121 = rules [8][9];
rule122 = rules [8][10];
rule123 = rules [8][11];
rule124 = rules [8][12];
rule125 = rules [8][13];
rule126 = rules [9][0];
rule127 = rules [9][1];
rule128 = rules [9][2];
rule129 = rules [9][3];
rule130 = rules [9][4];
rule131 = rules [9][5];
rule132 = rules [9][6];
rule133 = rules [9][7];
rule134 = rules [9][8];
rule135 = rules [9][9];
rule136 = rules [9][10];
rule137 = rules [9][11];
rule138 = rules [9][12];
rule139 = rules [9][13];
rule140 = rules [10][0];
rule141 = rules [10][1];
rule142 = rules [10][2];
rule143 = rules [10][3];
rule144 = rules [10][4];
rule145 = rules [10][5];
rule146 = rules [10][6];
rule147 = rules [10][7];
rule148 = rules [10][8];
rule149 = rules [10][9];
rule150 = rules [10][10];
rule151 = rules [10][11];

rule152 = rules [10][12];
rule153 = rules [10][13];
rule154 = rules [11][0];
rule155 = rules [11][1];
rule156 = rules [11][2];
rule157 = rules [11][3];
rule158 = rules [11][4];
rule159 = rules [11][5];
rule160 = rules [11][6];
rule161 = rules [11][7];
rule162 = rules [11][8];
rule163 = rules [11][9];
rule164 = rules [11][10];
rule165 = rules [11][11];
rule166 = rules [11][12];
rule167 = rules [11][13];
rule168 = rules [12][0];
rule169 = rules [12][1];
rule170 = rules [12][2];
rule171 = rules [12][3];
rule172 = rules [12][4];
rule173 = rules [12][5];
rule174 = rules [12][6];
rule175 = rules [12][7];
rule176 = rules [12][8];
rule177 = rules [12][9];
rule178 = rules [12][10];
rule179 = rules [12][11];
rule180 = rules [12][12];
rule181 = rules [12][13];
rule182 = rules [13][0];
rule183 = rules [13][1];
rule184 = rules [13][2];
rule185 = rules [13][3];
rule186 = rules [13][4];
rule187 = rules [13][5];
rule188 = rules [13][6];
rule189 = rules [13][7];
rule190 = rules [13][8];
rule191 = rules [13][9];

A-14

```
rule192 = rules [13][10];
rule193 = rules [13][11];
rule194 = rules [13][12];
rule195 = rules [13][13];
}
void Defuzzifikasi(){
    RuleEvaluation();
    float tujuh = 7;
    float sebelas = 11;
    float tigapuluh = 30;
    float tigadua = 32;
    firing=0;
    defuzzy = (rule0 * tigadua) + (rule1 * tigadua) + (rule2 * tigadua) +
(rule3 * tigadua) + (rule4 * tigadua) + (rule5 * tigapuluh) + (rule6 *
tigapuluh) + (rule7 * tigadua) + (rule8 * tigadua) + (rule9 * tigadua) +
(rule10 * tigadua) + (rule11 * tigadua) + (rule12 * tigapuluh) + (rule13 *
tigapuluh) + (rule14 * tigadua) + (rule15 * tigadua) + (rule16 * tigadua)
+ (rule17 * tigapuluh) + (rule18 * tigapuluh) + (rule19 * tigapuluh) +
(rule20 * sebelas) + (rule21 * tigadua) + (rule22 * tigadua) + (rule23 *
tigadua) + (rule24 * tigapuluh) + (rule25 * tigapuluh) + (rule26 *
tigapuluh) + (rule27 * sebelas) + (rule28 * tigapuluh) + (rule29 *
tigapuluh) + (rule30 * tigapuluh) + (rule31 * tigapuluh) + (rule32 *
tigapuluh) + (rule33 * tigapuluh) + (rule34 * sebelas) + (rule35 *
tigapuluh) + (rule36 * tigapuluh) + (rule37 * tigapuluh) + (rule38 *
tigapuluh) + (rule39 * tigapuluh) + (rule40 * tigapuluh) + (rule41 *
sebelas) + (rule42 * sebelas) + (rule43 * sebelas) + (rule44 * tujuh) +
(rule45 * tujuh) + (rule46 * tujuh) + (rule47 * sebelas) + (rule48 * sebelas)
+ (rule49 * sebelas) + (rule50 * sebelas) + (rule51 * tujuh) + (rule52 *
tujuh) + (rule53 * tujuh) + (rule54 * sebelas) + (rule55 * sebelas) + (rule56
* sebelas) + (rule57 * sebelas) + (rule58 * sebelas) + (rule59 * tujuh) +
(rule60 * tujuh) + (rule61 * sebelas) + (rule62 * sebelas) + (rule63 *
sebelas) + (rule64 * sebelas) + (rule65 * sebelas) + (rule66 * tujuh) +
(rule67 * tujuh) + (rule68 * sebelas) + (rule69 * sebelas) + (rule70 *
sebelas) + (rule71 * tujuh) + (rule72 * tujuh) + (rule73 * tujuh) + (rule74
* tujuh) + (rule75 * sebelas) + (rule76 * sebelas) + (rule77 * sebelas) +
(rule78 * tujuh) + (rule79 * tujuh) + (rule80 * tujuh) + (rule81 * tujuh) +
(rule82 * sebelas) + (rule83 * sebelas) + (rule84 * sebelas) + (rule85 *
tujuh) + (rule86 * tujuh) + (rule87 * tujuh) + (rule88 * tujuh) + (rule89 *
tujuh) + (rule90 * sebelas) + (rule91 * sebelas) + (rule92 * tujuh) + (rule93
* tujuh) + (rule94 * tujuh) + (rule95 * tujuh) + (rule96 * tujuh) + (rule97
```

```

* sebelas) + (rule98 * tigadua) + (rule99 * tigadua) + (rule100 * tigadua)
+ (rule101 * tigadua) + (rule102 * tigadua) + (rule103 * tigapuluh) +
(rule104 * tigapuluh) + (rule105 * tigadua) + (rule106 * tigadua) +
(rule107 * tigadua) + (rule108 * tigadua) + (rule109 * tigadua) + (rule110
* tigapuluh) + (rule111 * tigapuluh) + (rule112 * tigadua) + (rule113 *
tigadua) + (rule114 * tigadua) + (rule115 * tigapuluh) + (rule116 *
tigapuluh)+ (rule117 * tigapuluh) + (rule118 * sebelas) + (rule119 *
tigadua) + (rule120 * tigadua) + (rule121 * tigadua) + (rule122 *
tigapuluh) + (rule123 * tigapuluh) + (rule124 * tigapuluh) + (rule125 *
sebelas)+ (rule126 * tigapuluh) + (rule127 * tigapuluh) + (rule128 *
tigapuluh) + (rule129 * tigapuluh) + (rule130 * tigapuluh) + (rule131 *
tigapuluh) + (rule132 * sebelas) + (rule133 * tigapuluh) + (rule134 *
tigapuluh)+ (rule135 * tigapuluh) + (rule136 * tigapuluh) + (rule137 *
tigapuluh) + (rule138 * tigapuluh) + (rule139 * sebelas) + (rule140 *
sebelas) + (rule141 * sebelas) + (rule142 * tujuh) + (rule143 * tujuh)+
(rule144 * tujuh) + (rule145 * sebelas) + (rule146 * sebelas) + (rule147 *
sebelas) + (rule148 * sebelas) + (rule149 * tujuh) + (rule150 * tujuh) +
(rule151 * tujuh) + (rule152 * sebelas) + (rule153 * sebelas) + (rule154 *
sebelas) + (rule155 * sebelas) + (rule156 * sebelas) + (rule157 * tujuh) +
(rule158 * tujuh) + (rule159 * sebelas) + (rule160 * sebelas) + (rule161 *
sebelas) + (rule162 * sebelas) + (rule163 * sebelas) + (rule164 * tujuh) +
(rule165 * tujuh)+ (rule166 * sebelas) + (rule167 * sebelas) + (rule168 *
sebelas) + (rule169 * tujuh) + (rule170 * tujuh) + (rule171 * tujuh) +
(rule172 * tujuh) + (rule173 * sebelas) + (rule174 * sebelas)+ (rule175 *
sebelas) + (rule176 * tujuh) + (rule177 * tujuh) + (rule178 * tujuh) +
(rule179 * tujuh) + (rule180 * sebelas) + (rule181 * sebelas) + (rule182 *
sebelas) + (rule183 * tujuh)+ (rule184 * tujuh) + (rule185 * tujuh) +
(rule186 * tujuh) + (rule187 * tujuh) + (rule188 * sebelas) + (rule189 *
sebelas) + (rule190 * tujuh) + (rule191 * tujuh) + (rule192 * tujuh)+
(rule193 * tujuh) + (rule194 * tujuh)+ (rule195 * sebelas) ;
int i, j;
for ( j=0; j<=13; j=j+1)
{ for ( i=0; i<=13; i=i+1)
  { firing = firing + rules [i][j];
  }
}
defuzzy = defuzzy / firing;
}
void loop()
{

```

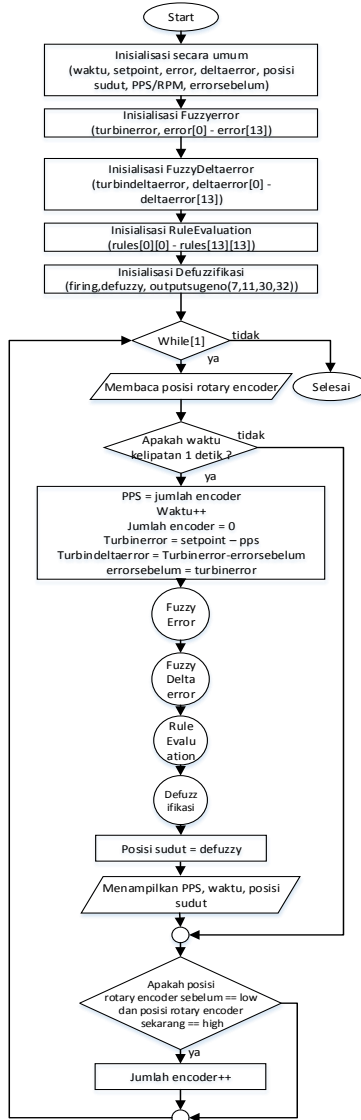
```

    n = digitalRead(encoder0PinA);
    if (millis() - lastmillis == 1000){ //Uptade every one second,
    pps = encoder0Pos; // Convert frecuency to RPM or PPS
    Serial.print("PPS =\t"); //print the word "PPS" and tab.
    Serial.print(pps); // print the pps value.
    waktu++;
    Serial.print("\t Waktu=\t"); //print the word "Waktu".
    Serial.print(waktu);
    encoder0Pos = 0; // Restart the PPS counter
    lastmillis = millis(); // Uptade lasmillis
//pengujian tracking
// if (waktu >= 200)
// {sp = 10;}
// if (waktu >= 400)
// {sp = 25;}
    turbinerror=sp-pps;
    turbindeltaerror=turbinerror-errorsebelum;
    FuzzyError();
    FuzzyDeltaerror();
    RuleEvaluation();
    Defuzzifikasi();
    pos=defuzzy;
    errorsebelum=turbinerror;
    Serial.print("\t Sudut=\t");
    Serial.println(pos);
    }
    if ((encoder0PinALast == LOW) && (n == HIGH)) {
    encoder0Pos++;
    }
    encoder0PinALast = n;
    myservo.write(pos);
    myservo1.write(pos);
    myservo2.write(pos);
}

```

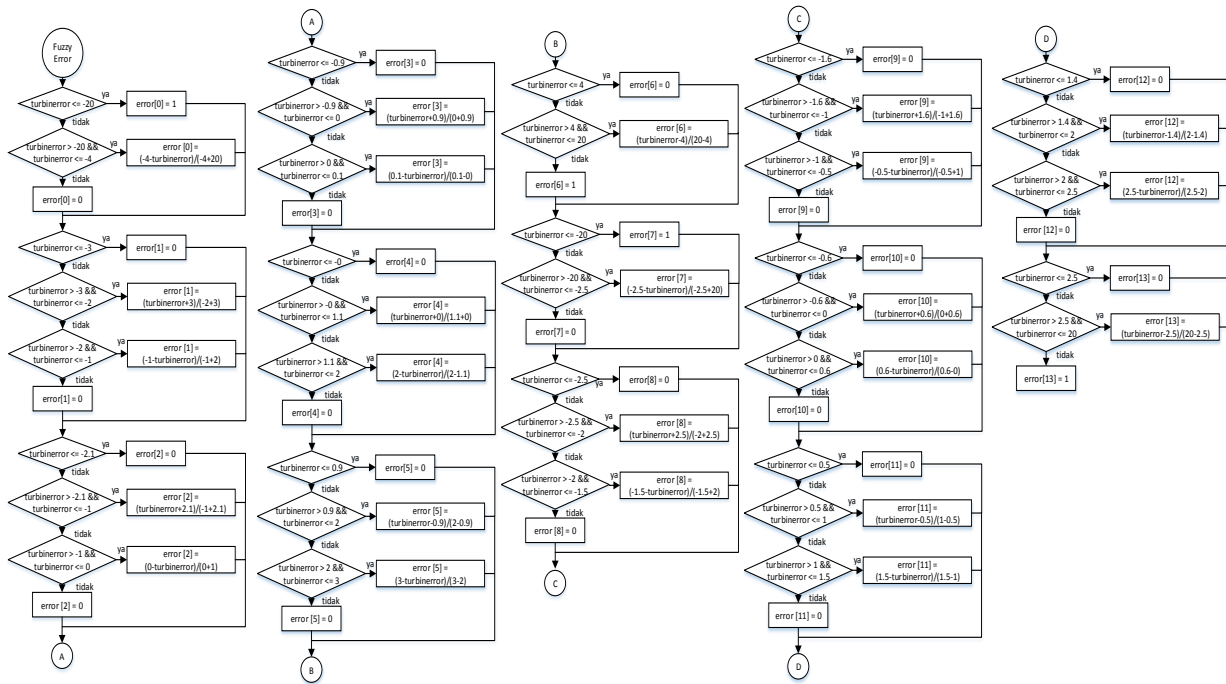

LAMPIRAN B

Flowchart Program Arduino

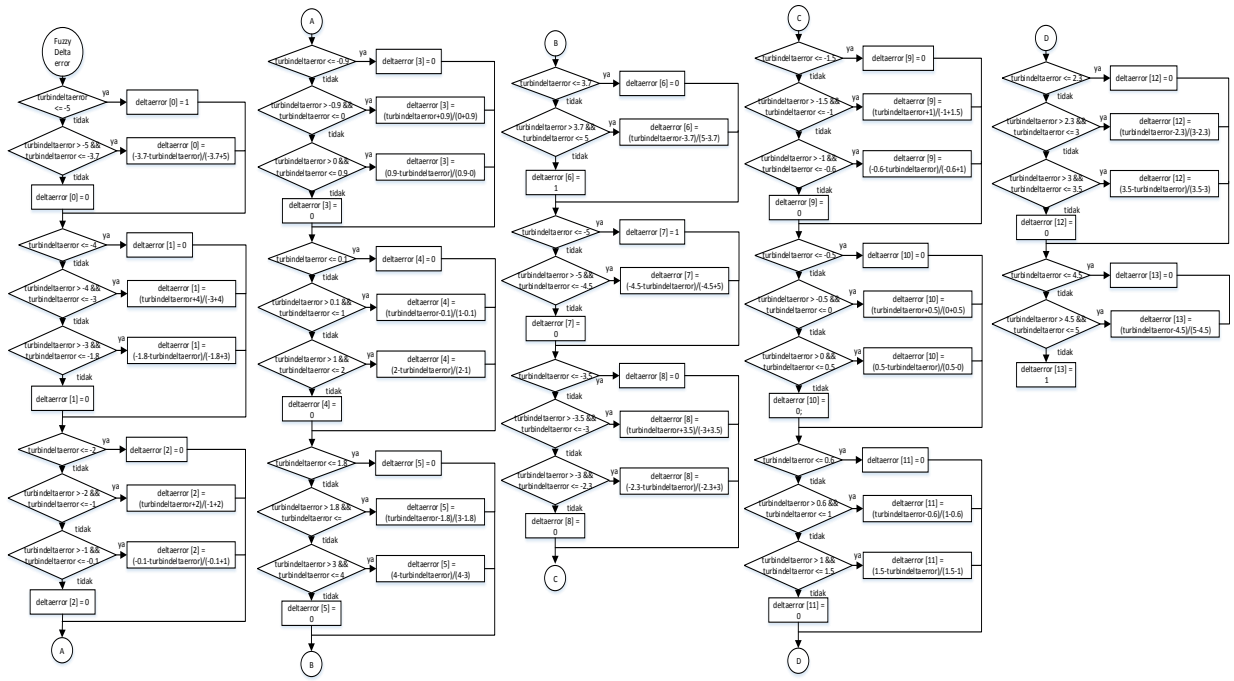


Gambar B-1. Flowchart program arduino keseluruhan

B-2

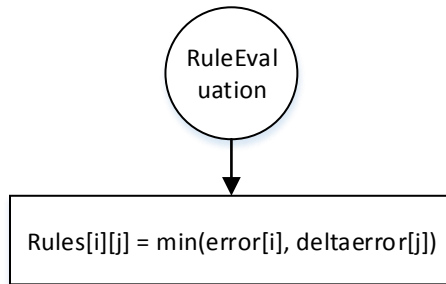


Gambar B-2. Flowchart program arduino fuzzzyerror

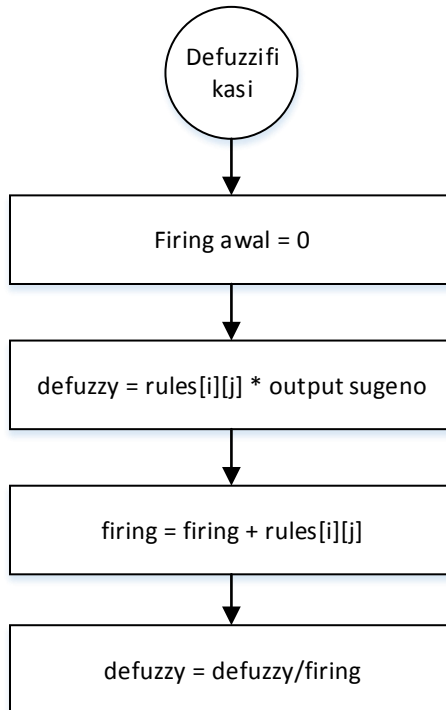


Gambar B-3. Flowchart program arduino fuzzydeltairror

B-4



Gambar B-4. Flowchart program arduino rule evaluation



Gambar B-5. Flowchart program arduino defuzzifikasi

BIODATA PENULIS



Penulis dilahirkan di Tangerang, 05 September 1993, merupakan anak kelima dari 5 bersaudara. Penulis telah menempuh pendidikan formal yaitu di SD Negeri Karawaci Baru 1 Tangerang, SMP Budi Utomo Perak Jombang, SMA Budi Utomo Perak Jombang. Setelah lulus dari SMA tahun 2011, penulis mengikuti UMPN dan diterima di Jurusan Teknik Elektro Politeknik Negeri Jakarta pada tahun 2011. Penulis lulus dari Politeknik Negeri Jakarta tahun 2014 kemudian melanjutkan pendidikan S1 dan diterima di Jurusan Teknik Fisika Fakultas Teknologi Industri - ITS pada tahun 2015 dan terdaftar dengan NRP. 2414106031. Di Jurusan Teknik Fisika ini Penulis mengambil Bidang Studi Rekayasa Instrumentasi. Penulis sempat aktif di beberapa kegiatan Seminar yang diselenggarakan oleh Jurusan, Himpunan Mahasiswa Teknik Fisika (HMTF).