



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

38753/H/10



RSF  
621.406  
Har  
r-1  
2009

FINAL PROJECT RF1483

## BUILD DESIGN OF BLADE PITCH ANGLE CONTROL SYSTEM BASED CLASSIC-FUZZY ON WIND TURBINE PROTOTYPE

**ADAM HARIKA**  
NRP 2405 100 012

Supervisor

Ir. Ronny Dwi Noriyati, MKes.

Ir. Ali Musyafa', MSc.

DEPARTEMENT OF ENGINEERING PHYSICS  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2009

PERPUSTAKAAN ITS	
Tgl. Terima	11 - 8 - 2009
Terima Dari	ff
No. Induk	733

**LEMBAR PENGESAHAN**

**RANCANG BANGUN *BLADE PITCH ANGLE CONTROL*  
SYSTEM BERBASIS *CLASSIC-FUZZY* PADA PROTOTYPE  
*WIND TURBINE***

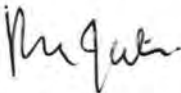
**TUGAS AKHIR**

**Oleh:**

**ADAM HARIKA  
NRP 2405 100 012**

**Surabaya, Agustus 2008  
Mengetahui/Menyetujui**

**Pembimbing I**



**Ir. RONNY D.N., MKes  
NIP 131 409 099**

**Pembimbing II**



**Ir. ALI MUSYAFA', MSc  
NIP 131 652 210**

**Ketua Jurusan  
Teknik Fisika FTI-ITS**

**DR. BAMBANG L.W., ST. MT  
NIP 132 137 895**



**LEMBAR PENGESAHAN**

**RANCANG BANGUN *BLADE PITCH ANGLE CONTROL*  
SYSTEM BERBASIS *CLASSIC-FUZZY* PADA PROTOTYPE  
*WIND TURBINE***

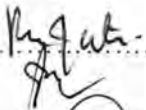


**TUGAS AKHIR**

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

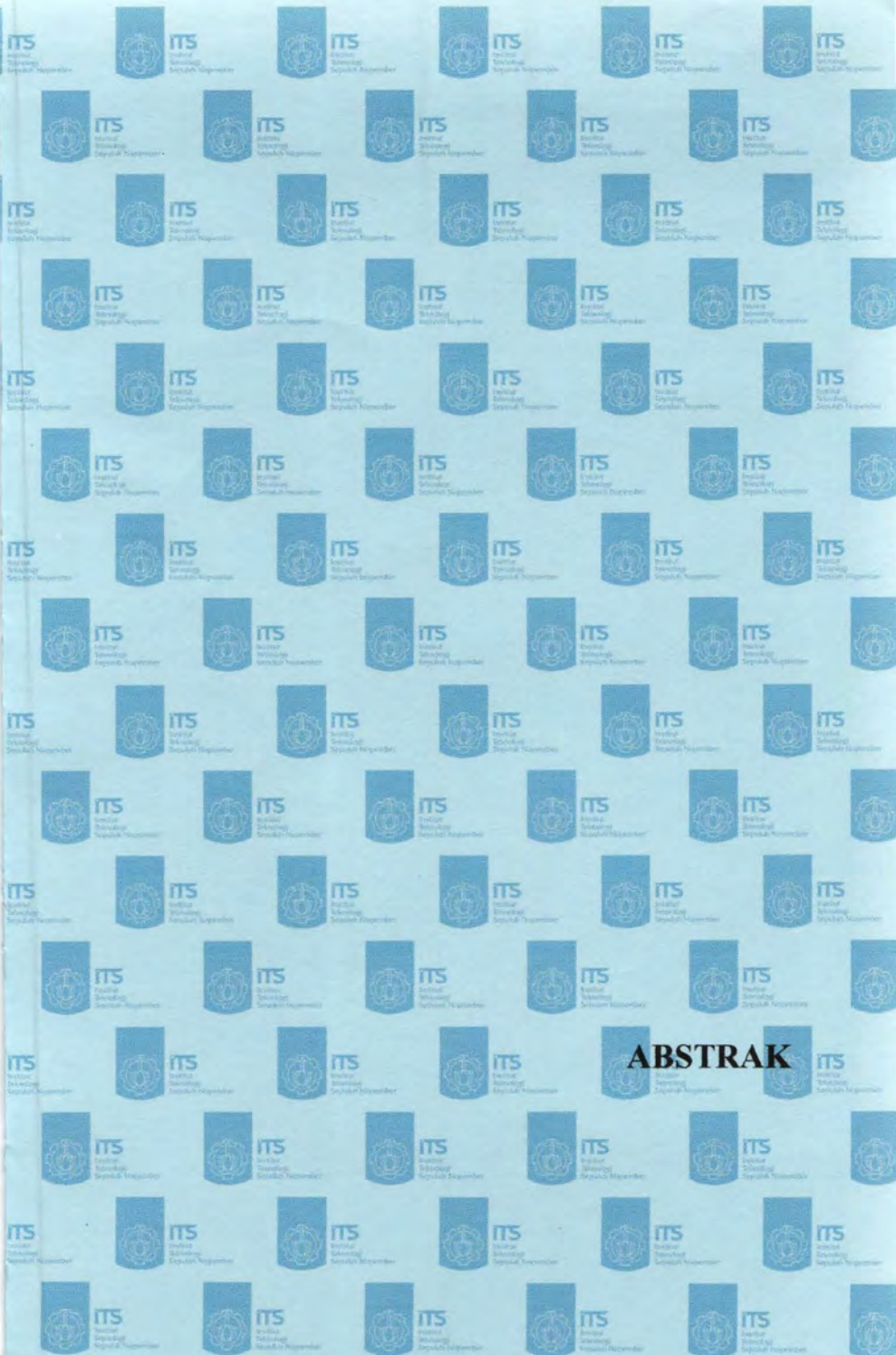
Oleh:

**ADAM HARIKA**  
**NRP 2405 100 012**

Disetujui oleh Tim Penguji Tugas Akhir:

1. Ir. Ronny D.N., Mkes. ....  ..... (Pembimbing I)
2. Ir. Ali Musyafa', MSc .....  ..... (Pembimbing II)
3. Ir. Apriani Kusumawardhani, MSc. ....  ..... (Penguji I)
4. Ridho Hantoro, ST, MT. ....  ..... (Penguji II)

**SURABAYA**  
**AGUSTUS, 2009**



**ABSTRAK**

# **RANCANG BANGUN *BLADE PITCH ANGLE CONTROL SYSTEM* BERBASIS *CLASSIC-FUZZY* PADA PROTOTIPE *WIND TURBINE***

**Nama** : Adam Harika  
**NRP** : 2405 100 012  
**Jurusan** : Teknik Fisika FTI-ITS  
**Dosen Pembimbing** : Ir. Ronny D.N., Mkes  
Ir. Ali Musyafa', MSc

## **ABSTRAK**

Desain *wind turbine* konvensional tanpa adanya sistem kontrol dapat menyebabkan kerja *wind turbine* kurang optimal. Hal ini karena energi yang ditangkap sangat bergantung pada naik turunnya kecepatan angin. Dalam tugas akhir ini dirancang sebuah sistem kontrol pada *wind turbine* dengan metode *blade pitch angle control system* yang mengadopsi sistem kontrol berbasis *classic-fuzzy*. Tujuannya adalah untuk menjaga kecepatan putar *shaft* agar tetap pada range operasi generator, yang dianalogikan dengan sebuah nilai *set point* tertentu yang akan dipertahankan oleh kontroler. Variabel inputnya adalah kecepatan sudut *shaft* yang disensor menggunakan *rotary encoder*. Berdasarkan kecepatan sudut *shaft*, sistem kontrol mengendalikan sudut kemiringan dari *blade (angle of attack)*. Perubahan sudut serang ini akan secara signifikan mempengaruhi kecepatan putar *shaft*. Pada pengujiannya diterapkan dua sistem Kontrol Logika Fuzzy (KLF) dengan perbedaan fungsi keanggotaan output. Kontroler pertama dinilai gagal melakukan aksi kontrol karena nilai eror *steady state* diatas 5%. Sedangkan kontroler kedua yang merupakan pembaharuan terhadap kontroler pertama dapat bekerja dengan baik dalam mempertahankan *set point*. Ini membuktikan bahwa penerapan *blade pitch angle control system* pada wind turbin dinilai efektif.

Kata kunci : *blade pitch angle control system, wind turbine, classic-fuzzy.*



## **BUILD DESIGN OF BLADE PITCH ANGLE CONTROL SYSTEM BASED CLASSIC-FUZZY ON WIND TURBINE PROTOTYPE**

**Name** : Adam Harika  
**NRP** : 2405 100 012  
**Department** : Engineering Physics, FTI-ITS  
**Supervisor** : Ir. Ronny D.N., Mkes  
Ir. Ali Musyafa', MSc

### **ABSTRACT**

Conventional design of wind turbine without control systems can make wind turbine less optimal. This is because the energy that can be captured is depending on the fluctuation of the wind speed. At this Final Assignment a blade pitch angle control system classic-fuzzy based is designed. This control system has a purpose to maintain the shaft's rotational velocity remain at the operational range of the generator. In this case, generator range is assumed as a set point that will be maintained by the controller. Input variable in this system is rotor rotational velocity censored by rotary encoder. Based on it the blade pitch angle control system will control angle of attack of the blade relative to the wind direction. The change of the aerodynamics will significantly affect the rotor's rotational velocity. At the test of the wind turbine is used two Fuzzy Logic Controller (FLC) with difference membership function of the output data. First controller is failed to perform the control action because steady state error is above 5%. While the second one can perform a good control action in keeping respond at the set point. It is proven that blade pitch angle control system is well applied and effective for wind turbine.

**Key words** : blade pitch angle control system, wind turbine, classic-fuzzy.



## KATA PENGANTAR

## KATA PENGANTAR

Alhamdulillah, puji dan syukur senantiasa terpanjatkan kepada Allah SWT yang maha agung dan maha bijaksana. Atas berkah, petunjuk dan karunia-Nya penulis mampu untuk melaksanakan dan menyelesaikan tugas akhir yang berjudul :

### **RANCANG BANGUN *BLADE PITCH ANGLE* CONTROL SYSTEM BERBASIS *CLASSIC-FUZZY* PADA PROTOTYPE *WIND TURBINE***

Tugas akhir ini disusun guna memenuhi persyaratan bagi seorang mahasiswa untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Selama menyelesaikan tugas akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. DR. Bambang Lelono Widjiantoro, ST. MT selaku Ketua Jurusan Teknik Fisika atas segala kemudahan yang diberikan.
2. Ir. Agung Budiono, MEng selaku dosen wali yang telah memberikan segala arahan dan wejangan beliau.
3. Ir. Ronny Dwi Noriyati, MKes selaku pembimbing I atas segala kesabaran dan bimbingannya.
4. Ir. Ali Musyafa' MSc selaku pembimbing II atas segala dukungannya berupa pengetahuan, materi dan moril.
5. Ir. Ya'umar, MT selaku kalab laboratorium Rekayasa Instrumentasi atas segala saran dan kemudahan.
6. Bapak dan Ibu dosen Teknik Fisika yang telah banyak memberikan ilmunya sehingga penulis dapat menyelesaikan jenjang kuliah sampai tugas akhir ini.
7. Ibuku Sistri Hariyatini dan Ayahku Nur Kasan beserta segenap keluarga besarku, adikku Sansika A.R dan Adinda



- A.R terima kasih tiada tara atas ke-welas asihan beliau sekalian.
8. Seluruh mahasiswa Teknik Fisika, terutama angkatan Juara 2005, terima kasih atas segalanya.
  9. Para sahabat terbaikkku, terima kasih atas semua dukungan dan semangatnya.
  10. Teman-temanku seperjuangan di ruang TA Laboratorium Pengukuran Fisis, terima kasih segala bantuan dan hiburannya.
  11. Bapak Sugeng dan segenap kru bengkel las dan bubud Manukan Krajan, terima kasih atas segala bantuannya.
  12. Laboratorium Non Metal PPNS, terima kasih atas bantuannya.
  13. Laboratorium Mekanika Fluida Teknik Mesin FTI-ITS, terima kasih atas blowernya.
  14. Teman-teman kosku di Blok J/39, *thanks guys*.
  15. Citraningrum Puspaning Tyas yang selalu mencurahkan motivasi dan semangatnya padaku hingga Pulau Dewata, terima kasih telah datang disaat yang tepat.
  16. Semua pihak yang turut membantu terselesaikannya Tugas Akhir ini, terima kasih banyak.

Penulis menyadari bahwa tugas akhir ini tidaklah sempurna, tetapi penulis berharap ini dapat memberikan kontribusi yang berarti dan dapat menambah wawasan bagi pembaca dan mahasiswa Teknik Teknik Fisika yang nanti dapat digunakan sebagai referensi pengerjaan Tugas Akhir baru. Semoga awal dari permulaan yang panjang ini dapat membawa manfaat dan hikmah bagi kita semua dan juga semoga hari esok lebih baik dari hari ini.

**VIVAT TF, VIVAT SOLIDARITY, TOGETHER FOREVER**

Surabaya, 2009  
Penulis



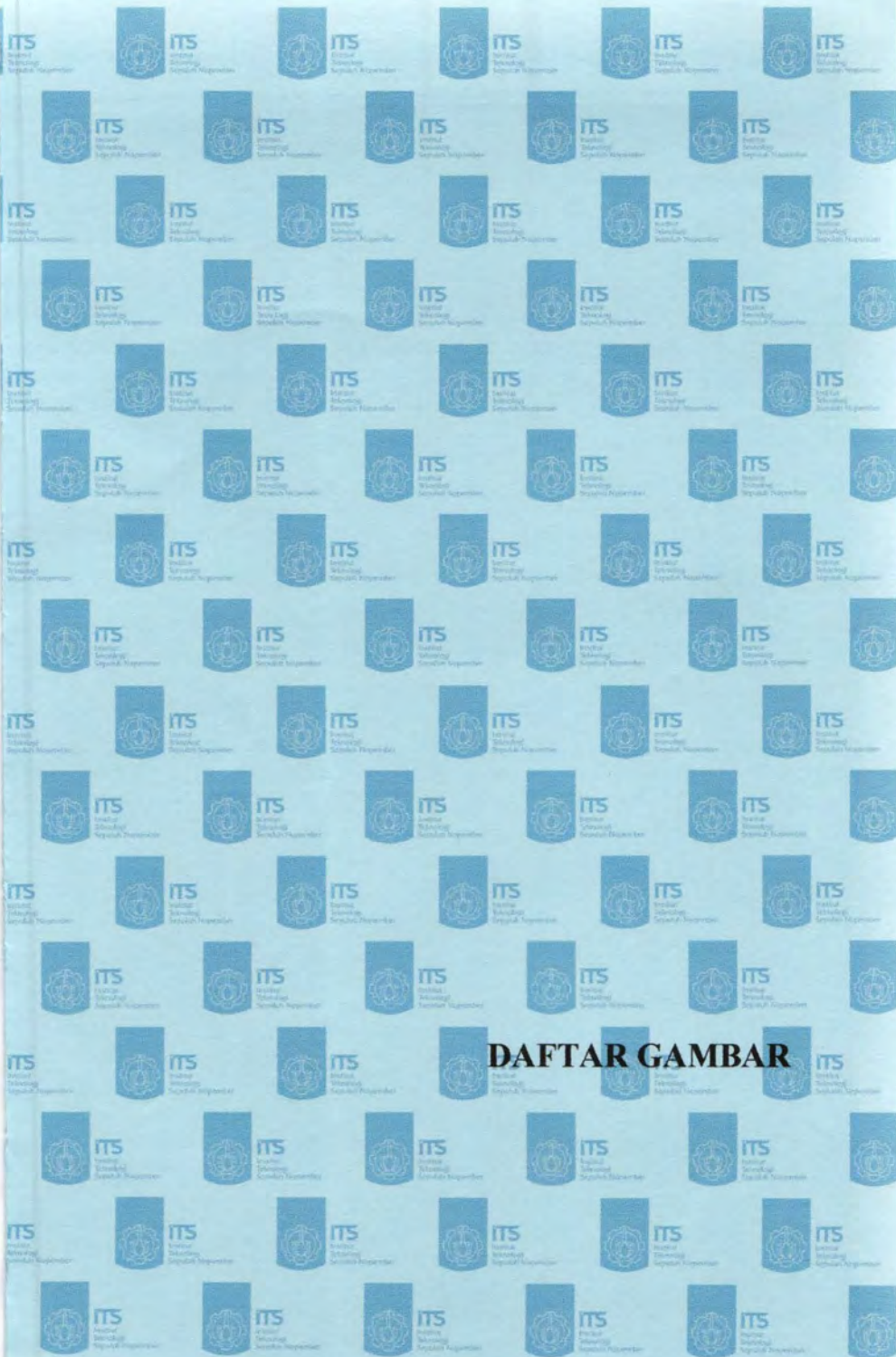
## DAFTAR ISI

## DAFTAR ISI

Halaman Judul .....	i
Lembar Pengesahan .....	v
Abstrak .....	ix
Abstract .....	xi
Kata Pengantar .....	viii
Daftar Pustaka .....	xv
Daftar Gambar .....	xvii
Daftar Grafik .....	xix
Daftar Tabel .....	xxi
 BAB I PENDAHULUAN .....	 1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Pendekatan Sistem .....	2
1.4 Tujuan .....	2
1.5 Metodologi .....	3
1.6 Sistematika Laporan .....	4
 BAB II TEORI DASAR .....	 5
2.1 Blade Pitch Angle Control System .....	5
2.2 Anemometer .....	9
2.3 Prinsip Dasar Logika Fuzzy .....	10
2.4 Mikrokontroler .....	23
2.5 RS (Recommended Standart)-232 .....	25
2.6 Motor Servo[14] .....	27
2.7 Rotary Encoder .....	32

BAB III RANCANG BANGUN PROTOTIPE WIND TURBINE DAN BLADE PITCH ANGLE CONTROL SYSTEM.....	37
3.1 Prototype Wind Turbin .....	37
3.2 Blade Pitch Angle Control System .....	42
3.3 Sensor dan Aktuator.....	52
3.4 Rotational Connector .....	53
3.5 Perancangan Software Monitoring Sederhana .....	55
3.6 Hasil Perancangan.....	55
 BAB IV PENGUJIAN DAN ANALISA DATA.....	57
4.1 Pengujian Rotating Connector .....	57
4.2 Pengujian Sensor <i>Rotary Encoder</i> .....	60
4.3 Pengujian Blade Pitch Angle Control System .....	65
4.4 Koefisien Power.....	70
 BAB V KESIMPULAN DAN SARAN .....	73
5.2 Kesimpulan .....	73
5.2 Saran.....	74
 DAFTAR PUSTAKA	
 LAMPIRAN A DATA RPM	
 LAMPIRAN B SOFTWARE MONITORING	
 LAMPIRAN C PROTOTIPE WIND TURBINE	
 LAMPIRAN D LISTING PROGRAM MIKROKONTROLER	



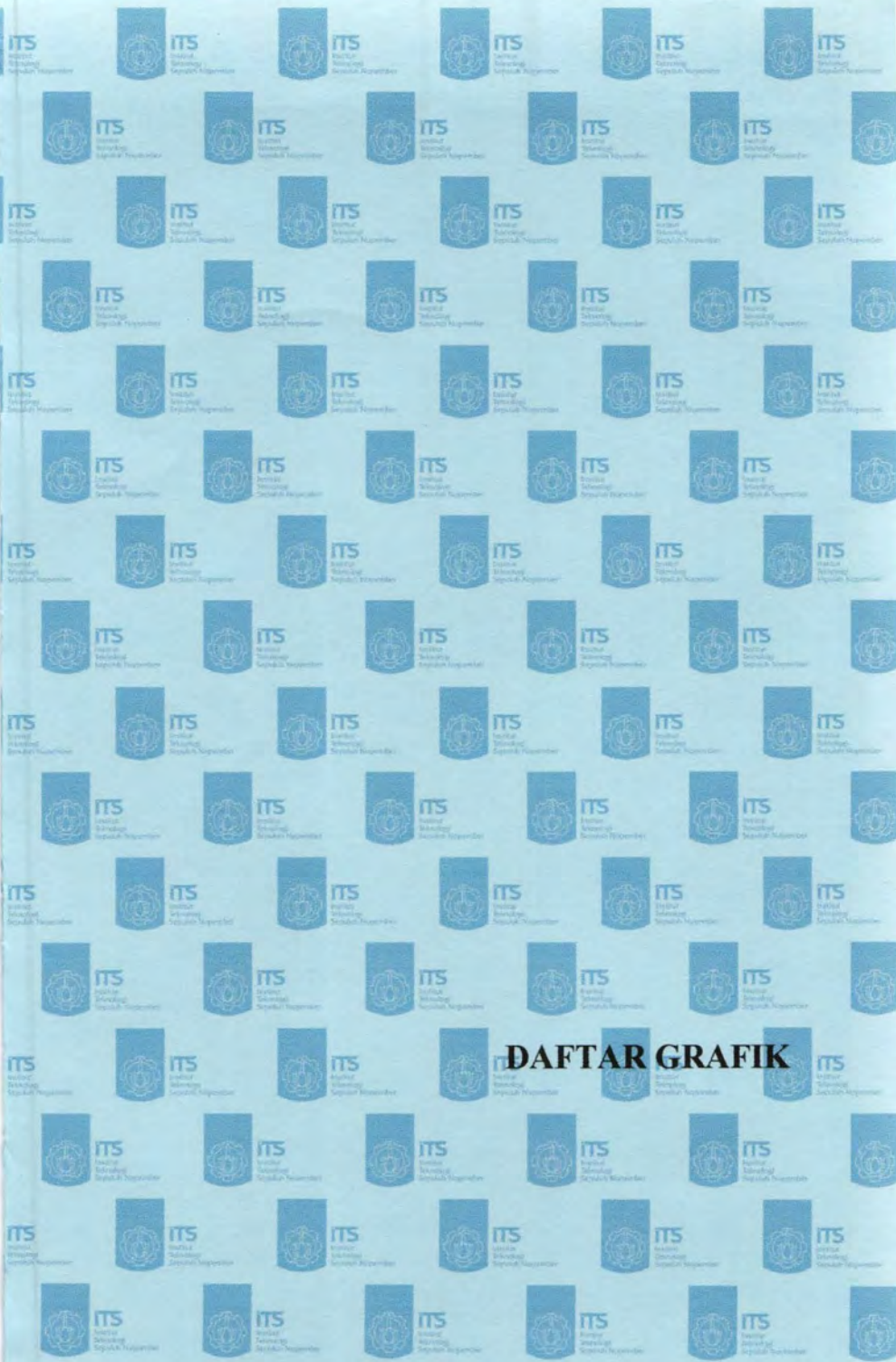


## DAFTAR GAMBAR

## DAFTAR GAMBAR

Gambar 2.1 Vektor Gaya Pada <i>Air Foil</i> .....	6
Gambar 2.2 Tube Angin yang Melewati Wind Turbine .....	6
Gambar 2.3 Blade Pitch Angle Control System .....	8
Gambar 2.4 Anemometer .....	10
Gambar 2.5 Range Kesejukan dengan .....	11
Gambar 2.6 Struktur Dasar Logika Fuzzy .....	12
Gambar 2.7 Bentuk fungsi keanggotaan Gaussian .....	14
Gambar 2.8 Bentuk Fungsi keanggotaan Segitiga .....	14
Gambar 2.9 Bentuk Fungsi keanggotaan Trapesium .....	15
Gambar 2.10 Input fuzzy dengan 3 fungsi keanggotaan .....	15
Gambar 2.11 Struktur Fuzzy Murni .....	18
Gambar 2.12 Struktur Fuzzy Mamdani .....	19
Gambar 2.13 Struktur Logika Fuzzy Takagi-Sugeno .....	20
Gambar 2.14 Fuzzy logic Control System .....	21
Gambar 2.15 Fuzzy logic Output Chart .....	22
Gambar 2.16 Konfigurasi kaki-kaki ATmega 8535 .....	24
Gambar 2.17 Rangkaian Minimum Sistem Mikrokontroler .....	25
Gambar 2.18 Serial Port .....	26
Gambar 2.19 Motor Servo .....	28
Gambar 2.20 Rangkaian Motor Servo .....	28
Gambar 2.21 Pensinyalan Motor Servo .....	30
Gambar 2.22 Contoh Posisi dan Waktu Pemberian Pulsa .....	31
Gambar 2.23 Konfigurasi Pin pada Motor Servo .....	32
Gambar 2.24 Pemilihan Servo Berdasarkan Spesifikasinya .....	32
Gambar 2.25 Rotary encoder Absolute .....	33
Gambar 2.26 Rotary encoder Relative .....	36
Gambar 3.1 Rancangan <i>Air Foil</i> Naca0012 .....	38
Gambar 3.2 Rancangan Prototipe <i>Wind Turbine</i> Depan.....	40
Gambar 3.3 Rancangan Prototipe <i>Wind Turbine</i> Atas.....	40
Gambar 3.4 Minimum Sistem pada DT-AVR .....	42
Gambar 3.5 Konfigurasi Pin DT-AVR .....	42

Gambar 3.6 K-125i USB to ISP Programmer.....	43
Gambar 3.7 <i>Flow Chart Sistem</i> .....	44
Gambar 3.8 Diagram Blok Sistem.....	44
Gambar 3.9 Fungsi Keanggotaan Error.....	47
Gambar 3.10 Fungsi Keanggotaan Delta Error.....	48
Gambar 3.11 Membership Function Data PWM.....	49
Gambar 3.12 Membership Function Data PWM 2.....	50
Gambar 3.13 Rotary Encoder.....	52
Gambar 3.14 <i>Mercotac Rotational Connector</i> .....	53
Gambar 3.15 <i>Rotational Connector Bearing</i> .....	54
Gambar 3.16 <i>Rotational Connector Bearing</i> .....	54
Gambar 3.17 <i>Flow Chart Software Monitoring</i> .....	55
Gambar 4.1 Pengamatan Tegangan Baterai Langsung .....	58
Gambar 4.2 Pengamatan Tegangan <i>Rotating Connector</i> .....	58



# DAFTAR GRAFIK



## DAFTAR GRAFIK

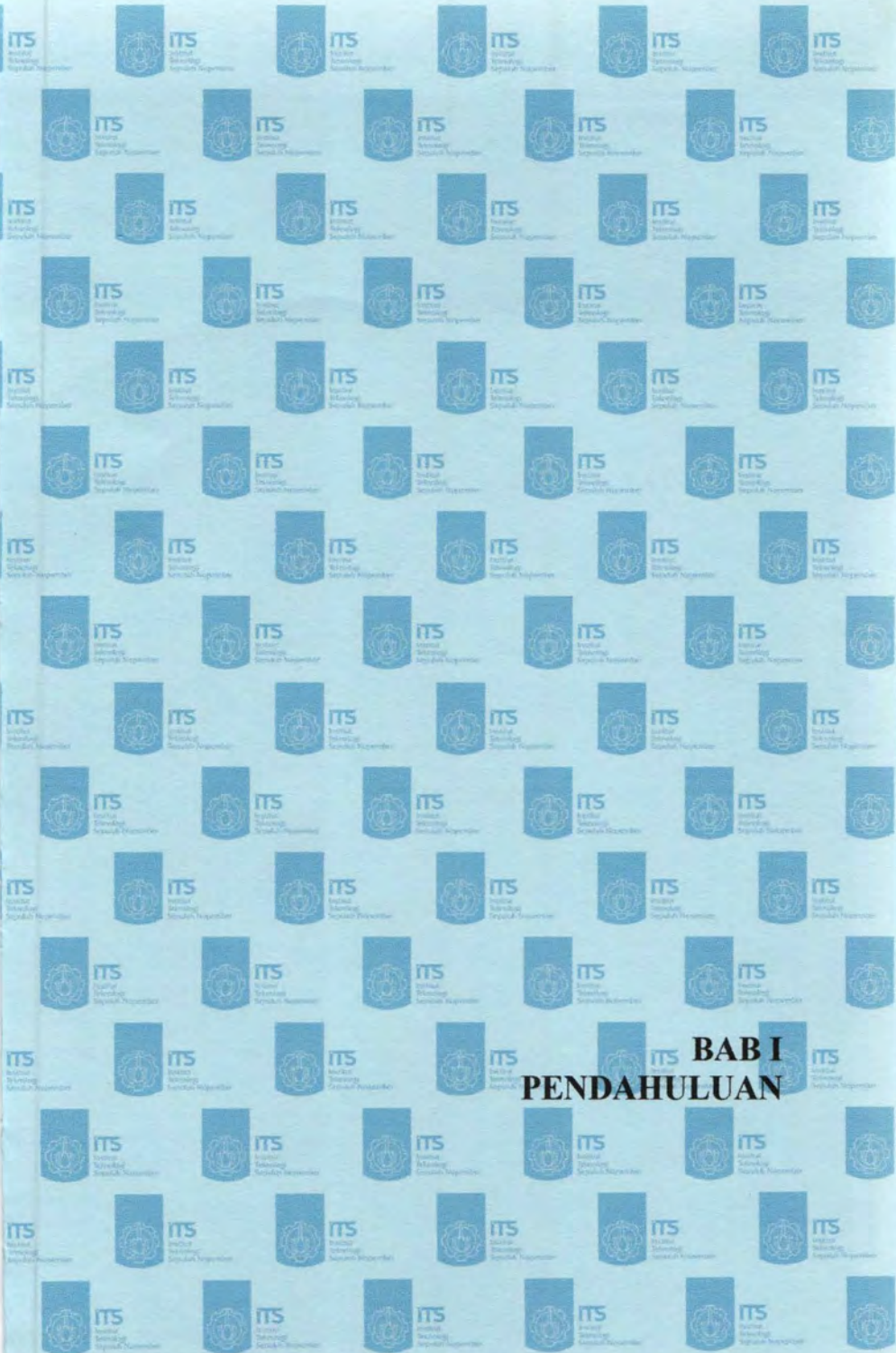
Grafik 4.1 Respon PPS Terhadap Waktu dengan KLF1 dan Penurunan Vangin 5 ke 4 m/s.....	66
Grafik 4.2 Respon PPS Terhadap Waktu dengan KLF1 dan Kenaikan Vangin 4 ke 5 m/s.....	66
Grafik 4.3 Respon PPS Terhadap Waktu dengan KLF1 dan Vangin Berfluktuatif 3.8- 5.2 m/s.....	67
Grafik 4.4 Respon PPS Terhadap Waktu dengan KLF2 dan Penurunan Vangin 5 ke 4 m/s.....	67
Grafik 4.5 Respon PPS Terhadap Waktu dengan KLF2 dan Kenaikan Vangin 4 ke 5 m/s.....	68
Grafik 4.6 Respon PPS Terhadap Waktu dengan KLF2 dan Vangin Berfluktuatif 3.8- 5.2 m/s.....	68



## DAFTAR TABEL

## DAFTAR TABEL

Tabel 2.1 Diskripsi Pin .....	24
Tabel 2.3 Kode Biner Rotary encoder Tiga Kontak .....	34
Tabel 2.4 Kode Gray Rotary encoder Tiga Kontak .....	35
Tabel 2.2 Keterangan Pin RS232 .....	27
Tabel 3.1 Spesifikasi Rancangan Wind Turbine.....	41
Tabel 3.2 Data PPS (Pulsa Per Secon) Rotary Encoder.....	45
Tabel 3.3 Rule Base Fuzzy .....	50
Tabel 4.1 Pengukuran Baterei.....	57
Tabel 4.2 Pengujian Rotating Connector .....	59
Tabel 4.3a Pengujian Rotary Encoder.....	61
Tabel 4.3b Pengujian Rotary Encoder (Lanjutan) .....	62
Tabel 4.4 Nilai Akurasi Pengujian Alat.....	64
Tabel 4.5 Karakteristik Respon.....	70
Tabel 4.6 Kecepatan Angin Sebelum dan Sesudah Blade ..	71



## BAB I PENDAHULUAN



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

*Wind turbine* adalah salah satu pembangkit energi listrik yang masih belum banyak dimanfaatkan di Indonesia. Hal ini sangat disayangkan, karena sebenarnya Indonesia memiliki beberapa tempat dengan potensi angin yang memadai. Jarangnya pemanfaatan *wind turbine* menyebabkan perkembangan desain dari *wind turbine* di Indonesia masih sangat minimum, hanya sebatas desain konvensional tanpa adanya sistem kontrol.

Hadirnya tema dari Tugas Akhir ini adalah untuk mengembangkan desain dari *wind turbine* dengan menambahkan sistem kontrol pada *wind turbine*. Tujuan utama dari sistem kontrol pada *wind turbine* adalah untuk mengontrol kecepatan sudut dari shaft penggerak rotor dari generator. Kontrol kecepatan sudut ini dibutuhkan generator untuk menghasilkan kecepatan tertentu agar dapat beroperasi secara penuh. Bila kecepatan kurang dari range operasi, maka tidak akan dihasilkan energi listrik yang cukup begitu pula bila kecepatan melebihi range operasi dari generator, maka generator akan rusak<sup>[10]</sup>. Adapun range operasi generator dalam hal ini diwakili oleh nilai set point tertentu yang akan dipertahankan oleh kontroler.

Dalam desain *wind turbine* ini akan dipasang sebuah *Blade Pitch Angle Control System*. Sistem kontrol ini nantinya akan digunakan untuk mengontrol sudut kemiringan dari *blade* (*pitch angle*). Dengan berubahnya sudut dari *blade* maka akan secara signifikan mengubah aerodinamis dari *blade*. Perubahan aerodinamis ini kemudian akan mengubah besarnya *power* angin yang ditangkap. Dari sinilah kemudian akan mengubah kecepatan dari perputaran *shaft*. Inputan dari sistem kontrol ini sendiri disuplai oleh keluaran dari sebuah rotary encoder yang akan dipasang pada *wind turbine*. Adapun sistem kontrol yang akan diadopsi adalah sistem kontrol *classic-fuzzy*.

## 1.2 Permasalahan

Adapun permasalahan yang diangkat pada penyusunan Tugas Akhir ini adalah bagaimana cara mendesain dan membangun *Blade Pitch Angle Control System* berbasis *classic-fuzzy* pada sebuah *prototype wind turbine*.

## 1.3 Pendekatan Sistem

Adapun pendekatan-pendekatan yang dilakukan pada sistem yang dirancang ini adalah :

- Perancangan dan analisa yang dilakukan adalah dari segi sistem kontrolnya, tidak pada desain aerodinamis *blade*.
- Sistem *wind turbine* yang akan dibangun merupakan sebuah model dengan ukuran mini (diameter 1 meter).
- *Range* kerja generator dianalogikan sebagai set point tertentu yang akan dipertahankan oleh kontroler (dalam hal ini 13 pps).
- Coefisien Power *blade* ditinjau pada kecepatan sudut sesuai set point dan kecepatan angin maksimum (5 m/s).
- Penentuan set point adalah nilai tengah dari kecepatan sudut minimum dan maksimum yang diperoleh dari pengambilan data awal.
- Variabel yang dikontrol berupa kecepatan sudut berdasarkan perubahan sudut *blade*.
- Mikrokontroler yang digunakan adalah AVR8535 dan bahasa pemrograman yang digunakan berupa bahasa C++ CodeVision.
- Algoritma Kontrol yang digunakan berupa Algoritma *Classic-Fuzzy* yang didefinisikan sebagai *fuzzy* dengan rule *if-then* dan membership function berupa persamaan garis linier dan metode *defuzzification Midle of Maximum (MOM)*.

## 1.4 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah untuk merancang dan membangun *Blade Pitch Angle Control System* berbasis *Classic-Fuzzy* pada sebuah *model wind turbine*.

## 1.5 Metodologi

Adapun langkah-langkah yang ditempuh dalam mengerjakan tugas akhir ini adalah sebagai berikut :

- Studi Literatur

Studi literatur dilakukan untuk memperoleh dan memahami dasar teori mengenai segala hal yang berkenaan dengan sistem yang akan dibangun. Adapaun hal-hal tersebut meliputi proses pada kontrol sudut *blade*, komponen-komponen dari *blade pitch angle control system* (meliputi rotary encoder, motor servo, mikrokontroler AT-Mega 8535 dan *rotating connector*) dan logika *classic-fuzzy*.

- Rancang Bangun *Blade Pitch Angle Control System*

Merancang sistem kontrol, plant dan *software* sekaligus melakukan survei dan memilih komponen-komponen yang tepat untuk direalisasikan dalam bentuk *hardware* sesungguhnya.

- Pengambilan Data Awal

Dilakukan tanpa menggunakan kontroler untuk mengetahui performansi awal dari plant sebagai bahan untuk merancang Kontroler Logika Fuzzy (KLF). Dari sini akan diketahui kecepatan sudut minimum dan maksimum dari *wind turbine*. Dari juga dapat ditentukan *set point* dari sistem.

- Pengujian dan Analisa

Melakukan pengujian sistem yang telah dibangun (meliputi sensor, konektor dan kontroler) menggunakan blower dan melakukan analisa terhadap hasil dan respon yang diperoleh.

- Penyusunan Laporan Tugas Akhir

Merupakan pembukuan dan pendokumentasian dari semua langkah yang telah ditempuh dalam mengerjakan tugas akhir berupa sebuah tulisan ilmiah.

## **1.6 Sistematika Laporan**

Laporan ini disusun mengikuti sistematika yang telah ditetapkan dalam pedoman penulisan laporan Tugas Akhir Institut Teknologi Sepuluh Nopember. Adapun sistematikanya adalah sebagai berikut :

### **Bab I Pendahuluan**

Berisi tentang latar belakang, permasalahan, pendekatan sistem, tujuan dan metodologi dari Tugas Akhir ini.

### **Bab II Dasar Teori**

Berisi teori-teori yang mendasari tema dari Tugas Akhir ini.

### **Bab III Rancang Bangun Prototipe Wind Turbin dan *Blade Pitch Angle Control System***

Berisi langkah-langkah perancangan dari plant dan sistem kontrol yang akan dibangun sesuai tema Tugas Akhir ini.

### **Bab IV Pengujian dan Analisa Data**

Berisi tentang hasil pengujian sistem yang telah dibangun dan analisa data.

### **Bab V Kesimpulan dan Saran**

Berisi tentang kesimpulan dari Tugas Akhir dan saran-saran perbaikan kedepan.



## BAB II DASAR TEORI

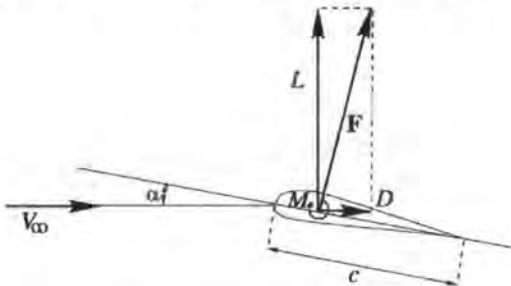
## BAB II

### DASAR TEORI

#### 2.1 Blade Pitch Angle Control System

##### 2.1.1 Prinsip Dasar

Prinsip dasar bahwa sebuah *wind turbine* dapat berputar pada porosnya adalah karena adanya vektor dari gaya *lift* dan gaya *drag* yang dihasilkan akibat bentuk aerodinamis dari penampang *blade* tersebut. Pada gambar 2.1 dijelaskan ketika sebuah *air foil* terkena angin dari arah depan, maka akan menghasilkan vektor gaya *lift* ( $L$ ) dan *drag* ( $D$ ). Gaya *lift* dan gaya *drag* ini perubahannya dipengaruhi langsung oleh bentuk geometri *blade*, kecepatan dan arah angin terhadap garis utama *blade*. Akibat dari perubahan gaya *lift* dan *drag*, maka kecepatan sudut dan torsi poros akan berubah pula. *Blade Pitch Angle Control System* adalah salah satu mekanisme kontrol pada *wind turbine* yang bekerja dengan mengontrol aerodinamis dari *blade* melalui kontrol kemiringan sudut *blade* terhadap arah tiupan angin (*angle of attack*) seperti tampak pada gambar 2.1. Perubahan sudut *blade* ini akan mempengaruhi kecepatan sudut (RPM) dari *shaft* karena adanya perubahan jumlah daya tiup angin yang diterima oleh *blade* yang dikonversi menjadi kecepatan putar *shaft*.



Gambar 2.1 Vektor Gaya Pada *Air Foil* dengan *Angle of Attack* Berbeda <sup>[15]</sup>

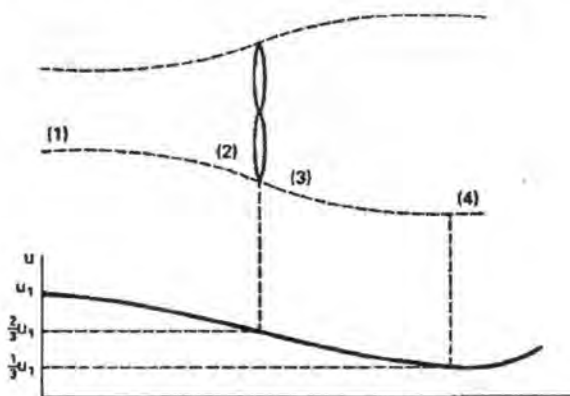
Daya dari angin yang dapat ditangkap oleh sebuah *horizontal axis wind turbine* (HAWT) dapat diturunkan dari persamaan energi kinetik angin yang bergerak dengan kecepatan tertentu kearah x. adapun persamaan energi yang menabrak *wind turbine* adalah sebagai berikut<sup>[3]</sup>:

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

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

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

Kemudian untuk mengetahui besarnya daya yang dapat diekstrak oleh wind turbin dapat dilakukan dengan menghitung selisih daya angin sebelum dan sesudah menabrak *wind turbine*.



Gambar 2.2 Tube Angin yang Melewati *Wind Turbine* <sup>[3]</sup>

Bila ada tube angin yang bergerak dengan kecepatan  $v$  akan menabrak wind turbin, maka pastinya kecepatannya akan semakin

berkurang sejalan dengan semakin dekatnya jaraknya terhadap *wind turbine*, karena tekanannya naik akibat ruang gerakannya yang semakin sempit (mampat). Ketika menabrak *wind turbine*, maka energi kinetik angin tersebut diubah oleh *wind turbine* menjadi energi rotasional. Dan setelah jarak tertentu, maka kecepatan angin akan kembali seperti semula akibat ruang gerakannya telah melebar dan mendapat energi dari udara disekitarnya. Menurut Dwinell, J. H. (*Principles of Aerodynamics*, McGraw- Hill, New York, 1949). Persamaan untuk muka tube angin yang melewati *wind turbine* ideal adalah sebagai berikut :

$$v_2 = v_3 = 2/3 v_1$$

$$v_4 = 1/3 v_1$$

$$A_2 = A_3 = 3/2 A_1$$

$$A_4 = 3 A_1$$

Jadi daya angin yang diekstrak adalah :

$$P = P_1 - P_4 = \frac{1}{2} \rho_{air} (A_1 v_1^3 - A_4 v_4^3) = \frac{1}{2} \rho_{air} \left( \frac{8}{9} A_1 v_1^3 \right) \dots (2.3)$$

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

$$P = \frac{1}{2} \rho_{air} \left[ \frac{8}{9} \left( \frac{2}{3} A_2 \right) v_1^3 \right] = \frac{1}{2} \rho_{air} \left( \frac{16}{27} A_2 v_1^3 \right) \dots (2.4)$$

Dimana :

$P$  = Daya *wind turbine* (watt)

$\rho_{air}$  = Massa jenis udara ( $\text{kg/m}^3$ )

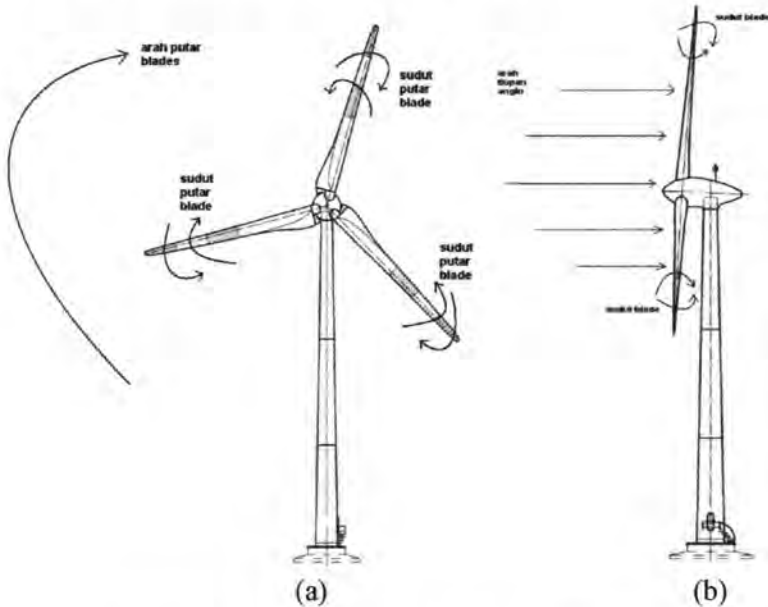
$V$  = Kecepatan angin (m/s)

$A_r$  = Luas sapuan penampang *blade* ( $\text{m}^2$ )

$C_p$  = *Power coefisien*



$16/27 = 0.59$  disebut koefisien Betz yang menyatakan efisiensi maksimum dari sebuah *wind turbine* atau koefisien power ( $C_p$ ). Tapi pada kenyataannya,  $C_p$  berkisar antara 0-45<sup>[3]</sup>.



Gambar 2.3 *Blade Pitch Angle Control System*<sup>[15]</sup>  
(a) Tampak Depan (b) Tampak Samping

Dari rumus tersebut diketahui bahwa selain pada kecepatan angin, power juga tergantung pada  $C_p$  (Coeffisien Power). Semakin besar nilai  $C_p$  maka akan semakin besar power yang dapat ditangkap oleh *wind turbine*.  $C_p$  sendiri adalah merupakan fungsi dari  $\lambda$  (*tip speed ratio*) dan  $\theta$  (*pitch angle*). Jadi persamaan 2.4 dapat ditulis kembali menjadi<sup>[15]</sup> :

$$P = \frac{1}{2} \rho_{atr} C_p(\lambda, \theta) A_r v_w^3 \quad \dots (2.5)$$

Sedangkan  $\lambda$  sendiri dirumuskan sbagai berikut :

$$\lambda = \frac{\omega R}{v} \quad \dots(2.6)$$

Dimana :

$\lambda$  = *tip speed ratio*

$\omega$  = kecepatan sudut (rps)

$v$  = kecepatan angin (m/s)

$R$  = jari-jari rotor *blade* (m)

Jika diasumsikan  $\omega$  adalah konstan sesuai *set point* yang diinginkan dan  $R$  *blade* adalah konstan, maka  $C_p$  hanya akan bergantung pada  $v$  (kecepatan angin) dan  $\theta$  (*pitch angle*), dari sinilah kemudian  $\theta$  dijadikan variabel yang dikontrol sebagai kompensasi perubahan kecepatan angin ( $v$ ) untuk mendapatkan power yang diinginkan. Sedangkan untuk mendapatkan  $\theta$  sesuai dengan yang dibutuhkan dilakukan pengambilan data dengan menggunakan kecepatan angin ( $v$ ) yang ditentukan.

Hubungan antara power dengan kecepatan sudut (rpm) *shaft* adalah sebagai berikut [15] :

$$P = \tau_{aero} \omega \quad \dots(2.7)$$

Dimana :

$P$  = daya angin (watt)

$\tau_{aero}$  = Torsi aerodinamis dari rotor *blade* (Nm)

$\omega$  = kecepatan sudut rotor *blade* (rps)

Jadi daya ( $P$ ) berbanding lurus dengan kecepatan sudut rotor *blade*.

## 2.2 Anemometer

Anemometer adalah alat untuk mengukur kecepatan angin. Pada prototipe ini anemometer digunakan untuk mengetahui

kecepatan angin yang sedang bekerja pada *wind turbine*. Untuk lebih jelasnya gambar dari anemometer dapat dilihat dibawah ini :



Gambar 2.4 Anemometer <sup>[22]</sup>

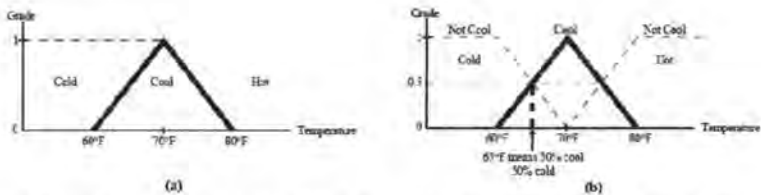
Pada Tugas Akhir ini anemometer akan digunakan untuk mengambil data kecepatan angin yang kemudian dicatat.

### 2.3 Prinsip Dasar Logika Fuzzy

Logika Fuzzy adalah salah satu cabang dari *artificial intelligent* (kecerdasan buatan) yang berhubungan dengan algoritma perasaan yang digunakan untuk menciptakan pola pikir manusia dan penentuan keputusan. Algoritma ini digunakan dalam aplikasi dimana pemrosesan data tidak dapat direpresentasikan dalam bentuk biner. Sebagai contoh, pernyataan “udara terasa sejuk” dan “lelaki itu masih muda” bukanlah pernyataan diskrit. Kedua pernyataan tersebut tidak menggambarkan data konkret tentang temperatur udara atau usia dari seseorang (suhu udara adalah 65 F atau usia lelaki itu 21 tahun).

Logika Fuzzy menginterpretasikan pernyataan samar-samar seperti ini sehingga memenuhi logika masuk akal. Dalam kasus udara sejuk, logika fuzzy memiliki kemampuan yang dapat menginterpretasikan dua hal, level kesejukan dan hubungannya dengan kehangatan untuk memastikan bahwa “sejuk” berarti suatu keadaan diantara panas dan dingin. Dalam barisan bilangan biner, panas akan bernilai suatu nilai biner (misalnya Panas adalah logika 1) dan dingin bernilai yang lain (Logika 0), hal ini

menyebabkan tidak ada nilai yang dapat menginterpretasikan “sejuk”.



Gambar 2.5 (a) *Range* Kesejukan dengan (b) Garis Putus-Putus Menunjukkan *Range* Tidak Sejuk [20]

Dalam kekontrasan logika biner, logika fuzzy dapat dikatakan logika abu-abu (*gray logic*), yang menciptakan sebuah jalan untuk menyatakan nilai “diantara” nilai-nilai data. Logika Fuzzy menghubungkan sebuah level atau strata dengan *range* data, memberikan sebuah nilai 1 pada titik maksimum dan 0 pada titik minimum. Sebagai contoh, gambar 2.5 (a) mengilustrasikan sebuah representasi dari logika fuzzy dari *range* temperatur udara “sejuk”, dimana 70 F mengindikasikan “udara sejuk sempurna” (eg. strata kesejukan bernilai 1). Semua temperatur lebih dari 80 F dinilai panas dan semua temperatur dibawah 60 F dinilai dingin. Ini berakibat bahwa temperatur diatas 80 F dan dibawah 60 F mempunyai strata kesejukan bernilai 0, berarti tidak sejuk sama sekali. Gambar 1b menunjukkan representasi lain dari *range* kesejukan, dimana garis putus-putus menunjukkan bahwa temperatur panas dan dingin berarti tidak sejuk. Pada suhu 65 F, algoritma logika fuzzy menilai bahwa temperatur tersebut adalah 50% sejuk dan 50% dingin, mengindikasikan sebuah level kesejukan. Dibawah 60 F, logika fuzzy menilai temperatur adalah dingin<sup>[20]</sup>.

Pada dasarnya struktur logika fuzzy dapat digambarkan seperti berikut :





Gambar 2.6 Struktur Dasar Logika *Fuzzy* <sup>[8]</sup>

Fungsi dari bagian-bagian di atas adalah sebagai berikut:

- **Fuzzifikasi**  
 Berfungsi untuk mentransformasikan sinyal masukan yang bersifat *crisp* ( bukan *fuzzy* ) ke himpunan *fuzzy* dengan menggunakan operator fuzzifikasi.
- **Basis Pengetahuan**  
 Berisi basis data dan aturan dasar yang mendefinisikan himpunan *fuzzy* atas daerah – daerah masukan dan keluaran dan menyusunnya dalam perangkat aturan kontrol.
- **Logika Pengambil Keputusan**  
 Merupakan inti dari logika *Fuzzy* yang mempunyai kemampuan seperti manusia dalam mengambil keputusan. Aksi atur *fuzzy* disimpulkan dengan menggunakan implikasi *fuzzy* dan mekanisme inferensi *fuzzy*.
- **Defuzzifikasi**  
 Berfungsi untuk mentransformasikan kesimpulan tentang aksi atur yang bersifat *fuzzy* menjadi sinyal sebenarnya yang bersifat *crisp* dengan menggunakan operator defuzzifikasi.

### 2.3.1 Fuzzifikasi

Fuzzifikasi merupakan suatu proses merubah variabel *non-fuzzy (crisp)* kedalam variabel *fuzzy*, variable *input (crisp)* dipetakan ke bentuk himpunan *fuzzy* sesuai dengan variasi semesta pembicaraan *input*. Pemetaan titik-titik numerik ( *crisp*

points)  $x = (x^1, x^2, \dots, x^n)^T \in U$  ke himpunan fuzzy  $A$  pada semesta pembicaraan  $U$ . Data yang telah dipetakan selanjutnya dikonversikan ke dalam bentuk *linguistik* yang sesuai dengan label dari himpunan fuzzy yang telah terdefinisi untuk variabel input sistem<sup>[8]</sup>.

Didalam pemetaan ini terdapat dua kemungkinan pemetaan yaitu :

- Fuzzyfikasi *singleton* :  $A$  adalah *fuzzy singleton* dengan support  $x$ , artinya:

$$\mu_A(x) = \begin{cases} 1 & \text{untuk } x = x_0 \\ 0 & \text{untuk } x \in U \text{ yang lain} \end{cases} \quad \dots(2.8)$$

- Fuzzyfikasi *nonsingleton*:  $\mu_A(x) = 1$  dan  $\mu_A(x')$  menurun dari 1 sebagaimana  $x'$  bergerak menjauh dari  $x$ . Sebagai contoh :

$$\mu_A(x) = \exp\left(-\frac{(x' - x)^T (x' - x)}{\sigma^2}\right) \quad \dots(2.9)$$

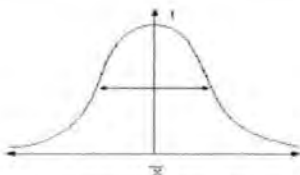
Fuzzifikasi memiliki dua komponen yang utama, yaitu :

- Fungsi keanggotaan himpunan fuzzy.  
Fungsi keanggotaan merupakan sebuah kurva yang menggambarkan pemetaan dari input ke derajat keanggotaan antara 0 dan 1. Melalui fungsi keanggotaan yang telah disusun maka dari nilai-nilai masukan tersebut menjadi informasi fuzzy yang berguna nantinya untuk proses pengolahan secara fuzzy pula. Banyaknya jumlah fungsi keanggotaan dalam *fuzzy set* menentukan banyaknya aturan yang harus dibuat<sup>[8]</sup>.

Keanggotaan dalam himpunan fuzzy mempunyai bentuk yang berbeda-beda terdiri dari :

➤ Fungsi Gaussian

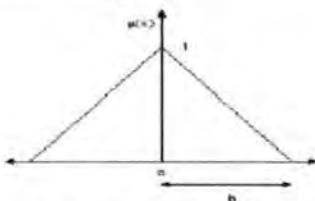
$$\mu_A(u) = \exp\left[-\frac{(u-x)^2}{2\sigma^2}\right] \quad \dots(2.10)$$



Gambar 2.7 Bentuk fungsi keanggotaan Gaussian <sup>[8]</sup>

➤ Fungsi Segitiga

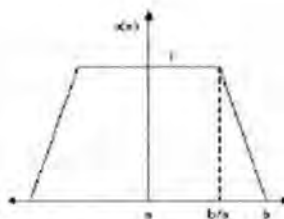
$$\mu_A(u) = 1 - \frac{\sqrt{(u-a)^2}}{b} \quad \dots(2.11)$$



Gambar 2.8 Bentuk Fungsi keanggotaan Segitiga <sup>[8]</sup>

➤ Trapesium

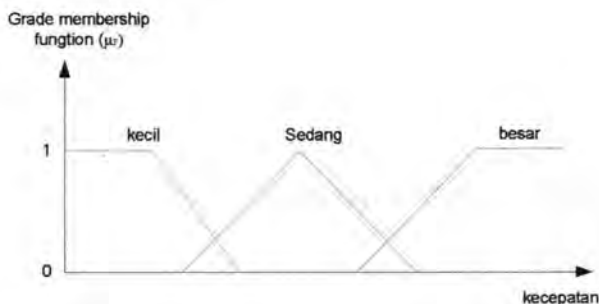
$$\mu_A(u) = \begin{cases} 1 & \dots; 0 \geq (u-a) \leq \frac{b}{a} \\ 2 - 2 \frac{\sqrt{(u-a)^2}}{b} & \dots; \frac{b}{2} \leq (u-a) \leq b \end{cases} \quad \dots(2.12)$$



Gambar 2.9 Bentuk Fungsi keanggotaan Trapesium<sup>[8]</sup>

▪ Label.

Didalam *Fuzzy set* tentunya memiliki beberapa fungsi keanggotaan, jumlah dari keanggotaan inipun disesuaikan dengan banyaknya kebutuhan. Setiap fungsi keanggotaan dapat didefinisikan dengan label atau nama. Dapat dinyatakan dengan besar, sedang, kecil atau sesuai dengan keinginan<sup>[8]</sup>.



Gambar 2.10 Input fuzzy dengan 3 fungsi keanggotaan<sup>[8]</sup>

### 2.3.2 Rule Base

Basis pengetahuan terdiri dari fakta (*Data Base*), dan kaidah atur (*Rule Base*). Fakta merupakan bagian pengetahuan yang memuat informasi tentang objek, peristiwa, atau situasi. Fakta umumnya menyatakan kondisi statik dari suatu objek. Sedangkan kaidah (*Rule base*) berisi informasi tentang cara membangkitkan fakta baru atau hipotesa fakta yang sudah ada<sup>[8]</sup>.



- Basis Data (*Data Base*).

Basis data berfungsi untuk mendefinisikan himpunan-himpunan *fuzzy* dari sinyal masukan dan sinyal keluaran agar dapat digunakan oleh variabel linguistik dalam basis aturan. Dalam pendefinisian tersebut biasanya dilakukan secara *subjektif* dengan menggunakan pendekatan heuristik dan didasarkan pada pengalaman dan pertimbangan yang menyangkut kerekayasaan, sehingga bergantung penuh pada perancang.

- Kaidah Atur (*Rule Base*).

Kaidah atur dalam *fuzzy* ini biasanya tersusun dengan pernyataan :

**IF** (*antecedent*) **THEN** (*consequent*) atau dapat juga

**IF**  $x$  is  $A$  **THEN**  $y$  is  $B$ .

*Antecedent* : berisi himpunan fakta *input* (sebab).

*Consequent* : berisi himpunan fakta *output* (akibat).

**IF ... THEN ...** dalam logika *fuzzy* akan melakukan pemetaan dari himpunan *fuzzy input* ke himpunan *fuzzy output*<sup>[8]</sup>.

### 2.3.3 Logika Pengambil Keputusan

Sering pula disebut sebagai *Fuzzy Inference system* (FIS) merupakan bagian terpenting dalam logika *fuzzy*. Langkah yang dilakukan pada tahap ini yaitu mengevaluasi aturan, dimana mengevaluasi aturan mempunyai arti yaitu logika *fuzzy* mengolah dan menyimpulkan proses yang tersusun dari rule *IF...THEN*, setiap *rule* menghasilkan satu *output*. Pada dasarnya satu *rule* akan aktif apabila kondisi *input* memenuhi aturan pernyataan *IF*. Pengaktifan aturan pernyataan *IF* menghasilkan *output* kontrol yang didasarkan pada aturan pernyataan *THEN*. Dalam sistem *fuzzy* digunakan banyak *rule* yang menyatakan satu atau lebih pernyataan *IF*. Suatu *rule* dapat pula mempunyai beberapa kondisi *input*, yang satu sama lainnya dihubungkan dengan *AND* atau *OR* untuk mendapatkan *rule output*<sup>[8]</sup>.

### 2.3.4 Defuzzifikasi

Defuzzifikasi merupakan proses merubah *output fuzzy* dari FIS (*fuzzy inference system*) menjadi *output crisp*. Bentuk umum proses defuzzifikasi dinyatakan dengan:

$$Z_0 = \text{defuzzier}(z) \quad \dots(2.13)$$

dimana  $z$  adalah aksi pengendalian *fuzzy*,  $Z_0$  adalah aksi pengendali *crisp*, dan *defuzzifier* adalah operator defuzzifikasi.

Terdapat beberapa macam metode defuzzifikasi, yaitu :

- Metode Titik Pusat (*Center Of Area, COA*).

Metode ini membagi dua momen pertama fungsi keanggotaan, dan harga  $v_0$  yang menandai garis pembagi adalah harga  $V$  yang ter-defuzzifikasi.

Secara algoritmik dinyatakan [8]:

$$v_0 = \frac{\int v \mu_v(v) dv}{\int \mu_v(v) dv} \quad \dots(2.14)$$

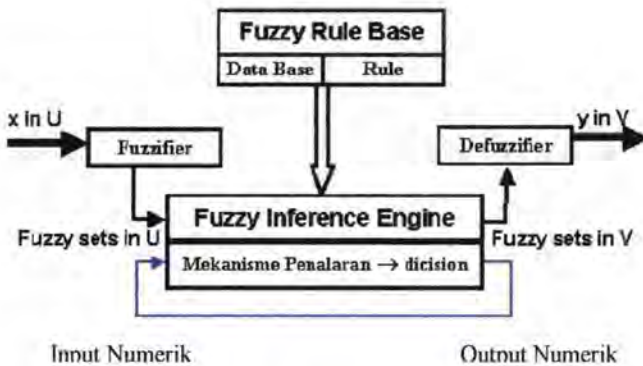
Sedangkan dalam semesta diskrit dapat dinyatakan :

$$v_0 = \frac{\sum_{k=1}^m v_k \mu_v(v_k)}{\sum_{k=1}^m \mu_v(v_k)} \quad \dots(2.15)$$

- Metode Rata-rata Maksimum (*Midle Of Maximum, MOM*).  
Merupakan metode defuzzifikasi yang merepresentasikan nilai tengah dari keluaran yang fungsi anggotanya maximum. Fungsinya ditunjukkan sebagai:

$$z_0 = \frac{\sum_{i=1}^n z_i}{l} \quad \dots(2.16)$$





Gambar 2.12 Struktur Fuzzy Mamdani<sup>[8]</sup>

▪ Logika Fuzzy Takagi-Sugeno

Merupakan logika *fuzzy* dimana *input*-nya berupa Linguistik dan *output*-nya Numerik. Kaidah *fuzzy* Takagi-Sugeno:

Kaidah *fuzzy* dari *type* ini dapat dinyatakan :

$$L^{(l)} : \text{IF } x_1 \text{ is } F_1^{-1} \text{ and } \dots \text{ and } x_n \text{ is } F_n^{-1}, \text{ THEN } Y^l = c_0^{-1} + c_1^{-1}x_1 + c_n^{-1}x_n \quad \dots(2.16)$$

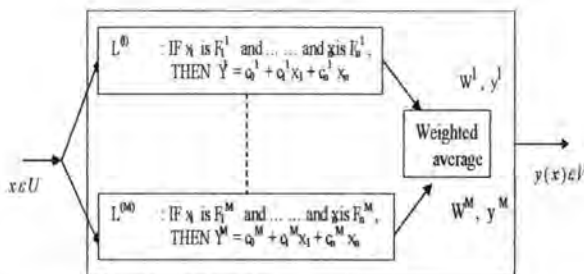
Dimana  $F_1^{-1}$  adalah *set fuzzy*,  $c_1$  adalah parameter berharga riil,  $y^l$  keluaran sistem oleh kaidah  $L^{(l)}$  dimana  $l = 1, 2, \dots, M$  adalah jumlah kaidah fuzzy.

Dari kaidah tersebut diatas dapat dilihat bahwa, bagian **IF** berupa *set fuzzy*, sedangkan pada bagian **THEN** adalah bernilai *crisp*, berupa kombinasi linier dari variabel masukan. Masukan *vector* berharga riil  $\underline{x} = (x_1, \dots, x_n)^T$ , dan keluarannya adalah  $y(\underline{x})$  yang bernilai sesuai dengan *weighted average*  $y^l$  sebagai berikut<sup>[8]</sup>:

$$y(x) = \frac{\sum_{l=1}^M w^l y^l}{\sum_{l=1}^M w^l} \quad \dots(2.17)$$

Sedangkan rumus *weighing*  $w^l$  dirumuskan :

$$w^l = \prod_{i=1}^M \mu_{F^i}(x_i) \quad \dots(2.18)$$



Gambar 2.13 Struktur Logika Fuzzy Takagi-Sugeno<sup>[8]</sup>

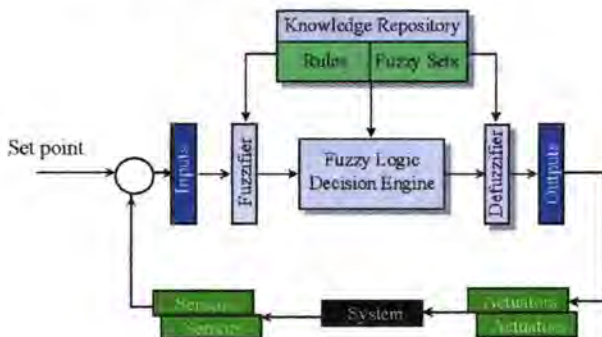
### 2.3.1 Logika fuzzy Controller (FLC)

Gambar 2.14 mengilustrasikan operasi dari logika fuzzy *control system*. Input dari sistem *fuzzy* adalah *output* dari proses, yang di masukkan kedalam sistem melalui *interface*. Sebagai contoh, dalam sebuah aplikasi control temperatur, data input dapat dimasukkan kedalam *fuzzy controller* menggunakan sebuah modul input analog. Informasi input ini kemudian diteruskan melalui proses logika fuzzy, dimana prosesor akan membandingkan input tersebut dengan sebuah rule base untuk mendapatkan sebuah output. Pemrosesan logika fuzzy termasuk eksekusi dari aturan IF...THEN yang berdasarkan dari kondisi input. Sebuah strata dari input diklasifikasikan seberapa cocokkah dengan grafik seting data tertentu. akibatnya, bila sebuah system logika fuzzy menggunakan inputan analog yang mempunyai



range hitungan dari 0 sampai 4095, grafik akan merepresentasikan setting input yang akan mencakup span dari 0 sampai 4095 hitungan<sup>[8]</sup>.

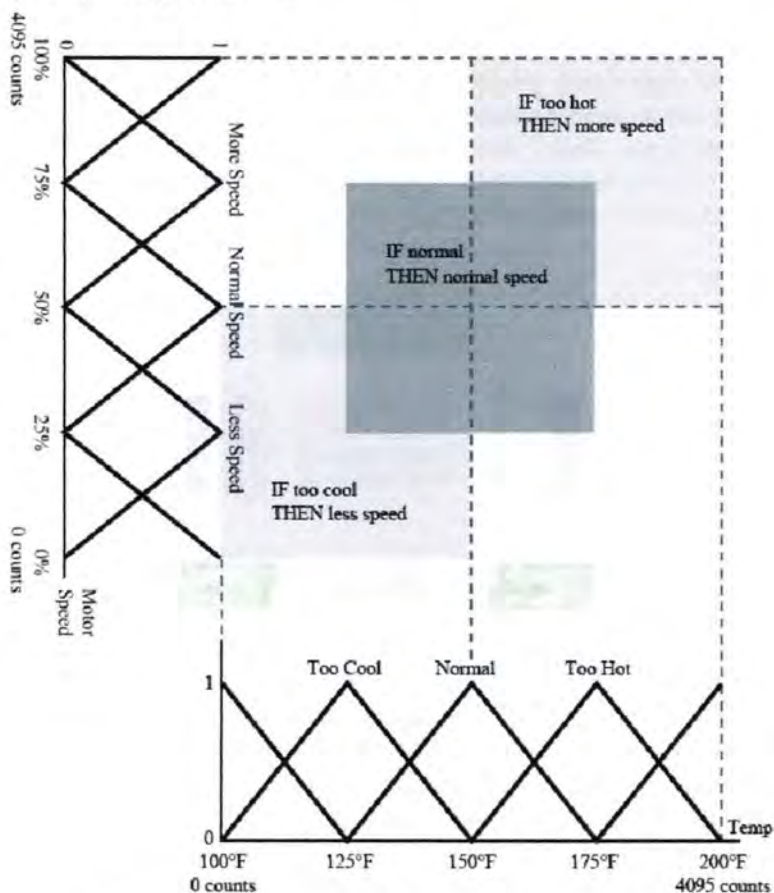
Output dari system *fuzzy controller* juga didefinisikan dengan strata, yang mana strata tersebut menentukan nilai output yang tepat untuk element kontrol. Sebagai contoh, output dari sebuah *system fuzzy* dapat mengontrol motor sebuah *air conditioner* (AC), yang berputar lebih cepat atau lebih lambat bergantung kepada strata output pada grafik output.



Gambar 2.14 Fuzzy logic Control System <sup>[20]</sup>

Gambar 2.15 mengilustrasikan sebuah grafik output logika fuzzy untuk system pendingin tersebut. Grafik tersebut memiliki strata input dan output sekaligus, yang mana *horizontal axis* adalah input (temperatur) dan *vertical axis* adalah output (kecepatan motor dari AC). Pada grafik ini, sebuah pulsa input dapat men-trigger lebih dari satu kondisi output. Contohnya, bila (IF) input temperatur adalah 137.5F, kemudian (THEN) temperatur tersebut adalah bagian dari dua setting input (50% too cool dan 50% normal). Sebagai konsekuensinya, input ini akan men-trigger dua output (input too cool akan men-trigger output less speed sedangkan input normal akan men-trigger output normal speed). Karena logika fuzzy *controller* hanya bisa mempunyai satu output, maka proses ini harus mengalami proses

selanjutnya yang disebut *defuzzification* untuk menentukan nilai output final yang *actual*<sup>[20]</sup>.



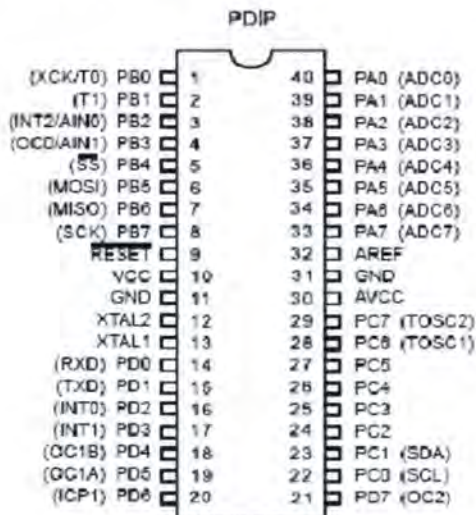
Gambar 2.15 Fuzzy logic Output Chart<sup>[20]</sup>

Seperti yang telah disebutkan pada sub Pendekatan Sistem, *Classic-Fuzzy* pada kasus ini didefinisikan sebagai sistem *fuzzy* dengan rule if-then dan membership function segitiga. Maka dalam aplikasinya, fungsi keanggotaannya hanya akan berupa

pengelompokan data secara linier. Sebagai contoh berkaitan dengan contoh diatas, maka pengelompokan misalnya suhu 100-125 F adalah *Too Cool*, suhu 126-174 F adalah Normal dan suhu 175-200 adalah *Too Hot*. Tetapi nantinya pada plant akan dilakukan pengelompokan dengan *range* yang lebih kecil, sehingga hasilnya akan lebih halus, dengan konsekuensi akan terdapat banyak statement rule IF-THEN<sup>[20]</sup>.

## 2.4 Mikrokontroler

Merupakan keluarga ATmega yang dikeluarkan oleh pabrik IC Atmel yang merupakan pabrik IC yang sangat menguasai teknologi pembuatan Flash PEROM, jadi sudah selayaknya kalau Atmel memasukkan Flash PEROM ke dalam mikrokontroler buatannya. Dimana pada mikrokontroller ini memiliki 40 kaki dengan kapasitas EEPROM sebesar 2 Kbytes dan Flash PEROM sebesar 4 Kbytes. Dimana kaki-kaki pada ATmega 8535 dapat dilihat pada gambar 2.16 dibawah ini<sup>[9]</sup>:

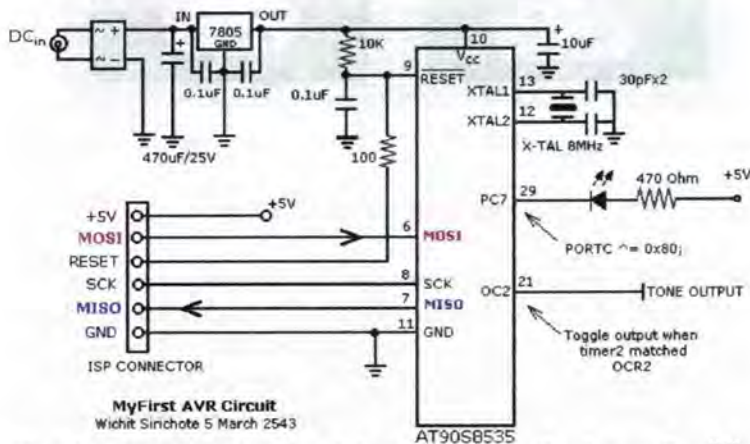


Gambar 2.16 Konfigurasi kaki-kaki ATmega 8535<sup>[9]</sup>

Tabel 2.1 Diskripsi Pin <sup>[9]</sup>

Nomor Pin	Nama Pin	Alternatif	Keterangan
11	GND		Ground
10	VCC		Power Supply
33...40	PA.0 s/d PA.7	ADC0 s/d ADC7	PA merupakan port dari ADC yang sudah terintegrasi pada IC keluaran ATmega. Pada port ini program langsung diatur pada listing yang kemudian akan langsung dikonversi pada IC.
1...8	PB.0 s/d PB.7		Port 1 berfungsi sebagai I/O biasa atau menerima low order address byte selama pada saat Flash Programming Port ini mempunyai internal pull up dan berfungsi sebagai input dengan memberikan logika 1 Sebagai output port ini dapat memberikan output sink keempat buah input TTL
21...29	PC.0 s/d PC.7	PWM	Port 2 berfungsi sebagai I/O biasa atau high order address, atau pada port ini sebagai port outputan dari PWM
14...20	PORT D		Sebagai I/O biasa Port 3 mempunyai sifat yang sama dengan Port 1 maupun Port 2. Sedangkan sebagai fungsi spesial port-port ini mempunyai keterangan sbb :
14	PD.0	RXD	Port Serial Input
15	PD.1	TXD	Port Serial Output
16	PD.2	INT0	Port Internal Interrupt 0
17	PD.3	INT1	Port Internal Interrupt 1
18	PD.4	OC1B	Port Eksternal Timer 0 Input
19	PD.5	OC1A	Port Eksternal Timer 1 Input
20	PD.6	ICP1	Eksternal Data Memory Write Strobe
9	RST		Reset akan aktif dengan memberikan input high selama 2 cycle.
13	XTAL1		Input Oscillator
12	XTAL2		Output Oscillator

Pada microcontroller ini memiliki rangkaian dasar dari minimum system yang merupakan rangkaian dasar penyusun mikrokontroller dimana pada rangkaian ini mempunyai sebuah kristal yang dipasang diantara kaki 13 dan 12 pada mikrokontroller.



Gambar 2.17 Rangkaian minimum sistem mikrokontroller [14]

## 2.5 RS (Recommended Standart)-232

Salah satu perangkat penunjang proses komunikasi data serial pada terminal komputer dengan dunia luar adalah port serial seperti yang terlihat pada gambar 2.18. Seperti halnya port paralel yang sering digunakan untuk keperluan interfacing, maka port serial juga merupakan gerbang I/O dari sebuah komputer. Hanya saja di dalam port serial akan dilakukan perubahan bentuk data paralel menjadi data serial pada jalur keluaran dan sebaliknya data serial menjadi data paralel pada jalur masukan.

Jadi pada komunikasi serial hanya ada satu bit informasi yang ditransmisikan melalui port serial pada suatu saat. Dengan demikian secara otomatis hanya ada satu kabel yang diperlukan dalam satu proses transmisi data ini. Dikatakan pula bahwa komunikasi data secara serial memiliki kesederhanaan dalam hal pengkabelannya. Hal tersebut dapat terwujud tidak lain karena



adanya port serial yang merupakan piranti dari sebuah computer yang sanggup mengubah bentuk data paralel menjadi data serial dan juga sebaliknya dari data serial menjadi data paralel<sup>[16]</sup>.



Gambar 2.18 *Serial Port*<sup>[16]</sup>

RS232 merupakan salah satu standar interface dalam proses transfer data antar komputer terutama dalam bentuk serial transfer. RS232 merupakan kependekan dari *Recommended Standard number 232*. Standar ini dibuat oleh *Elektronik Industry Association (EIA)*, untuk *interface* antara peralatan terminal data dan peralatan komunikasi data, dengan menggunakan data biner sebagai data yang ditransmisi. RS232 adalah suatu *Serial Data Interface Standard* yang dikeluarkan oleh EIA. Standarisasi meliputi konektor, fungsi serta posisi tiap per timing (pewaktu) dan level tegangan/arus. Standar ini juga berisikan karakteristik sinyal listrik, karakteristik mekanik dan cara operasional rangkaian fungsional<sup>[16]</sup>.

Tabel 2.2 Keterangan Pin RS232 <sup>[16]</sup>

Pin	Keterangan
1	<i>Carrier Detect</i> (CD) (dari DCE) Menerima sinyal datang dari modem
2	<i>Received Data</i> (RD) Data diterima dari DCE
3	<i>Transmitted Data</i> (TD) Data ditransmisikan ke DCE
4	<i>Data Terminal Ready</i> (DTR) Sinyal handshaking ditransmisikan
5	<i>Ground</i> ( <i>Common reference voltage</i> )
6	<i>Data Set Ready</i> (DSR) Sinyal handshaking datang
7	<i>Request To Send</i> (RTS) Sinyal kontrol aliran untuk DCE
8	<i>Clear To Send</i> (CTS) Sinyal kontrol aliran datang dari DCE
9	<i>Ring Indicator</i> (RI) ( <i>from DCE</i> ) Sinyal datang dari modem

## 2.6 Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya.



Gambar 2.19 Motor Servo <sup>[14]</sup>

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian control elektronik dan internal gear untuk mengendalikan pergerakan dan sudut angularnya.



Gambar 2.20 Rangkaian Motor Servo <sup>[14]</sup>

Motor servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena internal gearnya. Lebih dalam dapat digambarkan bahwa sebuah motor servo memiliki :

- 3 jalur kabel : power, ground, dan control
- Sinyal control mengendalikan posisi
- Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum.
- Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback control.

### 2.6.1 Jenis-Jenis Motor Servo :

- Motor Servo Standar  $180^\circ$   
Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai  $90^\circ$  sehingga total defleksi sudut dari kanan – tengah – kiri adalah  $180^\circ$ .
- Motor Servo Continuous  
Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu)<sup>[14]</sup>.

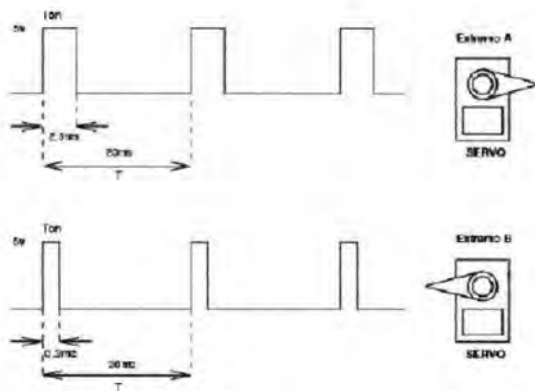
### 2.6.2 Aplikasi Motor Servo

Aplikasi motor servo hanya terbatas, biasanya digunakan pada sistem yang membutuhkan torsi dan kepresisian tinggi. Dengan kata lain sistem tersebut memiliki spesifikasi untuk beroperasi pada posisi sudut tertentu. Contoh aplikasinya misalnya :

- *Manipulators.*
- *Moving cameras.*
- *Robot arms.*

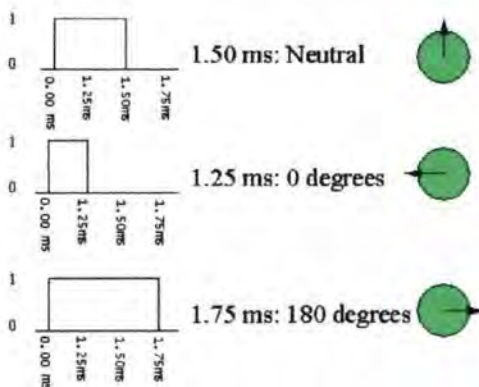
### 2.6.3 Mode Pensinyalan Motor Servo

Mode pensinyalan kontrol motor servo adalah berdasarkan sinyal step PWM. Dimana arah dan besar perubahan sudutnya bergantung pada *duty cycle* yang diberikan melalui PWM.



Gambar 2.21 Pensinyalan Motor Servo<sup>[14]</sup>

Tipikal pensinyalan motor servo standart (beroperasi pada sudut 0-180 derajat) adalah dimana bila diberikan pulsa dengan besar 1.5ms mencapai gerakan 90 derajat, bila kita memberikan data kurang dari 1.5 ms maka posisi mendekati 0 derajat dan bila kita memberikan data lebih dari 1.5 ms maka posisi mendekati 180 derajat. Pensinyalan ini dapat dilihat pada gambar 2.22 dibawah ini.



Gambar 2.22 Contoh Posisi dan Waktu Pemberian Pulsa<sup>[14]</sup>

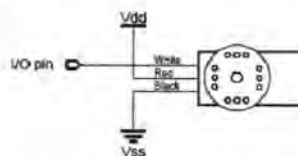


Beberapa ketentuan pemberian sinyal kontrol pada motor servo adalah sebagai berikut :

- Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50Hz.
- Dimana pada saat sinyal dengan frekuensi 50Hz tersebut dicapai pada kondisi *Ton duty cycle* 1.5ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut  $90^\circ$  / netral).
- Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle*, dan akan bertahan diposisi tersebut.
- Dan sebaliknya, jika *Ton duty cycle* dari sinyal yang diberikan lebih dari 1.5ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan diposisi tersebut<sup>[14]</sup>.

#### 2.6.4 Konfigurasi Pin pada Motor Servo

Motor servo memiliki tiga kabel (pin) sebagai inputannya. Dan secara tipikal (sudah standart) maka kabel-kabelnya memiliki susunan dan warna-warna tertentu untuk satu macam inputan. Susunan kabelnya secara berurutan adalah hitam, merah dan putih, dimana warna hitam merupakan inputan untuk ground, kabel merah merupakan inputan untuk Vcc dan kabel putih merupakan inputan untuk sinyal PWM (kontrol). Untuk lebih jelasnya konfigurasi pin (kabel) motor servo dapat dilihat pada gambar 2.23 dibawah ini.



Gambar 2.23 Konfigurasi Pin pada Motor Servo <sup>[14]</sup>

### 2.6.5 Cara Pemilihan Motor Servo

Adalah penting untuk memilih sebuah motor servo dengan spesifikasi yang sesuai dengan sistem yang akan dibangun. Diantaranya, ada tiga factor utama yang harus diperhatikan dalam pemilihan motor servo, yaitu :

1. Tentukan **Jenis** dan **Berat** beban yang akan di kontrol.
2. Tentukan kecepatan respon yang diperlukan oleh sistem.
3. Tentukan jenis dari motor servo sesuai dengan spesifikasi anda.

#### Technical Specifications

- > Power 6vdc max
- > Average Speed 60 rpm  
Note: with 5vdc and no torque
- > Weight 45.0 grams/1.59oz
- > Torque 3.10 kg-cm/17oz-in
- > Size mm (L x W x H)  
40.5x23.0x38.0
- > Size in (L x W x H)  
1.60x.79x1.50
- > Manual adjustment port



Gambar 2.24 Pemilihan Servo Berdasarkan Spesifikasinya <sup>[14]</sup>

### 2.7 Rotary Encoder

*Rotary encoder*, atau disebut juga *Shaft encoder*, merupakan perangkat elektromekanikal yang digunakan untuk mengkonversi posisi angular (sudut) dari *shaft* (lubang) atau roda ke dalam kode digital, menjadikannya semacam transduser. Perangkat ini biasanya digunakan dalam bidang robotika, perangkat masukan komputer (seperti optomekanikal *mouse* dan *trackball*), serta digunakan dalam kendali putaran radar, dll. Terdapat dua tipe utama *rotary encoder*, yaitu tipe absolut dan tipe relatif.

### 2.7.1 Rotary encoder Absolute

Tipe ini (*rotary encoder* absolut) menghasilkan kode digital yang unik/khas untuk masing-masing beda sudut poros.

#### A. Konstruksi

Plat baja dipotong dengan bentuk tertentu kemudian ditempelkan ke piringan/cakram dengan penyekat dimana terpasang kuat dengan poros (*shaft*). Saat piringan berputar, beberapa kontakny menyentuh plat baja, dan kontak yang lain tak menyentuh plat (yang berlubang). Plat baja tersebut terhubung dengan sumber arus listrik, dan masing-masing kontak terhubung ke sensor elektrik. Bentuk potongan plat baja tersebut dirancang sedemikian rupa sehingga memungkinkan masing-masing posisi poros membentuk kode biner yang unik dimana beberapa kontak terhubung ke sumber arus (*switch ON*) dan yang lain tak terhubung (*switch OFF*)<sup>[14]</sup>.



Gambar 2.25 *Rotary encoder Absolute*<sup>[14]</sup>

Kode tersebut dapat dibaca oleh peralatan kontrol seperti mikroprosesor atau mikrokontroler, untuk menerjemahkan sudut dari poros.

#### B. Standar Pengkodean Biner

Secara umum, dimana terdapat  $n$  kontak, maka jumlah posisi poros adalah  $2n$ . Misal,  $n = 3$ , maka terdapat 8 posisi poros. Di bawah ini contoh kode biner, dimana *encoder* memiliki 3 kontak :

Tabel 2.3 Kode Biner *Rotary encoder* Tiga Kontak <sup>[14]</sup>

Sector	Kontak 1	Kontak 2	Kontak 3	Sudut
1	OFF	OFF	OFF	0° - 45°
2	OFF	OFF	ON	45° - 90°
3	OFF	ON	OFF	90° - 135°
4	OFF	ON	ON	135° - 180°
5	ON	OFF	OFF	180° - 225°
6	ON	OFF	ON	225° - 270°
7	ON	ON	OFF	270° - 315°
8	ON	ON	ON	315° - 360°

Pada contoh di atas, kontak-kontak akan menghasilkan standar hitungan biner sebagai putaran *Disk* (piringan). Akan tetapi, hal ini memiliki kekurangan, yaitu bila piringan berhenti antara 2 perbatasan sektor, atau kontak-kontaknya tidak lurus secara sempurna, sehingga tidak memungkinkan untuk menerjemahkan sudut poros. Contoh, apa yang terjadi bila sudut poros berubah dari 179,9° sampai 180,1° (dari sektor 4 ke sektor 5). Sesuai dengan tabel di atas, kondisi kontak akan berubah dari *OFF-ON-ON* ke *ON-OFF-OFF*. Akan tetapi, hal ini bisa saja tidak sesuai dengan kenyataannya. Dalam prakteknya, kontak tidak pernah lurus sempurna, sehingga masing-masing akan terhubung pada momen (waktu) yang berbeda. Jika kontak 1 tersambung pertama, diikuti kontak 3 dan kemudian kontak 2, maka sekuen/urutan kode aktualnya sebagai berikut<sup>[14]</sup>:

*OFF-ON-ON* (posisi start)

*ON-ON-ON* (awal, kontak 1 *ON*)

*ON-ON-OFF* (selanjutnya, kontak 3 *OFF*)

*ON-OFF-OFF* (akhir, kontak 2 *OFF*)

Sekarang bisa diamati untuk sektor yang berkaitan dengan kode kode ini dalam tabel. Misalnya di sektor 4, 8, 7 dan 5. Sehingga, dari urutan kode yang dihasilkan, poros tampak seperti

meloncat dari sektor 4 ke sektor 8, kemudian kembali ke sektor 7, dan akhirnya kembali ke sektor 5. Kondisi ini tentu saja tidak diinginkan karena bisa merusak sistem. Sebagai contoh, jika encoder digunakan dalam *Arm Robot* (robot lengan). Kontroler akan berpikir bahwa *Arm* dalam posisi yang salah, dan mencoba untuk membetulkan kesalahan tersebut dengan memutar lengan sampai  $180^\circ$ , dan mungkin bisa menyebabkan kerusakan lengan robot<sup>[14]</sup>.

### C. Standar Pengkodean Gray

Untuk menghindari permasalahan di atas, maka digunakan *Gray Encoding*. Metode ini merupakan sistem penghitungan biner dimana dua kode yang berbatasan, dibedakan ke dalam satu posisi saja. Untuk 3 kontak pada contoh di atas, kode versi *Gray* ditunjukkan sebagai berikut:

Tabel 2.4 Kode *Gray Rotary encoder* Tiga Kontak <sup>[14]</sup>

Sektor	Kontak 1	Kontak 2	Kontak 3	Sudut
1	OFF	OFF	OFF	$0^\circ - 45^\circ$
2	OFF	OFF	ON	$45^\circ - 90^\circ$
3	OFF	ON	ON	$90^\circ - 135^\circ$
4	OFF	ON	OFF	$135^\circ - 180^\circ$
5	ON	ON	OFF	$180^\circ - 225^\circ$
6	ON	ON	ON	$225^\circ - 270^\circ$
7	ON	OFF	ON	$270^\circ - 315^\circ$
8	ON	OFF	OFF	$315^\circ - 360^\circ$

Dalam contoh ini, perubahan dari sektor 4 ke sektor 5, dan juga seluruh perubahan/transisi, hanya satu kontak yang mengubah kondisinya dari *ON* ke *OFF* atau sebaliknya. Hal ini berarti bahwa sekuen dari kode yang salah seperti yang ditunjukkan dalam tabel sebelumnya, tidak dapat terjadi.



### 2.7.2 Relative/Incremental Rotary Encoder

*Rotary encoder* Relatif (sering disebut juga *Incremental Encoder*) digunakan ketika metode pengkodean absolut tidak bisa digunakan (disebabkan ukuran dan bentuk piringan/cakram). Metode ini juga menggunakan piringan yang dipasang pada poros, tetapi ukuran piringan/cakram lebih kecil, dengan jumlah garis radial yang banyak, seperti jeruji roda. Sebuah saklar optik, seperti photodiode, menghasilkan pulsa listrik. Kemudian rangkaian kontrol elektronika menghitung pulsa untuk menerjemahkan sudut putar dari poros.



Gambar 2.26 *Rotary encoder* Relative <sup>[14]</sup>

Sistem ini dalam bentuk yang paling sederhana, tidak dapat mengukur sudut absolut dari poros. Sistem ini hanya dapat mengukur perubahan sudut relatif menjadi data yang berubah ubah, seperti posisi poros pada saat power di-ON-kan. Ketidakpastian posisi ini tidak menjadi masalah untuk perangkat input komputer seperti mouse dan trackball. Ketika posisi absolut harus diketahui, maka bisa ditambahkan sensor kedua untuk mendeteksi ketika poros melewati posisi nol (*zero*). Permasalahan kedua yang bisa muncul di sistem ini adalah tidak dapat menentukan arah putaran poros. Untuk mengatasinya, dengan menambahkan sensor optik menjadi 2 buah sensor yang dipasang pada sudut yang berbeda. Arah putaran dapat diketahui dari hasil pembacaan dua sensor tersebut. Tipe *encoder* ini dikenal dengan *Quadrature Encoder* <sup>[14]</sup>.

**BAB III**  
**RANCANG BANGUN PROTIPE WIND**  
**TURBIN DAN *BLADE PITCH ANGLE***  
***CONTROL SYSTEM***

### **BAB III**

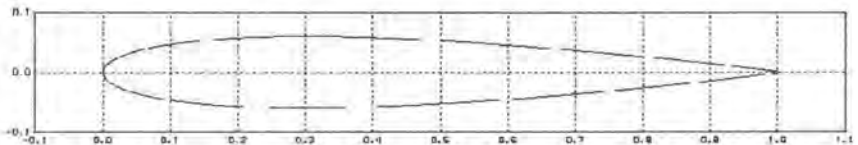
## **RANCANG BANGUN PROTOTYPE WIND TURBINE DAN BLADE PITCH ANGLE CONTROL SYSTEM**

Pada ini akan dibahas langkah-langkah perancangan dan pembangunan dari prototipe *wind turbine* dan *blade pitch angle control system*. Sistem yang akan dirancang dibagi menjadi dua bagian utama. Yaitu plant dan kontroler. Dan juga dilengkapi dengan sub sistem pendukung berupa sensor dan monitoring.

### **3.1 Prototipe Wind Turbine**

Prototipe *wind turbine* yang berkedudukan sebagai plant pada sistem yang akan dibangun, digambar menggunakan *software* autoCad. Adapun bagian terpenting dari prototipe ini sendiri adalah blade-nya, karena blade sangat menentukan putaran dari *wind turbine*.

Dalam kaitannya dengan hal ini, maka dipakailah blade dengan kontur *air foil* menyerupai standart NACA. Pada kasus ini dipakai NACA 0012 yang seperti telah teruji dalam Tugas Akhir saudara Bangun Endro bertema Pengujian Rotor Blades Horizontal Axis Wind Turbine dengan NACA 0012, 4412, 4415 untuk Kecepatan 2-5 m/s<sup>[5]</sup>. Sebagai pertimbangan lain dari pemakaian NACA 0012 ini adalah bentuknya yang simetris, sehingga mudah untuk dibuat. Selain itu juga akan lebih presisi dalam pembuatan miniaturnya. Pada rancangan blade ini tidak terdapat gradasi dimensi pada garis horizontalnya (garis panjangnya) hal ini dilakukan untuk memudahkan pembuatan blade. Tetapi realisasi dari blade ini sendiri nantinya tentu tidak sempurna seperti standar NACA 0012 dalam segi ketebalan, kehalusan, berat dan bahannya. Meskipun demikian hal ini tidak menjadi masalah, karena seluruh bagian plant dan kontroler akan diwakili oleh system fuzzy yang akan dibangun. Jadi yang terpenting adalah data input dan outputnya. Dari data tersebut korelasinya akan diwakili oleh system fuzzy. Inilah yang membedakan kontroler cerdas dari kontroler konvensional.



Gambar 3.1 Rancangan *Air Foil* Naca 0012 (skala 1:10 cm) <sup>[21]</sup>

Dengan skala yang telah dibuat, maka didapatkanlah bahwa panjang air foil adalah 10 cm dan tebalnya 1.2 cm. sedangkan panjang dari blade adalah 50 cm. Rancangan blade ini kemudian diserahkan kepada Lab Non Metal di Politeknik Perkapalan Negeri Surabaya (PPNS) untuk dibuat dengan bahan dasar fiber.

Langkah selanjutnya adalah merancang penopang *blade* dan mekanisme mekanik (*gear box*) untuk mengontrol sudut *blade*. Perancangan *gear box* dilakukan menggunakan perhitungan dari spesifikasi motor servo. Bila digunakan motor servo standart, maka akan mempunyai *range duty cycle* dari 8-40. Data 8 mewakili sudut  $0^\circ$  dan 40 mewakili  $180^\circ$ . Tetapi pada kenyataannya (aplikasi servo) *range* data adalah 8-34. Data 8 mewakili sudut  $0^\circ$  dan data 34 mewakili  $180^\circ$ . Sudut operasi dari servo standart sendiri adalah  $180^\circ$ . Jadi resolusi dari servo adalah:

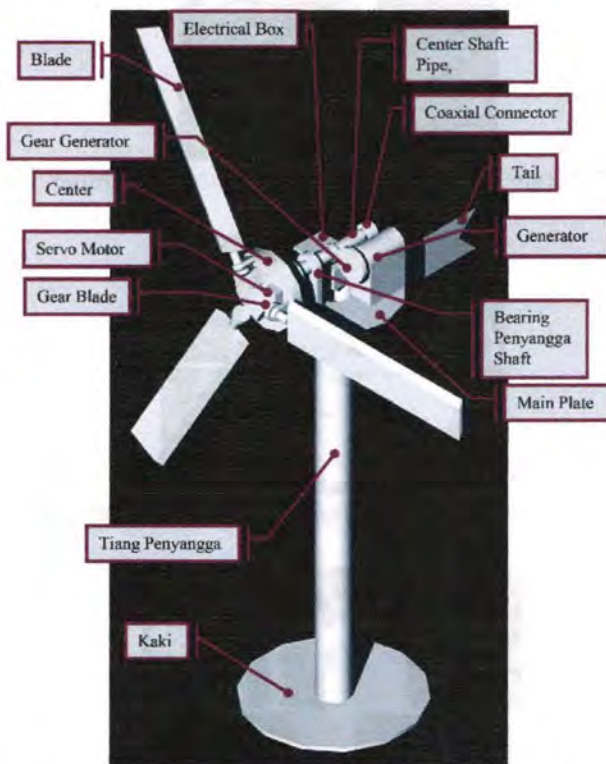
$$\begin{aligned} \text{resolusi servo} &= \frac{\text{span sudut servo}}{\text{span data servo}} \\ &= \frac{(180 - 0) (\text{deg})}{(34 - 8) (\text{dat})} = 6.92 \sim 7 \text{ deg/dat} \end{aligned}$$

Dengan kata lain untuk satu data, menimbulkan perubahan sebesar  $7^\circ$ . Untuk memperkecil resolusi tersebut, maka dibuatlah sebuah *gear box*. Setelah dilakukan survei lapangan dan dilakukan perancangan dari sistem penggerak dan pengubah sudut *blade* didapatkan bahwa gear yang ada dipasaran mempunyai perbandingan jumlah gerigi 20 (pada *shaft* motor) dan 25 (pada



*shaft blade*). Jadi perubahan sudut yang akan dicapai adalah  $5.6^{\circ}$  untuk tiap 1 data.

Adapun desain dari penopang disesuaikan dengan keperluan sistem mekanisme pengontrol sudut ini. Dirancang dengan bahan Teflon (nylon) berdiameter 18 cm dengan pertimbangan menyediakan tempat untuk 3 motor servo yang ukurannya adalah  $4 \times 2 \times 3.8$  cm. Gambar rancangan model wind turbine dalam autoCAD beserta spesifikasinya dapat dilihat dalam gambar dan tabel dibawah ini.

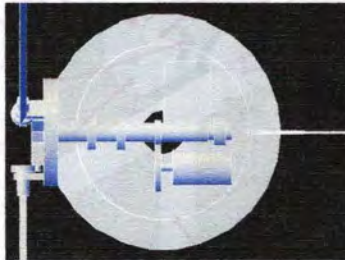


Gambar 3.2 Rancangan Prototipe *Wind Turbine* Tampak Prespektif

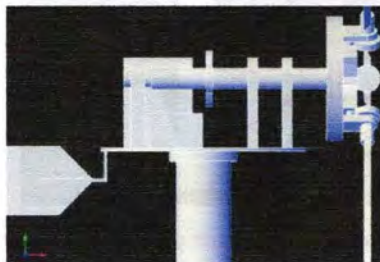




Gambar 3.2 Rancangan Prototipe *Wind Turbine* Tampak Depan




Gambar 3.3 Rancangan Prototipe *Wind Turbine* Tampak Atas



Gambar 3.3 Rancangan Prototipe *Wind Turbine* Tampak Samping

Tabel 3.1 Spesifikasi Rancangan Prototipe *Wind Turbine*

SPESIFIKASI	
<b>Blade</b>	
Panjang = 50 cm	
Lebar = 10 cm	
Shaft Blade = stainless steel, $\varnothing$ 1 cm, l = 10 cm	
Gear Blade = $\varnothing$ 2 cm, l = 1,5 cm	
Bahan = nylon PE 1000	
<b>Servo</b>	
Merk : GWServo S03	
Dimensi : p = 4.05 cm, l = 2 cm, h = 3.8 cm	
Tegangan = 6 VDC (max)	
Kecepatan Putar = 60 rpm	
Berat = 45 gram	
Torsi = 3,4 kg-cm	
Gear Servo = $\varnothing$ 1cm, l = 1,5cm	
	
<b>Center Plate</b>	
$\varnothing$ 18 cm, t = 1 cm	
Bahan = teflon	
<b>Rotor Shaft</b>	
Pipa Stainless Steel, $\varnothing$ 1,5 cm, t = 0.2 cm, p = 30 cm	
<b>Main Plate</b>	
$\varnothing$ 40 cm, t = 0.3 cm	
Bahan = plat besi	
<b>Tiang Penyangga</b>	
Pipa besi, $\varnothing$ 8 cm, t = 0.15 cm	
<b>Kaki</b>	
$\varnothing$ 50 cm, t = 0.1 cm	
Bahan = plat Besi	
<b>Rotational Connector</b>	
2 buah Slip Carbon Brusher	
Merk : Bosch	

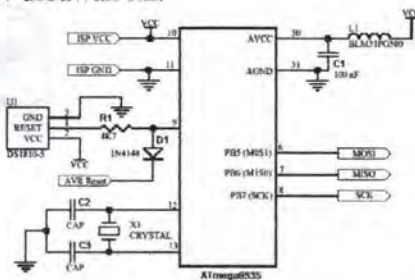
Desain dan spesifikasi ini dibangun di Lab Non Metal Politeknik Perkapalan Negeri Surabaya dan di bengkel las dan bubut milik Bapak Sugeng di Manukan Krajan, Surabaya.

### 3.2 Blade Pitch Angle Control System

Sub sistem ini terdiri dari beberapa komponen utama, yaitu, mikrokontroler AT-Mega8535, *rotary encoder* dan motor servo.

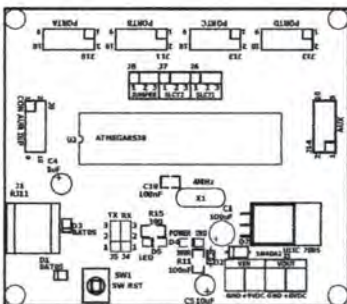
#### 3.2.1 Mikrokontroler AVR AT-Mega8535

Pada perancangan ini digunakan paket mikrokontroler DT-AVR produksi Inovative Electronics (IE). DT-AVR ini menggunakan At-Mega8535 dan sekaligus termasuk didalamnya minimum sistemnya. Rangkaian minimum sistem dapat dilihat pada gambar 3.4 dibawah ini.



Gambar 3.4 Minimum Sistem pada DT-AVR <sup>[17]</sup>

Sedangkan konfigurasi dan tata letak dari DT-AVR dapat dilihat pada gambar 3.5 dibawah ini.



(a)



(b)

Gambar 3.5 Konfigurasi Pin DT-AVR (a) Denah <sup>[17]</sup> (b) Real

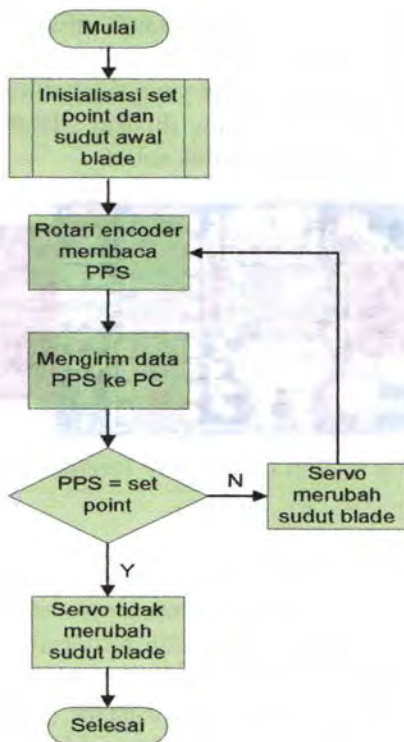
Untuk men-*download* (memprogram) mikrokontroler AT-Mega8535 ini dibutuhkan USB *downloader*, karena akses *download*-nya menggunakan laptop yang tidak dilengkapi dengan *parallel port*. USB *downloader* yang dipakai adalah K-125i USB to ISP Programmer produk dari klinik robot.



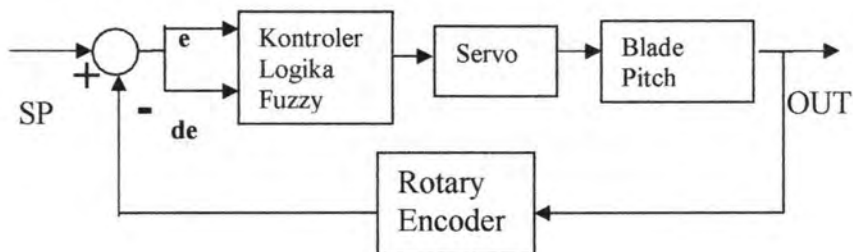
Gambar 3.6 K-125i USB to ISP Programmer <sup>[17]</sup>

### 3.2.2 Kontroler Fuzzy

Kontroler Fuzzy yang dibuat dalam perancangan ini dititikberatkan untuk mengontrol kecepatan sudut dari *shaft wind turbine*. Metode fuzzy yang digunakan termasuk dalam metode Mamdani yaitu dengan defuzzifikasi *The Midle of Maximum method* (MOM) dengan sedikit penyesuaian (penyederhanaan). Pada sistem kontrol fuzzy yang akan dibangun digunakan persamaan garis, bukan area (luasan) seperti yang ada pada *software* Matlab atau yang lain. Hal ini dilakukan untuk menyederhanakan dan mempermudah penulisannya dalam bahasa pemrograman. Jadi nilai keanggotaannya dikelompokkan dalam persamaan garis. Inputan dalam sistem ini adalah kecepatan sudut dan outputnya adalah data yang mewakili sinyal PWM untuk merubah sudut dari *blade*. Dengan berubahnya sudut serang dari angin yang ditangkap, maka kecepatan sudut juga akan berubah. Flowchart dan diagram blok sistem dapat dilihat pada gambar 3.7 dan 3.8.



Gambar 3.7 Flow Chart Sistem Kontrol



Gambar 3.8 Diagram Blok Sistem



Adapun *range* dari kecepatan sudut akan didapatkan dari pengambilan data awal (tanpa kontroler) akan dilakukan menggunakan blower dengan kecepatan angin 0-5 m/s. Dalam pengambilan data ini pula sudut dari blade diubah-ubah dari  $0^\circ$  –  $90^\circ$ . Kemudian ditentukan median (nilai tengah) dari kecepatan sudutnya dan diasumsikan sebagai *range* kerja optimal generator (*set point*).

### 3.2.3 Pengambilan Data Awal Wind Turbine dan Penentuan Fungsi Keanggotaan

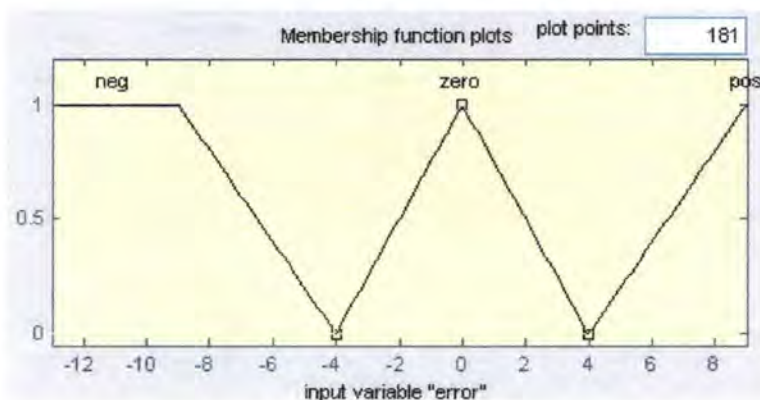
Untuk mengetahui *range* kecepatan sudut yang dihasilkan oleh *wind turbine* yang telah didesain, maka dilakukan pengambilan data awal. Pengambilan data dilakukan 5 kali dengan kecepatan angin dan sudut yang naik dan 5 kali dengan kecepatan angin dan sudut turun. Masing-masing dilakukan dengan alat ukur *Tachometer* dan juga hasil pembacaan *rotary encoder*. Keseluruhan data ada pada lampiran A, sedangkan rata-ratanya dapat dilihat pada tabel 3.2.

Tabel 3.2 Data PPS (Pulsa Per Secon) *Rotary Encoder*

PPS Rotary Encoder (pulsa/detik)			Rerata Keseluruhan					
No.	Data Servo (PWM)	Sudut (derajat)	Kecepatan Angin (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10,3	16
3	10	11,2	0	0	0	0	14,2	20,3
4	11	16,8	0	0	0	2,4	15,3	22,4
5	12	22,4	0	0	0	0	12,6	18
6	13	28	0	0	0	0	11,6	16,3

7	14	33,6	0	0	0	0	10,5	13,1
8	15	39,2	0	0	0	0	8,9	11
9	16	44,8	0	0	0	0	5,2	8,7
10	17	50,4	0	0	0	0	3,9	7
11	18	56	0	0	0	0	0,4	5
12	19	61,6	0	0	0	0	0	2,8
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

Dari hasil pengambilan data awal tersebut, kemudian dirancang sistem kontroler fuzzy. Perancangan ini didasarkan pada hasil pembacaan *rotary encoder*, seperti yang terlihat pada tabel 3.2 dengan diadakan pembulatan menjadi integer terdekat terlebih dahulu. Dari tabel tersebut dapat dilihat, bahwa *range* data input adalah 0-22 pps. Maka dapat ditentukan *set point* adalah 13 pps, dari sini bisa didapatkan bahwa *range* dari fungsi keanggotaan input eror adalah  $0-13 = -13$  PPS pada minimalnya dan  $22-13 = 9$  PPS pada maksimalnya. Eror nol dijadikan titik tengahnya. Bentuk dari fungsi keanggotaannya adalah sebagai berikut :

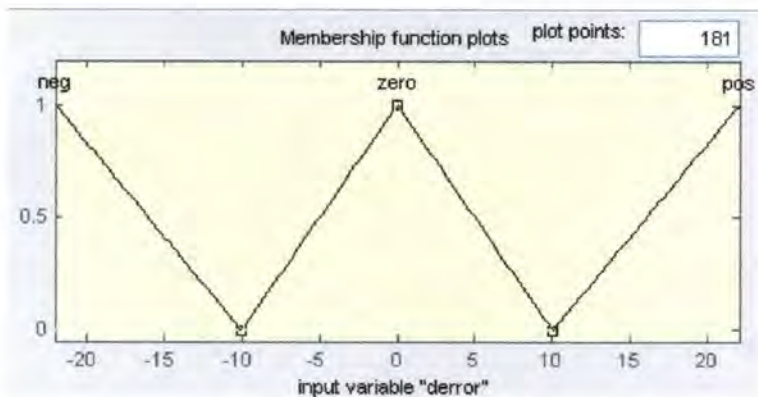


Gambar 3.9 Fungsi Keanggotaan Error

Dan persamaan garis fungsi keanggotaan error adalah sebagai berikut :

Jika	$-13 \leq x < -9$	maka	$y_1 = 1$
Jika	$-9 \leq x < -4$	maka	$y_1 = -0.2x - 0.8$
Jika	$-4 \leq x < 0$	maka	$y_1 = 0.25x + 1$
Jika	$0 \leq x < 4$	maka	$y_1 = -0.25x + 1$
Jika	$4 \leq x < 9$	maka	$y_1 = 0.2x - 0.8$

Pada fungsi keanggotaan input delta error didesain mengikuti error. *Range* dari delta error menurut error yang terjadi adalah error maksimal dikurangi dengan error minimal, yaitu  $-13 - 9 = -22$  pada minimalnya dan  $9 - (-13) = 22$  pada maksimalnya. Sehingga bentuk fungsi keanggotaannya adalah sebagai berikut :



Gambar 3.10 Fungsi Keanggotaan Delta Eror

Dan persamaan garis fungsi keanggotaannya adalah sebagai berikut :

Jika  $-22 \leq x_2 < -10$  maka  $y_2 = -0.083x_2 - 0.83$

Jika  $-10 \leq x_2 < 0$  maka  $y_2 = 0.1x_2 + 1$

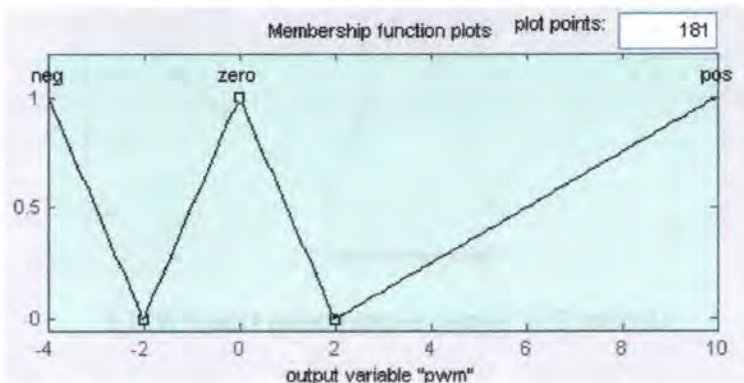
Jika  $0 \leq x_2 < 10$  maka  $y_2 = -0.1x_2 + 1$

Jika  $10 \leq x_2 < 22$  maka  $y_2 = 0.083x_2 - 0.83$

Adapun untuk fungsi keanggotaan output (data PWM) didapatkan dengan cara mengurangi data PWM hingga sudut yang menghasilkan putaran minimal dan menambah PWM hingga sudut maksimal yang menghasilkan putaran. Dalam hal ini jika dilihat dari tabel 3.2 maka didapatkan data PWM yang memungkinkan untuk menghasilkan kecepatan sudut sesuai *set point* 13 PPS adalah data PWM 12 untuk kecepatan 4 m/s dan data PWM 14 untuk kecepatan 5 m/s. Dapat pula diasumsikan data PWM berfluktuasi diantara 12-14 untuk kecepatan angin diantara 4-5 m/s. Kemudian dapat ditentukan *range* fungsi keanggotaannya adalah  $8-14 = -6$  PPS untuk minimalnya dan  $24-14 = 10$  PPS untuk maksimalnya. Data PWM 8-24 adalah data



yang mewakili sudut pengujian. Setelah itu dirancanglah bentuk dari fungsi keanggotaannya sebagai berikut :



Gambar 3.11 Fungsi Keanggotaan Data PWM

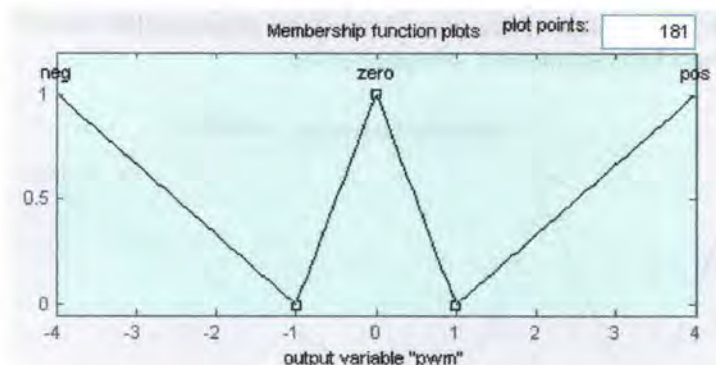
Yang kemudian didapatkan persamaan garisnya adalah sebagai berikut :

- $x_3 = -2y_3 - 2$  bila hasil rule adalah negative
- $x_3 = 2y_3 - 2$  bila hasil rule adalah zero
- $x_3 = -2y_3 + 2$  bila hasil rule adalah zero
- $x_3 = 8y_3 + 2$  bila hasil rule adalah positif

Rancangan fuzzy ini disebut KLF 1.

Dan sebagai penyempurnaan, dibuat fungsi keanggotaan yang berbeda untuk outputnya dengan menghilangkan data yang menghasilkan  $PPS = 0$  (menyeleksi data output). Hal ini dilakukan dengan pertimbangan akan banyak respon yang dibawah *set point*. Jadi *range* data minimal menjadi  $12-4 = 8$  dan *range* maksimal  $14+4 = 18$  sehingga menjadi :





Gambar 3.12 Fungsi Keanggotaan Data PWM 2

Dan persamaan garisnya adalah :

$x_3 = -3y_3 - 1$  bila hasil rule adalah negative

$x_3 = y_3 - 1$  bila hasil rule adalah zero

$x_3 = -y_3 + 1$  bila hasil rule adalah zero

$x_3 = 3y_3 + 1$  bila hasil rule adalah positif

Rancangan fuzzy yang kedua ini dinamakan sebagai KLF2.

Dari fungsi keanggotaan input dan output yang telah dirancang tersebut, kemudian disusun tabel basis aturan secara sederhana yang menggambarkan hubungan antara fungsi keanggotaan input eror dan input delta eror dengan fungsi keanggotaan output sebagai berikut :

Tabel 3.3 Rule Base Fuzzy

sudut		Error		
		neg	zero	Pos
dError	neg	neg	neg	Zero
	zero	neg	zero	Pos
	pos	zero	pos	Pos

Dari table tersebut dapat dituliskan sebagai berikut :

**If** is error neg and derror is neg **then** sudut is neg  
**If** is error neg and derror is zero **then** sudut is neg  
**If** is error neg and derror is pos **then** sudut is zero  
**If** is error zero and derror is neg **then** sudut is neg  
**If** is error zero and derror is zero **then** sudut is zero  
**If** is error zero and derror is pos **then** sudut is pos  
**If** is error pos and derror is neg **then** sudut is zero  
**If** is error pos and derror is zero **then** sudut is pos  
**If** is error pos and derror is pos **then** sudut is pos

Defuzzifikasi yang dipakai adalah *Midle of Maximum method* (MOM) dimana pada metode ini, hasil keputusannya adalah nilai tengah dari input terbesar (maksimal). Misalnya eror bernilai 0 dimana 0 memiliki derajat keanggotaan 1 terhadap zero dan derror bernilai -10 yang memiliki derajat keanggotaan 0 terhadap neg, karena fungsi keanggotaanya berupa persamaan garis maka nilai tengahnya tentulah berupa titik yang paling maksimum dalam garis tersebut, sehingga nilai outputnya adalah 1. Menurut rule base, bila error zero dan derror neg, maka keluarannya adalah neg. Dalam hal ini nilai 1 diplot kedalam fungsi keanggotaan sudut, sehingga mendapatkan nilai sebenarnya.

Pada aplikasinya pada mikrokontroler (bahasa pemrograman C), fungsi keanggotaan tersebut ditulis dalam bentuk persamaan garis yang untuk lebih jelasnya dapat dilihat pada Lampiran D. Dari sinilah kemudian kontroler dapat merespon inputan secara tepat.

### 3.3 Sensor dan Aktuator

Sensor dalam hal ini adalah sensor kecepatan putar (RPM) yang dirancang menggunakan rotary encoder. Sedangkan aktuatornya adalah berupa aktuator posisi sudut yang dirancang menggunakan motor servo.

### 3.3.1 Rotary Encoder

Sebagai sensor kecepatan sudut pada sistem ini digunakan *rotary encoder* tipe *relative* yang paling sederhana buatan sendiri. Adapun komponennya terdiri dari sebuah piringan dengan 20 lubang pada tepinya dan sebuah rangkaian *optocoupler* tipe U. *Optocoupler* diletakkan sedemikian hingga dapat mengapit piringan. Dengan posisi seperti ini, maka cahaya infra merah yang dipancarkan ke *photo transistor* akan terhalang oleh piringan ketika terkena titik yang tidak berlubang dan akan tersampaikan ketika terkena titik yang berlubang.



Gambar 3.13 Rotary Encoder <sup>[14]</sup>

Dari sini akan terbentuk pulsa *high* ketika terkena lubang dan akan terbentuk pulsa *low* ketika tidak terkena lubang. Adapun perhitungan kecepatannya bisa dilakukan dengan menghitung pulsa *high* yang terjadi. Jika terhitung 20 kali *high*, berarti telah terjadi satu putaran penuh. Waktu untuk pencapaian 20 kali pulsa *high* ini secara berkesinambungan dihitung oleh *timer* mikrokontroler, sehingga dapat diketahui berapa lama waktu yang dibutuhkan untuk menempuh satu putaran. Dari sinilah kemudian didapatkan RPS (*Rotation Per Second*) dan kemudian dikonversi menjadi RPM (*Rotation Per Minute*).

### 3.3.2 Motor Servo

Sebagai aktuator digunakan motor servo tipe standar merek GWServo seri S03 dengan spesifikasi seperti terlihat pada tabel 3.1. Untuk mengontrol sudutnya maka diperlukan sinyal PWM yang dapat dibangkitkan melalui mikrokontroler. Dimana *range*

kerja servo adalah dengan *duty cycle* 40 – 200 ms. Titik netral (sudut 90 derajat) didapatkan pada *duty cycle* 150 ms. Bila *duty cycle* kurang dari 150 ms, maka akan mendekati sudut nol derajat dan bila *duty cycle* lebih dari 150 ms, maka akan mendekati sudut 180 derajat linier relatif terhadap penambahan *duty cycle*<sup>[13]</sup>. Dalam hal ini perubahan tiap satu data pada mikro akan menghasilkan perubahan sudut 7° pada servo atau 5.6° pada blade.

### 3.4 Rotational Connector

Demi mendukung sampainya daya listrik kepada sistem pengubah sudut yang terdapat pada penopang blade yang dalam hal ini dalam keadaan berputar, maka dibutuhkan sebuah konektor yang bisa menyalurkan dengan keadaan berputar. Untuk itu dirancanglah sistem *rotational connector*. Setelah mengecek pada komponen yang ada dipasaran, didapatkan *rotational connector* merek Mercotac dengan tingkat efisiensi yang sangat tinggi. Konektor ini tidak dijual dipasaran Indonesia secara luas. Produk ini hanya bisa diakses melalui Pasar Glodok Jakarta dengan cara inden dan dipesankan dari Singapura denan harga yang tinggi, mencapai 2.5 juta Rupiah.



Gambar 3.14 Mercotac Rotational Connector<sup>[19]</sup>

Hal ini sangat menyulitkan secara biaya, maka dari itu dicoba untuk mendesain *rotational connector* sendiri. Pada kasus ini dilakukan berbagai survei dan penelitian mini. Desain awal yang didapatkan adalah memakai bearing seperti terlihat pada gambar 3.15. Konektor ini tidak bekerja dengan baik karena pada



desainnya kurang presisi titik tengahnya (center) dan juga memiliki gangguan yang besar karena permukaan sentuhnya kecil dan adanya oli (*stampad*) sehingga penyaluran arus tidak maksimum.



Gambar 3.15 *Rotational Connector Menggunakan Bearing*

Konnector hasil rancangan kedua adalah dengan menggunakan *Slip Carbon Brush* yang biasanya digunakan dalam bor listrik untuk menyalurkan arus pada lilitan motornya. Desain kedua ini dirasa yang paling memungkinkan dan paling baik penyaluran arusnya. Cara kerjanya adalah dengan menggunakan karbon pada ujung diamnya dan menggunakan lempengan tembaga pada ujung berputarnya. Jadi dengan adanya sentuhan (gesekan) antara karbon dan tembaga inilah yang memungkinkan tersalurkannya arus listrik.



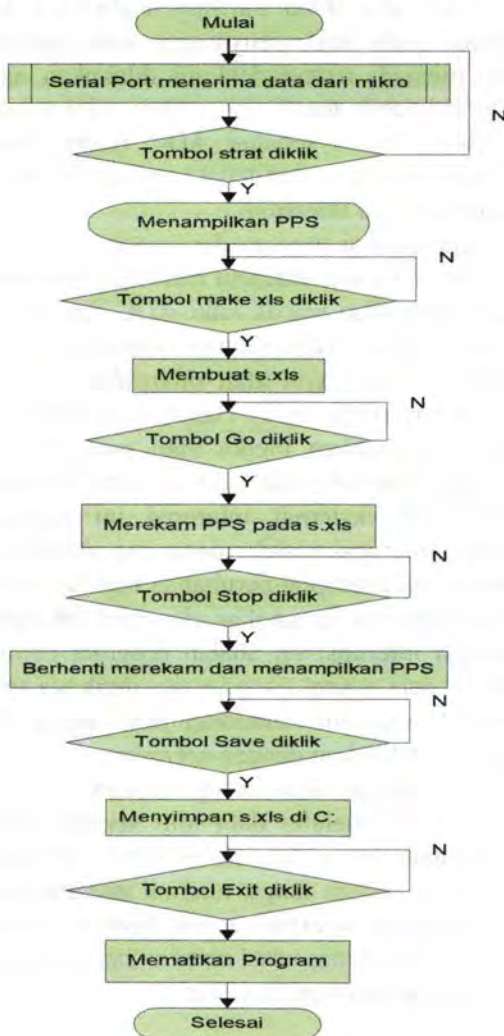
Gambar 3.16 *Rotational Connector Menggunakan Slip Carbon Brush*

### 3.5 Perancangan *Software Monitoring Sederhana*

Untuk memonitoring proses, maka disusunlah sebuah *software monitoring* sederhana menggunakan *software VB.6* dengan *interface serial port*. *Software* ini nantinya hanya akan menampilkan PPS secara *online*, *timing* dan indikator *set point*.



Kelengkapan lainnya adalah perekam data menggunakan Excel. Jadi melalui VB akan membuat sebuah *file* Excel yang akan merekam seluruh data kecepatan sudut yang terjadi pada plant.



Gambar 3.17 Flow Chart Software Monitoring

### 3.6 Hasil Perancangan

Dari perancangan yang telah dilakukan, kemudian direalisasikan dan dibangun menjadi sebuah model *Blade Pitch Angle Control System* dan *Wind Turbine*. Pada hasil akhirnya, terdapat perbedaan pada kaki penyangga wind turbine, dari bentuk lingkaran menjadi *tripod*. Hal ini dilakukan agar *wind turbine* dapat berdiri lebih tegak dan stabil. Alokasi pin pada mikrokontroler yaitu Pin A sebagai ADC (input dari *rotary encoder*), Pin C sebagai output dari hasil pembacaan ADC, Pin D diaktifkan sebagai eksternal *interrupt*.

Pada Pin A yang dipakai inputan dari *rotary encoder* adalah Pin A.0. *Rotary encoder* yang didesain memiliki outputan antara 0-5 volt. Dari sini kemudian masuk kaki ADC dan dibaca. Dari pembacaan ini ADC difungsikan sebagai semacam komparator, yaitu bila membaca nilai 4 volt atau dibawahnya, maka akan mengeluarkan nilai low pada Pin C. Dan bila membaca diatas 4 volt maka akan mengeluarkan logika high pada Pin C. Hasil keluaran ini kemudian masuk pada kaki eksternal interupsi pada Pin D.2. Fungsi dari eksternal interupsi ini adalah untuk mengaktifkan fungsi interupsi apabila menerima perubahan logika low ke high (*rising edge*) ataupun perubahan high ke low (*falling edge*). Pada perancangan ini digunakan eksternal interupsi *rising edge*. Isi dari fungsi interupsi ini adalah program *counter* data perubahan *high-low* dari *rotary encoder* per detiknya (pulsa per detik). Dari hasil *counter* ini kemudian akan masuk kedalam kontrol fuzzy dan dibandingkan dengan *set point*.

Kontrol fuzzy kemudian akan mengeluarkan output berupa perubahan sinyal PWM (melalui data mili second) yang pada akhirnya akan merubah sudut dari motor servo. Adapun sinyal PWM ini dibangkitkan dengan fungsi internal interupsi agar tidak mengganggu fungsi utama (kontroler logika fuzzy). Gambar hasil *software monitoring* sederhana, *wind turbine* dan listing program fuzzy dapat dilihat pada lampiran B, C dan D.

## BAB IV PENGUJIAN DAN ANALISA DATA

## BAB IV PENGUJIAN DAN ANALISA DATA

Setelah dilakukannya perancangan dan pembuatan alat, maka pada bab ini akan dibahas mengenai pengujian terhadap *hardware* yaitu pengujian terhadap *Rotating Connector*, sensor *Rotary Encoder* dan respon dari *Blade Pitch Angle Control System*.

### 4.1 Pengujian *Rotating Connector*

Pengujian terhadap *Rotating Connector* dilakukan dengan mengukur tegangan keluaran *power supply* (baterei) dan membandingkannya dengan setelah melewati *Rotating Connector*. Dalam pengujian ini digunakan sebuah baterai di ujung yang berputar dan diukur menggunakan multimeter dan diamati pula pada *osilloscope* dengan skala 10 Volt/div pada ujung yang diam.

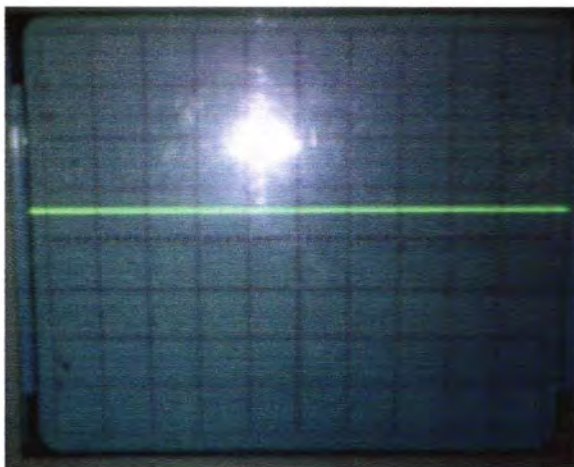
Tabel 4.1 Pengukuran Baterai

No	Tegangan Baterai (V)
1	7.85
2	7.86
3	7.85
4	7.85
5	7.85
$\Sigma = 39.26$	
$\bar{x} = 7.852$	





Gambar 4.1 Pengamatan Tegangan Baterei Secara Langsung



Gambar 4.2 Pengamatan Tegangan Baterei Setelah Melalui  
*Rotating Connector*



Tabel 4.2 Pengujian Rotating Connector

No	Kec. Sudut (rpm)	Tegangan Baterei (V)	Tegangan <i>Rotating Connector</i> (V)					
			1	2	3	4	5	$\bar{x}$
1	1,20	7.852	7.62	7,63	7,6	7,62	7.62	7.62
2	15,60	7.852	7.63	7.64	7.64	7.63	7.62	7.632
3	37,80	7.852	7,63	7.64	7.64	7.64	7.64	7.638
4	45,87	7.852	7.63	7.64	7.65	7.64	7.64	7.64
5	61,64	7.852	7.63	7.64	7.66	7.64	7.65	7.644
6	67,26	7.852	7.66	7.65	7.65	7.65	7.65	7.652

Dari data tersebut kemudian dihitung eror rata-rata yang mewakili *Loss* Tegangan, Ralat mutlak ( $\Delta$ ), Ralat nisbi (*I*) dan keseksamaan pengukuran (*K*)<sup>[11]</sup>.

$$\text{Loss Tegangan} = \frac{\Sigma(\bar{x} - X)}{n} \times 100\% \quad \dots(4.1)$$

$$\text{Ralat mutlak } (\Delta) = \frac{\Sigma(\bar{x} - X)^2}{n(n-1)} \quad \dots(4.2)$$

$$\text{Ralat Nisbi } (I) = \frac{\Delta}{\sqrt{n}} \times 100\% \quad \dots(4.3)$$

$$\text{Keseksamaan} = 100\% - I \quad \dots(4.4)$$

Dimana :

$\bar{x}$  = Data rata-rata (V)

X = Data yang sebenarnya (V)

n = Banyaknya data

Hasil perhitungan *Loss Tegangan*, Ralat mutlak ( $\Delta$ ), Ralat nisbi (I) dan keseksamaan pengukuran (K) seperti dibawah ini.

Loss Tegangan = 2.675 %

Ralat mutlak ( $\Delta$ ) = 0.009208

Ralat Nisbi (I) = 0.375899 %

Keseksamaan = 99.6421 %

#### **4.2 Pengujian Sensor *Rotary Encoder***

Pengujian *rotary encoder* dilakukan dengan membandingkan pembacaan *rotary encoder* dengan pembacaan Tacho Meter. Adapun hasil pembacaan *rotary encoder* adalah berupa Pulsa Per Second artinya adalah, dalam satu detik ada berapa pulsa (lubang) yang terbaca. Jadi kemudian karena lubangnya ada 20 buah. Maka pembacaan yang ada dibagi 20 untuk menemukan jumlah Rotation Per Second (RPS). Setelah itu untuk mendapatkan Rotation Per Menit (RPM) maka hasil RPS dikalikan 60. Hasilnya dapat dilihat pada tabel berikut ini.

Tabel 4.3a Pengujian *Rotary Encoder*

Pembacaan Standart Tachometer	Pembacaan Alat Rotary Encoder (rpm)										Rata-Rata (rpm)		Rata-Rata Pembacaan (rpm)	Error (rpm)	$(v_i - E)^2$
	Naik					Turun					Naik	Turun			
(rpm)	1	2	3	4	5	1	2	3	4	5					
1,20	0	3	0	0	3	3	0	0	0	3	1,2	1,2	1,2	0,00	0,07
7,60	6	6	6	9	9	3	6	9	9	9	7,2	7,2	7,2	0,40	0,02
8,08	9	12	9	6	9	6	6	9	9	9	9	7,8	8,4	0,32	0,00
11,73	12	12	12	12	12	9	12	12	12	12	12	11,4	11,7	0,03	0,06
15,05	15	15	15	15	15	15	15	15	12	15	15	14,4	14,7	0,35	0,01
15,60	15	18	15	15	18	15	15	18	18	15	16,2	16,2	16,2	0,60	0,11
21,06	21	21	21	21	21	21	21	21	21	21	21	21	21	0,06	0,04
26,24	27	24	24	24	27	27	27	27	24	27	25,2	26,4	25,8	0,44	0,03
26,71	27	27	27	27	30	27	27	27	24	27	27,6	26,4	27	0,29	0,00
31,00	30	30	30	30	33	30	30	30	30	30	30,6	30	30,3	0,70	0,19
31,67	33	30	33	30	33	30	33	30	33	33	31,8	31,8	31,8	0,13	0,02
33,12	33	33	33	33	33	33	33	33	33	33	33	33	33	0,12	0,02
34,93	36	36	36	33	36	33	33	33	36	36	35,4	34,2	34,8	0,13	0,02



Setelah diperoleh data pengujian, kemudian dihitung besarnya eror pengukuran. Eror atau kesalahan adalah penyimpangan nilai dari suatu pengukuran terhadap harga sebenarnya. Rata-rata eror yang didapatkan pada pengujian ini adalah 0.27.

Kemudian dihitung pula persentase nilai presisi dan akurasi. Presisi atau ketelitian adalah keterdekatan hasil pengukuran yang dilakukan berulang-ulang terhadap rata-rata pengukuran. Sedangkan akurasi didefinisikan sebagai keterdekatan hasil pengukuran suatu alat ukur terhadap suatu nilai standart yang disepakati, atau terhadap suatu nilai yang benar (*true value*). Persamaan nilai presisi dan akurasi <sup>[11]</sup>:

$$\text{Persen presisi \%} = 100\% - \left| \frac{X_n - \bar{X}_n}{\bar{X}_n} \right| \times 100\% \quad \dots(4.5)$$

$$\text{Akurasi (A)} = 1 - \left| \frac{Y_n - X_n}{Y_n} \right| \quad \dots(4.6)$$

Dimana :

$\bar{x}_n$  = Data rata-rata pembacaan berulang (rpm)

$X_n$  = Data pembacaan berulang (rpm)

$Y_n$  = Data Pembacaan Tacho Meter (rpm)

Dari table 4.4 diketahui bahwa tingkat akurasi rata-rata dari sensor adalah = 0.9876 dan tingkat presisinya = 98.753%.



Tabel 4.4 Nilai Akurasi Pengujian Alat

Pembacaan Standart Tachometer (rpm)	Akurasi	Presisi (%)
1,20	0,997	99,750
7,60	0,947	94,444
8,08	0,961	96,230
11,73	0,998	99,772
15,05	0,977	97,596
15,60	0,962	96,317
21,06	0,997	99,698
26,24	0,983	98,282
26,71	0,989	98,914
31,00	0,977	97,679
31,67	0,996	99,581
33,12	0,996	99,646
34,93	0,996	99,636
37,80	0,992	99,191
39,38	0,987	98,697
42,66	0,999	99,851
45,87	0,994	99,408
47,99	1,000	99,972
48,90	0,994	99,397
54,10	0,998	99,809
61,64	0,988	98,790
67,26	0,999	99,906
	$\mu = 0,9876$	$\mu = 98,753$

Untuk mengetahui baik tidaknya pembacaan *rotary encoder* maka perlu dilakukan kalibrasi dengan mencari *Uncertainly type A* ( $U_A$ ). Analisa  $U_A$  biasanya ditandai dengan adanya data pengukuran, misalnya  $n$  kali pengukuran, maka selanjutnya dari data tersebut akan ditemukan rata-ratanya, standar deviasi atau repetabilitynya<sup>[11]</sup>.

$$\text{Uncertainly type } U_A = \frac{SD}{\sqrt{n}} \quad \dots (4.7)$$

Dimana :

$n$  = banyaknya pengukuran

SD = standart deviasi

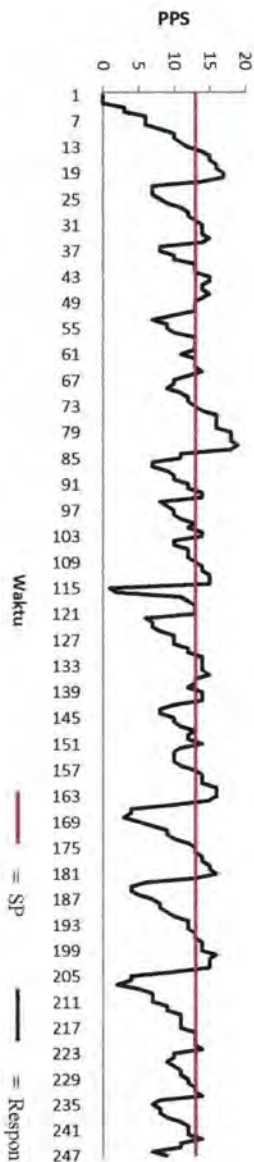
Standar deviasi adalah pengembangan dari suatu pengukuran, dimana dengan memberikan nilai  $n$  yang menunjukkan banyaknya data. Deviasi ini merupakan perbedaan antara masing-masing data<sup>[11]</sup>.

$$\text{Standard Deviasi (SD)} = \sqrt{\frac{\sum (X_n - \bar{X})^2}{n - 1}} \quad \dots (4.8)$$

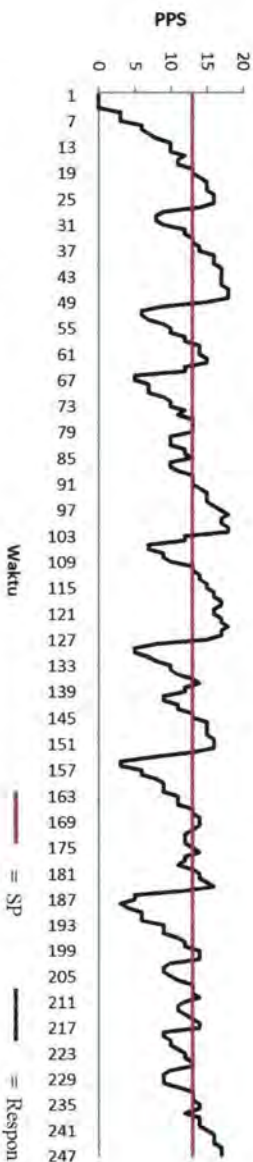
Dari tabel 4.3a dan 4.3b maka didapatkan nilai standart deviasi adalah 0.224 dan dari sini didapatkan *Uncertainty type A* = 0.0478.

#### 4.3 Pengujian Respon *Blade Pitch Angle Control System*

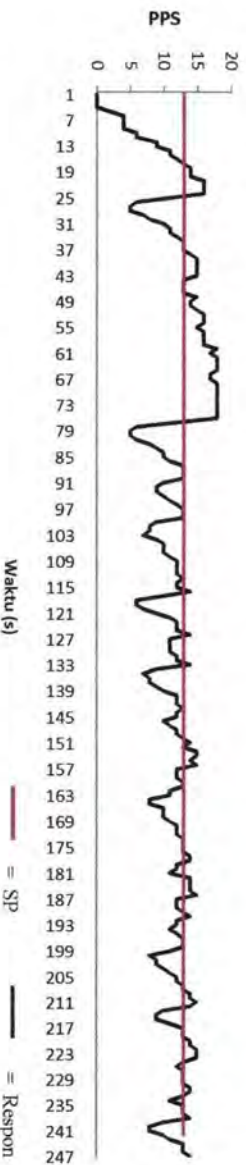
Pengujian respon *Blade Pitch Angle Control System* dilakukan dengan meniupkan angin melalui blower milik Laboratorium Mekanika Fluida Jurusan Teknik Mesin FTI-ITS. Pada pengujian, wind turbine dikenai angin dengan kecepatan antara 4-5 m/s. Kemudian diamati respon rpm terhadap waktunya untuk mencapai set point selama 4 menit. Pengujian pertama menggunakan KLF1 dan kemudian diganti dengan KLF2.



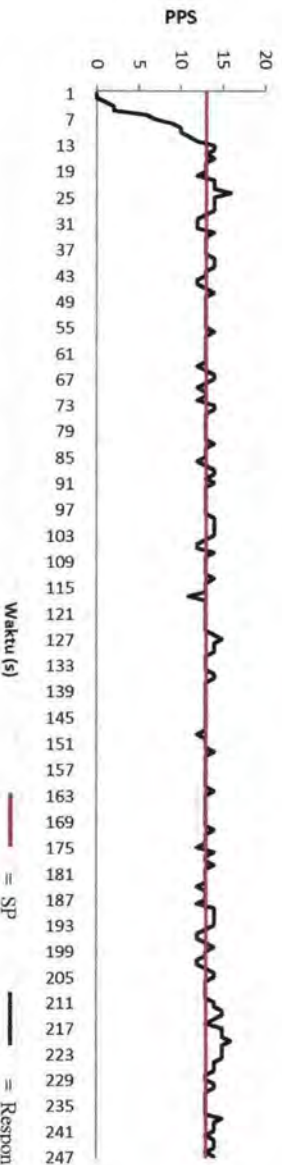
Grafik 4.1 Respon PPS Terhadap Waktu dengan KL/F1 dan Penurunan Vangin 5 ke 4 m/s



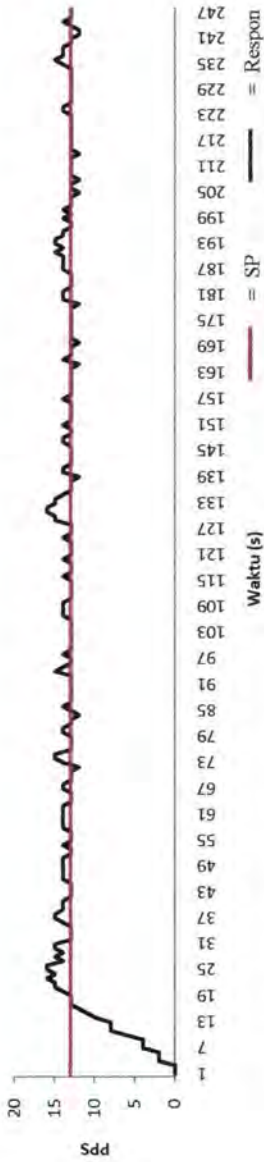
Grafik 4.2 Respon PPS Terhadap Waktu dengan KL/F1 dan Kenaikan Vangin 4 ke 5 m/s



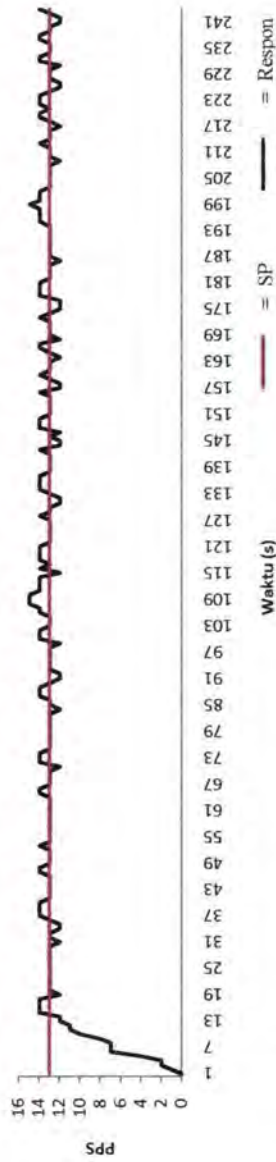
Grafik 4.3 Respon PPS Terhadap Waktu dengan KLF1 dan Vangin Berfluktuatif 3.8- 5.2 m/s



Grafik 4.4 Respon PPS Terhadap Waktu dengan KLF2 dan Penurunan Vangin 5 ke 4 m/s



Grafik 4.5 Respon PPS Terhadap Waktu dengan KLF2 dan Kenaikan Vangin 4 ke 5 m/s



Grafik 4.6 Respon PPS Terhadap Waktu dengan KLF2 dan Vangin Berfluktuatif 3.8- 5.2 m/s



Pada pengujian sistem menggunakan KLF1 dilakukan pengujian dengan tiga keadaan yang berbeda. Keadaan pertama adalah pengujian dengan kecepatan angin 5 m/s yang kemudian ditengah-tengah pengujian (detik ke-115) kecepatan anginnya dirubah menjadi 4 m/s untuk mengetahui responnya. Ketika kecepatan angin diturunkan, maka PPS langsung turun drastis hingga mencapai 1 PPS. Pada posisi ini kemudian sistem melakukan aksi kontrol sehingga PPS kembali ke posisi pengendalian. Grafik 4.2 menunjukkan pengujian dengan perlakuan kecepatan angin awal 4 m/s menjadi 5 m/s pada detik ke-124. Hasilnya adalah, PPS naik hingga mencapai 18 PPS. Pengujian ketiga adalah dengan memberikan kecepatan angin secara acak berkisar pada 3.8-5.2 m/s. Hasilnya sangat berfluktuasi (osilasi).

Sedangkan pada pengujian sistem menggunakan KLF2 juga dilakukan dengan tiga keadaan yang sama seperti pada pengujian sistem dengan KLF1. Pada grafik 4.4 dilakukan perubahan kecepatan dari 5 m/s menjadi 4 m/s pada detik ke-117 dan akhirnya terjadi penurunan PPS hingga mencapai angka 11 PPS, sebelum kemudian dapat melakukan aksi kontrol kembali. Demikian pula pada saat diuji terhadap kenaikan kecepatan angin dari 4 ke 5 m/s pada detik ke-132 seperti terlihat pada grafik 4.5 terjadi kenaikan PPS hingga 16 PPS sebelum kemudian dapat melakukan aksi kontrol kembali. Sedangkan pada pengujian dengan kecepatan angin yang berfluktuatif antara 3.8 - 5.2 m/s, responnya tergolong cukup baik. Karakteristik yang dapat dihitung meliputi maksimum *overshoot*, *rise time*, *settling time*, *error steady state* dan ITAE (*Integral Time Absolute Error*). Dan karakteristik yang dihitung adalah dari data pengujian KLF1 dan KLF2 dengan kecepatan angin yang berfluktuatif antara 3.8-5.2 m/s.

$$ITAE = \int_0^{\infty} t|e(t)| dt \quad \dots(4.9)$$

Tabel 4.5 Karakteristik Respon

Kontroler Logika Fuzzy	Rise Time (sec)	Max Over shoot (PPS)	Settling Time (sec)	ESS (%)	ITAE
KLF1	18	5	87	6.61	25015.545
KLF2	14	1	20	3.64	328

Dari tabel dapat dilihat bahwa sistem yang menggunakan KLF2 lebih baik dalam semua kriteria karakteristik daripada KLF1. KLF2 memiliki rise time 4 detik lebih cepat, maksimum overshoot lebih kecil 4 PPS. Perbedaan *settling time* bahkan mencapai 67 detik dan selisih *error steady state* mencapai 2.97%. Perbandingan secara performansi yang paling terlihat adalah ITAE, terlihat bahwa ITAE dari sistem dengan KLF1 jauh lebih tinggi, mencapai 25015.545. Sebuah nilai yang besar bila dibandingkan dengan nilai ITAE sistem dengan KLF2 yang hanya mencapai 328. Respon dari sistem yang menggunakan KLF1 belum bisa dikatakan steady, karena terus berfluktuatif (osilasi). Ini berarti aksi kontrolnya dapat dikatakan gagal ( $ESS > 5\%$ ). Kegagalan aksi kontrol ini disebabkan adanya sudut yang tidak menghasilkan PPS ikut dalam perhitungan data. Hal ini dibuktikan dengan banyaknya nilai kontrol yang berada dibawah setpoint. Sedangkan respon sistem dengan KLF2 dapat dikatakan aksi kontrolnya berhasil karena ESS dibawah 5%.

#### 4.4 Koefisien Power

Untuk menghitung koefisien power, maka diambil data kecepatan angin pada asumsi jarak 2 meter sebelum blade dan 2 meter setelah blade, kecepatan angin 5 m/s dan sudut blade  $33.6^\circ$  dimana PPS yang dihasilkan adalah sama dengan setpoint (13

PPS). Hasil pengambilan data yang telah dilakukan adalah sebagai berikut :

Tabel 4.6 Kecepatan Angin Sebelum dan Sesudah Blade

V angin Sebelum (m/s)	V angin Sesudah (m/s)
5.4	3.5
5.4	3.4
5.3	3.4
5.4	3.4
5.3	3.4
$\mu = 5.36$	$\mu = 3.42$

Koefisien power ( $C_p$ ) dihitung menurut persamaan 2.3 dengan merubah  $A1$  dan  $A4$  kedalam bentuk  $A2$ , sehingga persamaannya menjadi :

$$P = P1 - P4 = \frac{1}{2} \rho_{atr} \left( \frac{2}{3} A2 v_1^3 - 2 A2 v_4^3 \right) = \rho_{atr} A2 \left( \frac{1}{3} v_1^3 - v_4^3 \right) \quad \dots (4.10)$$

Maka dengan memasukkan nilai rata-rata kecepatan angin sebelum ( $v1$ ) dan kecepatan angin sesudah ( $v4$ ), diperoleh:

$$P1 = \frac{1}{2} \rho_{atr} \left( \frac{2}{3} A2 v_1^3 \right) = \frac{1}{3} \rho_{atr} \pi R^2 (v_1^3) \quad \dots (4.11)$$

$$= \frac{1}{3} \rho_{air} \pi R^2 (5.36)^3 = 51.330 \rho_{air} \pi R^2$$

$$P4 = \frac{1}{2} \rho_{atr} (2 A2 v_4^3) = \rho_{atr} \pi R^2 (v_4^3) \quad \dots (4.12)$$

$$= \rho_{air} \pi R^2 (3.42)^3 = 40.002 \rho_{air} \pi R^2$$

Jadi bisa dihitung  $C_p$  sebagai berikut :

$$C_p = \frac{P_1 - P_4}{P_1} = \frac{(51.330 - 40.002)(\rho_{air} \pi R^2)}{51.330(\rho_{air} \pi R^2)}$$

$$= 0.22$$

$C_p$  yang diperoleh termasuk buruk (kecil) karena  $C_p$  ideal adalah berkisar antara 0.35 – 0.4. Hal ini terjadi karena desain dari blade yang kurang presisi dan kurang proporsional bila dibandingkan dengan beban dari hub, blade, servo dan rangkaiannya. Dengan memakai  $C_p$  yang telah dihitung tersebut, kemudian dihitung power angin yang dapat diekstrak oleh wind turbin.

$$P = \frac{1}{2} \rho_{air} C_p A_2 v_2^3 \quad \dots (4.13)$$

$$= \frac{1}{2} \times 1.225 \left( \frac{kg}{m^3} \right) \times 0.22 \times 3.14 (0.5^2) m^2 \times 5^3 \left( \frac{m}{s} \right)^3$$

$$= 13.22 \text{ Watt}$$

## BAB V KESIMPULAN DAN SARAN



## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan eksperimen yang telah dilakukan, maka didapatkan beberapa kesimpulan sebagai berikut :

- Telah berhasil dirancang dan dibangun sebuah *Blade Pitch Angle Control System* pada sebuah model *wind turbine* dengan  $C_p$  0.22 pada nilai set poin 13 PPS, sudut  $33.6^\circ$  dan kecepatan angin 5 m/s.
- Perkiraan daya yang dapat dihasilkan *model wind turbine* adalah 13.22 watt menurut  $C_p$  yang pada poin kesimpulan diatas.
- Hasil perancangan *rotating connector* dapat mengalirkan daya listrik dengan *loss* daya berkisar antara 2.675 %.
- Hasil perancangan sensor RPM menggunakan rotary encoder dapat bekerja dengan tingkat akurasi 0.9876 dan tingkat presisi 98.753%.
- KLF1 memiliki rise time 18 detik, maksimum overshoot 5 PPS, settling time 87 detik, error steady state 6.61% dan ITAE bernilai 25015.545 dinilai gagal ( $ESS > 5\%$ ) dikarenakan banyaknya data sudut yang tidak menghasilkan putaran sudut.
- KLF2 memiliki rise time 14 detik, maksimum overshoot 1 PPS, settling time 20 detik, error steady state 3.64% dan ITAE bernilai 328 dinilai baik ( $ESS < 5\%$ ) dan dapat merespon perubahan kecepatan angin untuk mempertahankan setpoint 13 PPS jauh lebih baik daripada KLF1.

## 5.2 Saran

Beberapa hal yang perlu diperhatikan dalam kelanjutan penelitian pada Plant ini, antara lain :

- Dalam pengerjaan hendaknya blade didesain dengan baik terlebih dahulu bekerjasama dengan Bidang Minat Energi, jadi TA dikerjakan tim.
- Untuk mendesain sebuah KLF yang baik, maka pada range fungsi keanggotaannya perlu diseleksi dahulu data-datanya, misalnya pada penelitian ini data sudut *blade* yang menghasilkan kecepatan sudut nol di eliminasi.
- Agar ditambahkan sensor kecepatan angin untuk mensuplai input ketiga terhadap KLF.

## DAFTAR PUSTAKA

## DAFTAR PUSTAKA

- [1] Andrzej Piegat. 2005. *A New Definition Of The Fuzzy Set*. Faculty Of Computer Science And Information Systems Technical University Of Szczecin Ul. Żołnierska, Poland.
- [2] Chris Yonge. 2007. *Pitch Adjustment Of Windmill And Turbine Blades: A New Method*. Artikel.
- [3] Dr. Gary L. Johnson. 2001. *Wind Energy Systems*. \_\_\_\_.
- [4] Dwinell, J. H. 1949. *Principles of Aerodynamics*. New York. McGraw - Hill.
- [5] Endro, Bangun. 2008. *Pengujian Rotor Blades Horizontal Axis Wind Turbine dengan NACA 0012, 4412, 4415 untuk Kecepatan 2-5 m/s*. Surabaya. Institute Teknologi Sepuluh Nopember. Tugas Akhir.
- [6] F. Martin McNeill and Ellen Thro. 1994. *Fuzzy Logic A Practical Approach*, Academic Press Limited. London
- [7] Hari Sharma. 2007. *Effect Of Pitch Control And Power Conditioning On Power Quality Of Variable Speed Wind Turbine Generators*. Murdoch University Energy Research Institute (Mueri). Artikel.
- [8] Hartadi, Dwi. 2006. *Perancangan Sistem Pengendalian Temperatur Menggunakan Metode Fuzzy Gain Schedulling PID Controller Pada Continous Tank Reaktor (CSTR)*. Surabaya. Institute Teknologi Sepuluh Nopember. Tugas Akhir.
- [9] Hendawan Soebhakti, ST. 2007. *Basic AVR Microcontroller Tutorial*. Batam. Politeknik Batam.
- [10] Henk Polinder. 2006. *Basic Operation Principles and Electrical Conversion Systems of Wind Turbines*. Paper IEEE no. 069

- [11] Sabrennas, Aden. 2007. *Rancang Bangun Densitometer Dengan Sensor Fotodiode Berbasis Personal Computer*. Surabaya. Institute Teknologi Sepuluh Nopember. Tugas Akhir.
- [12] Steven F. Barrett and Daniel J. Pack. 2008. *Atmel AVR Microcontroller Primer: Programming and Interfacing*. Wyoming. Morgan & Claypool Publisher.
- [13] T. Olsen, E. Lang, A.C. Hansen, M.C. Cheney, G. Quandt. 2004. *Low Wind Speed Turbine Project Conceptual Design Study: Advanced Independent Pitch Control*. National Renewable Energy Laboratory. Collorado.
- [14] Tim Panitia Workshop KRI/KRCI. 2006. *Workshop KRI/KRCI 2007 (Modul)*. Surabaya. PENS-ITS.
- [15] Tony Burton, David Sharpe, Nick Jenkins, Ervin Bossanyi. 2001. *Wind Energy Handbook*. New York. JOHN WILEY & SONS, LTD.
- [16] [www.howstuffworks.com/serialport??.php](http://www.howstuffworks.com/serialport??.php)
- [17] [www.klinikrobot.com](http://www.klinikrobot.com)
- [18] [www.mathworks.com/Fuzzy Logic Toolbox 2.1](http://www.mathworks.com/Fuzzy Logic Toolbox 2.1)
- [19] [www.mercotac.com](http://www.mercotac.com)
- [20] \_\_\_\_\_. \_\_\_\_\_. *PLCs and Fuzzy Logic Industrial—Text and Video Company (Journal)*. \_\_\_\_\_. \_\_\_\_\_.
- [21] \_\_\_\_\_. \_\_\_\_\_. *Introduction to Fluid Mechanics – Su Third experiment: Lift and drag*. \_\_\_\_\_. \_\_\_\_\_.
- [22] \_\_\_\_\_. \_\_\_\_\_. *Pengenalan Ilmu Klimatologi*. \_\_\_\_\_. \_\_\_\_\_.





LAMPIRAN

## **LAMPIRAN A**

**DATA KECEPATAN SUDUT TACHOMETER DAN  
ROTARY ENCODER HASIL PENGAMBILAN DATA  
AWAL (TANPA KONTROLER) PROTOTIPE WIND  
TURBINE**

**Tabel A-1 Data RPM Tacho Meter Naik I**

RPM Tachometer (rotation/menit)				I				
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	91,1	144,2
3	10	11,2	0	0	0	0	126,4	180,2
4	11	16,8	0	0	0	17,7	135,3	198,4
5	12	22,4	0	0	0	0	116,2	163,1
6	13	28	0	0	0	0	107,8	153,4
7	14	33,6	0	0	0	0	100,1	117,1
8	15	39,2	0	0	0	0	81,2	100,1
9	16	44,8	0	0	0	0	45,5	80,9
10	17	50,4	0	0	0	0	35,9	62,6
11	18	56	0	0	0	0	0	44,2
12	19	61,6	0	0	0	0	0	27,5
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-2 Data RPM Tacho Meter Turun I**

RPM Tachometer (rotation/menit)			I					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	19,2	0	0	0	0	0
7	18	56	44,8	9,1	0	0	0	0
8	17	50,4	63,7	27,4	0	0	0	0
9	16	44,8	82,1	45,1	0	0	0	0
10	15	39,2	100,1	81,1	0	0	0	0
11	14	33,6	116,8	89,9	0	0	0	0
12	13	28	143,3	100,2	0	0	0	0
13	12	22,4	162,5	108,2	0	0	0	0
14	11	16,8	198,2	143,8	18,2	0	0	0
15	10	11,2	180,4	134,9	0	0	0	0
16	9	5,6	143,8	91,1	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-3 Data RPM Tacho Meter Naik II**

RPM Tachometer (rotation/menit)			II					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	89,9	144,2
3	10	11,2	0	0	0	0	126,1	189,1
4	11	16,8	0	0	0	17,6	135,2	206,5
5	12	22,4	0	0	0	0	116,8	162,7
6	13	28	0	0	0	0	108,2	145,8
7	14	33,6	0	0	0	0	91,1	109,2
8	15	39,2	0	0	0	0	71,4	100,1
9	16	44,8	0	0	0	0	45,3	82,2
10	17	50,4	0	0	0	0	35,8	63,7
11	18	56	0	0	0	0	8,8	54,2
12	19	61,6	0	0	0	0	0	36,3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0



**Tabel A-4 Data RPM Tacho Meter Turun II**

RPM Tachometer (rotation/menit)				II				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	17,9	0	0	0	0	0
7	18	56	45,4	0	0	0	0	0
8	17	50,4	63,6	35,4	0	0	0	0
9	16	44,8	81,2	44,6	0	0	0	0
10	15	39,2	100,3	80,9	0	0	0	0
11	14	33,6	117,7	91,4	0	0	0	0
12	13	28	151,9	108,3	0	0	0	0
13	12	22,4	161,7	99,8	0	0	0	0
14	11	16,8	198,2	144,7	27,4	0	0	0
15	10	11,2	190,1	135,5	0	0	0	0
16	9	5,6	144,6	99,1	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-5 Data RPM Tacho Meter Naik III**

RPM Tachometer (rotation/menit)				III				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	90,1	135,1
3	10	11,2	0	0	0	0	125,7	180,4
4	11	16,8	0	0	0	27,2	144,2	207,1
5	12	22,4	0	0	0	0	117,2	171,2
6	13	28	0	0	0	0	108,2	144,8
7	14	33,6	0	0	0	0	99,8	126,4
8	15	39,2	0	0	0	0	81,2	98,9
9	16	44,8	0	0	0	0	44,7	72,8
10	17	50,4	0	0	0	0	45,3	62,8
11	18	56	0	0	0	0	0	35,6
12	19	61,6	0	0	0	0	0	27,4
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-6 Data RPM Tacho Meter Turun III**

RPM Tachometer (rotation/menit)			III					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	27,2	0	0	0	0	0
7	18	56	45,2	0	0	0	0	0
8	17	50,4	63,2	36,2	0	0	0	0
9	16	44,8	81,8	54,4	0	0	0	0
10	15	39,2	98,6	81,2	0	0	0	0
11	14	33,6	117,5	90,4	0	0	0	0
12	13	28	143,9	98,7	0	0	0	0
13	12	22,4	161,8	107,9	0	0	0	0
14	11	16,8	207,3	143,2	36,4	0	0	0
15	10	11,2	189,6	126,7	0	0	0	0
16	9	5,6	153,2	90,1	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-7 Data RPM Tacho Meter Naik IV**

RPM Tachometer (rotation/menit)				IV				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	90,2	143,2
3	10	11,2	0	0	0	0	116,9	171,1
4	11	16,8	0	0	0	8,7	125,8	189,3
5	12	22,4	0	0	0	0	108,2	152,8
6	13	28	0	0	0	0	99,1	143,8
7	14	33,6	0	0	0	0	90,1	117,4
8	15	39,2	0	0	0	0	80,7	99,1
9	16	44,8	0	0	0	0	44,8	71,8
10	17	50,4	0	0	0	0	36,1	62,6
11	18	56	0	0	0	0	0	45,8
12	19	61,6	0	0	0	0	0	18,2
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-8 Data RPM Tacho Meter Turun IV**

RPM Tachometer (rotation/menit)			IV					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	27,2	0	0	0	0	0
7	18	56	36,3	0	0	0	0	0
8	17	50,4	63,6	27,2	0	0	0	0
9	16	44,8	72,1	54,2	0	0	0	0
10	15	39,2	98,9	72,1	0	0	0	0
11	14	33,6	126,1	107,7	0	0	0	0
12	13	28	143,5	100,1	0	0	0	0
13	12	22,4	162,2	116,8	0	0	0	0
14	11	16,8	197,7	134,2	35,8	0	0	0
15	10	11,2	189,1	135,2	0	0	0	0
16	9	5,6	144,2	90,2	0	0	0	0
17	8	0	0	0	0	0	0	0



**Tabel A-9 Data RPM Tacho Meter Naik V**

RPM Tachometer (rotation/menit)				V				
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	98,9	142,9
3	10	11,2	0	0	0	0	126,2	189,2
4	11	16,8	0	0	0	27,2	135	207,8
5	12	22,4	0	0	0	0	126,2	162,3
6	13	28	0	0	0	0	108,9	144,1
7	14	33,6	0	0	0	0	99,3	116,9
8	15	39,2	0	0	0	0	89,5	98,6
9	16	44,8	0	0	0	0	44,3	81,2
10	17	50,4	0	0	0	0	36,7	62,8
11	18	56	0	0	0	0	9,1	54,3
12	19	61,6	0	0	0	0	0	27,1
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-10 Data RPM Tacho Meter Turun V**

RPM Tachometer (rotation/menit)			V					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	0	0	0	0	0	0
7	18	56	45,8	8,9	0	0	0	0
8	17	50,4	63,3	35,8	0	0	0	0
9	16	44,8	81,2	45,2	0	0	0	0
10	15	39,2	98,8	81,9	0	0	0	0
11	14	33,6	116,3	90,2	0	0	0	0
12	13	28	152,6	108,3	0	0	0	0
13	12	22,4	162,8	116,8	0	0	0	0
14	11	16,8	207,4	134,7	26,3	0	0	0
15	10	11,2	189,9	126,3	0	0	0	0
16	9	5,6	144,2	99,4	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-11 Data RPM Tacho Meter Rereta Naik**

RPM Tachometer 3 Blade (rotation/menit)			Rerata Naik					
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Rerata Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	92,04	141,92
3	10	11,2	0	0	0	0	124,26	182
4	11	16,8	0	0	0	19,68	135,1	201,82
5	12	22,4	0	0	0	0	116,92	162,42
6	13	28	0	0	0	0	106,44	146,38
7	14	33,6	0	0	0	0	96,08	117,4
8	15	39,2	0	0	0	0	80,8	99,36
9	16	44,8	0	0	0	0	44,92	77,78
10	17	50,4	0	0	0	0	37,96	62,9
11	18	56	0	0	0	0	3,58	46,82
12	19	61,6	0	0	0	0	0	27,3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-12 Data RPM Tacho Meter Rereta Turun**

RPM Tachometer 3 Blade (rotation/menit)			Rerata Turun					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Rerata Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	18,3	0	0	0	0	0
7	18	56	43,5	3,6	0	0	0	0
8	17	50,4	63,48	32,4	0	0	0	0
9	16	44,8	79,68	48,7	0	0	0	0
10	15	39,2	99,34	79,44	0	0	0	0
11	14	33,6	118,88	93,92	0	0	0	0
12	13	28	147,04	103,12	0	0	0	0
13	12	22,4	162,2	109,9	0	0	0	0
14	11	16,8	201,76	140,12	28,82	0	0	0
15	10	11,2	187,82	131,72	0	0	0	0
16	9	5,6	146	93,98	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-13 Data RPM Tacho Meter Rereta Keseluruhan 3 Blade**

RPM Tachometer 3 Blade (rotation/menit)				Rerata Keseluruhan				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Rerata (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	93,01	143,96
3	10	11,2	0	0	0	0	127,99	184,91
4	11	16,8	0	0	0	24,25	137,61	201,79
5	12	22,4	0	0	0	0	113,41	162,31
6	13	28	0	0	0	0	104,78	146,71
7	14	33,6	0	0	0	0	95	118,14
8	15	39,2	0	0	0	0	80,12	99,35
9	16	44,8	0	0	0	0	46,81	78,73
10	17	50,4	0	0	0	0	35,18	63,19
11	18	56	0	0	0	0	3,59	45,16
12	19	61,6	0	0	0	0	0	22,8
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-14 Data PPS Rotary Encoder Naik I**

PPS Rotary Encoder (pulsa/detik)			I					
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10	16
3	10	11,2	0	0	0	0	14	20
4	11	16,8	0	0	0	2	15	22
5	12	22,4	0	0	0	0	13	18
6	13	28	0	0	0	0	12	17
7	14	33,6	0	0	0	0	11	13
8	15	39,2	0	0	0	0	9	11
9	16	44,8	0	0	0	0	5	9
10	17	50,4	0	0	0	0	4	7
11	18	56	0	0	0	0	0	5
12	19	61,6	0	0	0	0	0	3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0



**Tabel A-15 Data PPS Rotary Encoder Turun I**

PPS Rotary Encoder (pulsa/detik)			I					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	2	0	0	0	0	0
7	18	56	5	1	0	0	0	0
8	17	50,4	7	3	0	0	0	0
9	16	44,8	9	5	0	0	0	0
10	15	39,2	11	9	0	0	0	0
11	14	33,6	13	10	0	0	0	0
12	13	28	16	11	0	0	0	0
13	12	22,4	18	12	0	0	0	0
14	11	16,8	22	16	1	0	0	0
15	10	11,2	20	15	0	0	0	0
16	9	5,6	16	10	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-16 Data PPS Rotary Encoder Naik II**

PPS Rotary Encoder (pulsa/detik)				II				
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10	16
3	10	11,2	0	0	0	0	14	20
4	11	16,8	0	0	0	2	15	23
5	12	22,4	0	0	0	0	13	18
6	13	28	0	0	0	0	12	16
7	14	33,6	0	0	0	0	10	12
8	15	39,2	0	0	0	0	8	11
9	16	44,8	0	0	0	0	5	9
10	17	50,4	0	0	0	0	4	7
11	18	56	0	0	0	0	1	6
12	19	61,6	0	0	0	0	0	4
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-17 Data PPS Rotary Encoder Turun II**

PPS Rotary Encoder (pulsa/detik)				II				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	2	0	0	0	0	0
7	18	56	5	0	0	0	0	0
8	17	50,4	7	4	0	0	0	0
9	16	44,8	9	5	0	0	0	0
10	15	39,2	11	9	0	0	0	0
11	14	33,6	13	10	0	0	0	0
12	13	28	17	12	0	0	0	0
13	12	22,4	18	11	0	0	0	0
14	11	16,8	22	16	2	0	0	0
15	10	11,2	21	15	0	0	0	0
16	9	5,6	16	11	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-18 Data PPS Rotary Encoder Naik III**

PPS Rotary Encoder (pulsa/detik)			III					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10	15
3	10	11,2	0	0	0	0	14	20
4	11	16,8	0	0	0	3	16	23
5	12	22,4	0	0	0	0	13	19
6	13	28	0	0	0	0	12	16
7	14	33,6	0	0	0	0	11	14
8	15	39,2	0	0	0	0	9	11
9	16	44,8	0	0	0	0	5	8
10	17	50,4	0	0	0	0	5	7
11	18	56	0	0	0	0	0	4
12	19	61,6	0	0	0	0	0	3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-19 Data PPS Rotary Encoder Turun III**

PPS Rotary Encoder (pulsa/detik)				III				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	3	0	0	0	0	0
7	18	56	5	0	0	0	0	0
8	17	50,4	7	4	0	0	0	0
9	16	44,8	9	6	0	0	0	0
10	15	39,2	11	9	0	0	0	0
11	14	33,6	13	10	0	0	0	0
12	13	28	16	11	0	0	0	0
13	12	22,4	18	12	0	0	0	0
14	11	16,8	23	16	3	0	0	0
15	10	11,2	21	14	0	0	0	0
16	9	5,6	17	10	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-20 Data PPS Rotary Encoder Naik IV**

PPS Rotary Encoder (pulsa/detik)				IV				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10	16
3	10	11,2	0	0	0	0	13	19
4	11	16,8	0	0	0	1	14	21
5	12	22,4	0	0	0	0	12	17
6	13	28	0	0	0	0	11	16
7	14	33,6	0	0	0	0	10	13
8	15	39,2	0	0	0	0	9	11
9	16	44,8	0	0	0	0	5	8
10	17	50,4	0	0	0	0	4	7
11	18	56	0	0	0	0	0	5
12	19	61,6	0	0	0	0	0	2
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0



**Tabel A-21 Data PPS Rotary Encoder Turun IV**

PPS Rotary Encoder (pulsa/detik)				IV				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	3	0	0	0	0	0
7	18	56	4	0	0	0	0	0
8	17	50,4	7	3	0	0	0	0
9	16	44,8	8	6	0	0	0	0
10	15	39,2	11	8	0	0	0	0
11	14	33,6	14	12	0	0	0	0
12	13	28	16	11	0	0	0	0
13	12	22,4	18	13	0	0	0	0
14	11	16,8	22	15	4	0	0	0
15	10	11,2	21	15	0	0	0	0
16	9	5,6	16	10	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-22 Data PPS Rotary Encoder Naik V**

PPS Rotary Encoder (pulsa/detik)				V				
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	11	16
3	10	11,2	0	0	0	0	14	21
4	11	16,8	0	0	0	3	15	23
5	12	22,4	0	0	0	0	14	18
6	13	28	0	0	0	0	12	16
7	14	33,6	0	0	0	0	11	13
8	15	39,2	0	0	0	0	10	11
9	16	44,8	0	0	0	0	5	9
10	17	50,4	0	0	0	0	4	7
11	18	56	0	0	0	0	1	6
12	19	61,6	0	0	0	0	0	3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-23 Data PPS Rotary Encoder Turun V**

PPS Rotary Encoder (pulsa/detik)			V					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	3	0	0	0	0	0
7	18	56	5	1	0	0	0	0
8	17	50,4	7	4	0	0	0	0
9	16	44,8	9	5	0	0	0	0
10	15	39,2	11	9	0	0	0	0
11	14	33,6	13	10	0	0	0	0
12	13	28	17	12	0	0	0	0
13	12	22,4	18	13	0	0	0	0
14	11	16,8	23	15	3	0	0	0
15	10	11,2	20	14	0	0	0	0
16	9	5,6	16	11	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-24 Data PPS Rotary Encoder Rerata Naik**

PPS Rotary Encoder (pulsa/detik)			Rerata Naik					
No.	Data Servo (pwm)	Sudut (derajat)	Kecepatan Angin Naik (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10,2	15,8
3	10	11,2	0	0	0	0	13,8	20
4	11	16,8	0	0	0	2,2	15	22,4
5	12	22,4	0	0	0	0	13	18
6	13	28	0	0	0	0	11,8	16,2
7	14	33,6	0	0	0	0	10,6	13
8	15	39,2	0	0	0	0	9	11
9	16	44,8	0	0	0	0	5	8,6
10	17	50,4	0	0	0	0	4,2	7
11	18	56	0	0	0	0	0,4	5,2
12	19	61,6	0	0	0	0	0	3
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

**Tabel A-25 Data PPS Rotary Encoder Rerata Turun**

PPS Rotary Encoder (pulsa/detik)			Rerata Turun					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin Turun (m/s)					
			5	4	3	2	1	0
1	24	89,6	0	0	0	0	0	0
2	23	84	0	0	0	0	0	0
3	22	78,4	0	0	0	0	0	0
4	21	72,8	0	0	0	0	0	0
5	20	67,2	0	0	0	0	0	0
6	19	61,6	2,6	0	0	0	0	0
7	18	56	4,8	0,4	0	0	0	0
8	17	50,4	7	3,6	0	0	0	0
9	16	44,8	8,8	5,4	0	0	0	0
10	15	39,2	11	8,8	0	0	0	0
11	14	33,6	13,2	10,4	0	0	0	0
12	13	28	16,4	11,4	0	0	0	0
13	12	22,4	18	12,2	0	0	0	0
14	11	16,8	22,4	15,6	2,6	0	0	0
15	10	11,2	20,6	14,6	0	0	0	0
16	9	5,6	16,2	10,4	0	0	0	0
17	8	0	0	0	0	0	0	0

**Tabel A-26 Data PPS Rotary Encoder Rerata Keseluruhan**

PPS Rotary Encoder (pulsa/detik)				Rerata Keseluruhan				
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	10,3	16
3	10	11,2	0	0	0	0	14,2	20,3
4	11	16,8	0	0	0	2,4	15,3	22,4
5	12	22,4	0	0	0	0	12,6	18
6	13	28	0	0	0	0	11,6	16,3
7	14	33,6	0	0	0	0	10,5	13,1
8	15	39,2	0	0	0	0	8,9	11
9	16	44,8	0	0	0	0	5,2	8,7
10	17	50,4	0	0	0	0	3,9	7
11.	18	56	0	0	0	0	0,4	5
12	19	61,6	0	0	0	0	0	2,8
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0



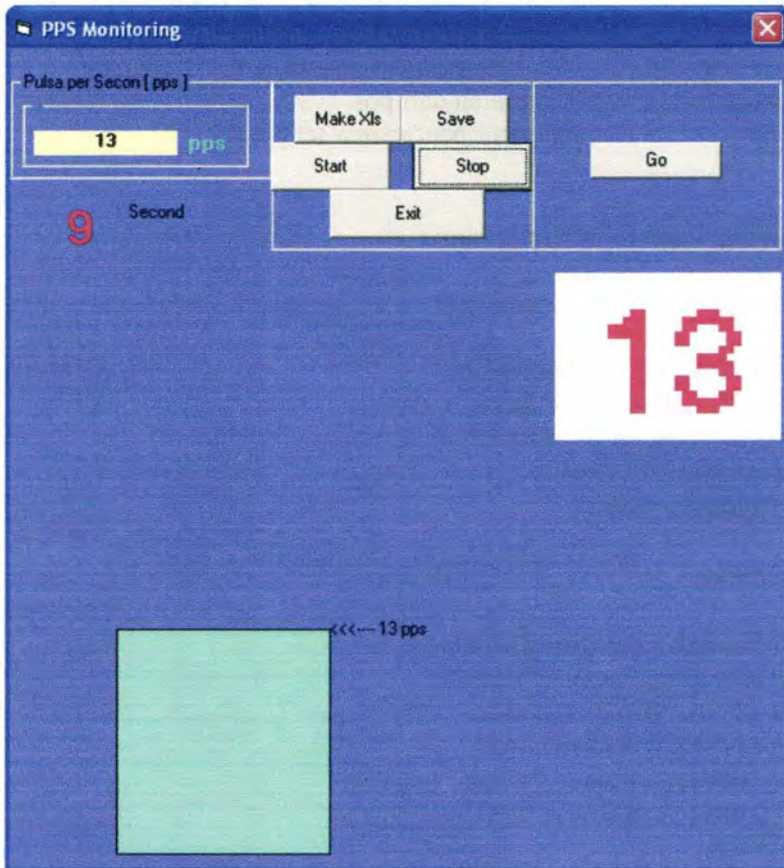
**Tabel A-27 Data Konversi PPS ke RPM**

RPM (Rotation/menit)			Rerata Keseluruhan					
No.	Data Servo (pwm)	Sudut (derajad)	Kecepatan Angin (m/s)					
			0	1	2	3	4	5
1	8	0	0	0	0	0	0	0
2	9	5,6	0	0	0	0	30,9	48
3	10	11,2	0	0	0	0	42,6	60,9
4	11	16,8	0	0	0	7,2	45,9	67,2
5	12	22,4	0	0	0	0	37,8	54
6	13	28	0	0	0	0	34,8	48,9
7	14	33,6	0	0	0	0	31,5	39,3
8	15	39,2	0	0	0	0	26,7	33
9	16	44,8	0	0	0	0	15,6	26,1
10	17	50,4	0	0	0	0	11,7	21
11	18	56	0	0	0	0	1,2	15
12	19	61,6	0	0	0	0	0	8,4
13	20	67,2	0	0	0	0	0	0
14	21	72,8	0	0	0	0	0	0
15	22	78,4	0	0	0	0	0	0
16	23	84	0	0	0	0	0	0
17	24	89,6	0	0	0	0	0	0

## **LAMPIRAN B**

### **TAMPILAN DAN LISTING PROGRAM MONITORING PPS SEDERHANA (VB.6)**

*Gambar B-1 Tampilan Software Monitoring*



Keterangan :

- Make Xls : Tombol untuk mengenerate file excel
- Start : Tombol untuk memulai monitoring
- Go : Tombol untuk memulai merekam kedalam Excel

Stop	: Tombol untuk menghentikan monitoring.
Save	: Tombol untuk menyimpan data hasil rekaman kedalam file excel yang telah degenerate.
Exit	: Tombol untuk menutup aplikasi.
Text box pps	: Untuk mengamati pps yang terjadi di plant.
Label1	: Perbesaran dari pps
Shape	: Indikator pencapaian set point.

### **Listing Code PPS Monitoring :**

```
Dim a, b, d, e, k, f, z As Integer
Dim c As String
Dim a1 As Excel.Application
Dim a2 As Excel.Workbook
Dim a3 As Excel.Worksheet
Private Sub clr_Click()
t1.Enabled = False
```

```
End Sub
```

```
Private Sub Command1_Click()
Set a1 = New Excel.Application
Set a2 = a1.Workbooks.Add
Set a3 = a2.Worksheets.Add
a3.Cells(1, 1).Value = "Time"
a3.Cells(1, 2).Value = "PPS"
End Sub
```

```
Private Sub Command2_Click()
a3.SaveAs "c:\s.xls"
a1.Quit
End Sub
```

```
Private Sub Command3_Click()
```

```
z = Text1.Text  
End Sub
```

```
Private Sub Command4_Click()  
k = 1  
End Sub
```

```
Private Sub ex_Click()  
End  
End Sub
```

```
Private Sub Form_Load()  
pret.PortOpen = True  
k = 0  
f = 1  
z = 1  
z = Text1.Text  
End Sub
```

```
Private Sub go_Click()  
pret.Output = "c"  
t1.Enabled = True
```

```
End Sub
```

```
Private Sub t1_Timer()
```

```
a = 13
```

```
If a = 0 Then  
b = 0  
End If
```

```
If a = 1 Then  
b = 1
```

End If

If  $a = 2$  Then

$b = 2$

End If

If  $a = 3$  Then

$b = 3$

End If

If  $a = 4$  Then

$b = 4$

End If

If  $a = 5$  Then

$b = 5$

End If

If  $a = 6$  Then

$b = 6$

End If

If  $a = 7$  Then

$b = 7$

End If

If  $a = 8$  Then

$b = 8$

End If

If  $a = 9$  Then

$b = 9$

End If

If  $a = 10$  Then



b = 10  
End If

If a = 11 Then  
b = 11  
End If

If a = 12 Then  
b = 12  
End If

If a = 13 Then  
b = 13  
End If

If a = 14 Then  
b = 14  
End If

If a = 15 Then  
b = 15  
End If

If a = 16 Then  
b = 16  
End If

If a = 17 Then  
b = 17  
End If

If a = 18 Then  
b = 18  
End If

If a = 19 Then  
b = 19  
End If

If a = 20 Then  
b = 20  
End If

If a = 21 Then  
b = 21  
End If

If a = 22 Then  
b = 22  
End If

If a = 23 Then  
b = 23  
End If

If a = 24 Then  
b = 24  
End If

If a = 25 Then  
b = 25  
End If

If a = 26 Then  
b = 26  
End If

If a = 27 Then  
b = 27  
End If

```
If a = 28 Then  
b = 28  
End If
```

```
If a = 29 Then  
b = 29  
End If
```

```
If a = 30 Then  
b = 30  
End If
```

```
If a > 30 Then  
b = 0  
End If
```

```
Label2.Caption = a  
Label6.Caption = a
```

```
Shape1.Height = b * 175  
Shape1.Top = 8145 - Shape1.Height
```

```
If k = 1 Then  
Label3.Caption = d  
d = d + 1  
f = f + 1  
a3.Cells(f, 1).Value = d  
a3.Cells(f, 2).Value = b  
End If  
End Sub
```

```
Private Sub Text2_Change()  
  
End Sub
```

**LAMPIRAN C**

**PROTOTIPE WIND TURBINE**

*Gambar C-1 Model Wind Turbin Tampak Depan*



*Gambar C-2 Model Wind Turbin Tampak Samping*



Spesifikasi :

Tinggi	: 1.5 meter (termasuk blade)
Diameter	: 1.18 meter (termasuk hub 0.18 meter)
Cp	: 0.22 (sudut $33.6^\circ$ dan kecepatan angin 5 m/s)
Daya	: 13.22 watt
Bahan Blade	: fiber
Bahan Hub	: Teflon
Bahan Body	: besi
Servo (3)	: Torsi 3.4 kgcm; Vcc 6volt; dimensi 4.05x2x3.8 cm



*Gambar C-3 Koneksi Baterei*



*Gambar C-4 Rangkaian Mikro dan Servo*



*Gambar C-5 Gear Box Servo dan Shaft Blade*



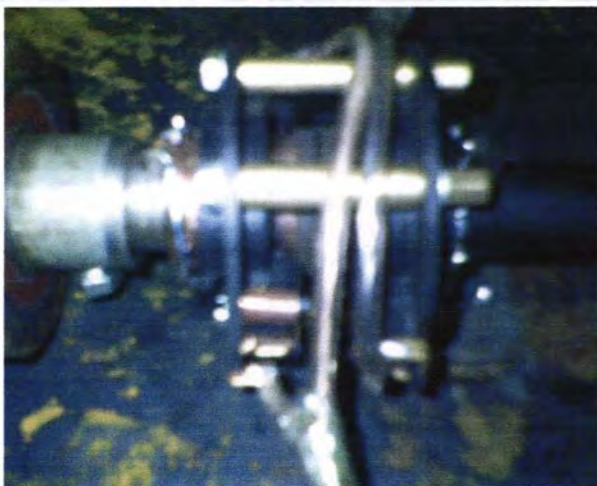
*Gambar C-6 Penyangga Shaft dan Rotating Connector*



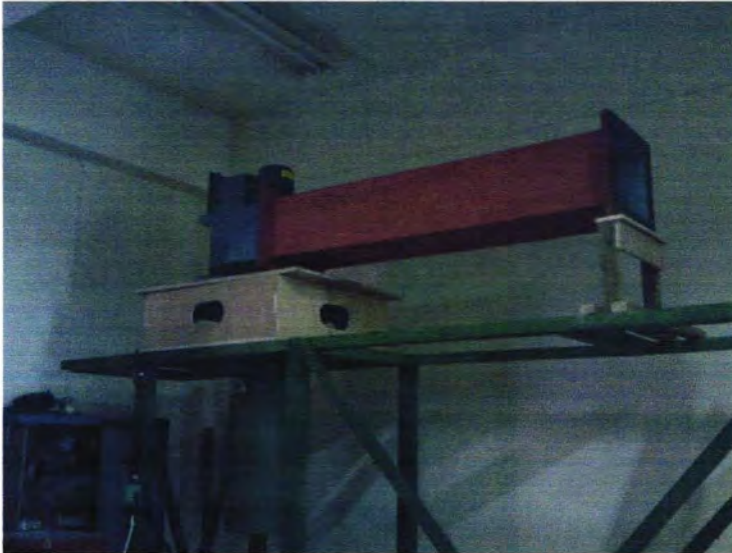
*Gambar C-7 Perubahan Desain Kaki*



*Gambar C-8 Rotating Connector*



***Gambar C-9 Blower Laboratorium Mekanika Fluida Teknik Mesin  
FTI-ITS***



Spesifikasi Blower :

Kekuatan Tiup : 0-5.4 m/s

Daya Motor : Tiga Fasa 1.5 HP

Tinggi Blower : 2.5 m (lantai-titik tengah lubang blower)

Dimensi lubang blower : 35 x 35 cm

Milik : Laboratorium Mekanika Fluida Teknik Mesin  
FTI-ITS



*Gambar C-10 Pengambilan Data1*



*Gambar C-11 Pengambilan Data 2*



*Gambar C-12 Pengambilan Data 3*





## **LAMPIRAN D**

### **LISTING PROGRAM PADA MIKROKONTROLER (AVR CODEVISION)**

**CodeVision AVR KLF1**

```

/*****

```

```

*

```

```

This program was produced by the
CodeWizardAVR V1.25.3 Professional

```

```

Automatic Program Generator

```

```

© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.

```

```

http://www.hpinfotech.com

```

```

Project :

```

```

Version :

```

```

Date   : 25/06/2009

```

```

Author : adam

```

```

Company : its

```

```

Comments:

```

```

Chip type      : ATmega8535

```

```

Program type   : Application

```

```

Clock frequency : 4,000000 MHz

```

```

Memory model    : Small

```

```

External SRAM size : 0

```

```

Data Stack size  : 128

```

```

*****

```

```

/

```

```

#include <mega8535.h>

```

```

#include <stdlib.h>

```

```

#include <delay.h>

```

```

#include <stdio.h>

```

```

#define mot_servo1 PORTB.1 //

```

```

#define mot_servo2 PORTB.3 //

```

```

#define mot_servo3 PORTB.5 //

```

```

#define ADC_VREF_TYPE 0x40

```

```

int kv,t1,a,t,ou,jpulsa,pps;
float y1,y2,y3,del,b,av;
signed int y, servo_data;
unsigned int yek;
char *out;

```

```

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
jpulsa=jpulsa+1;

}

```

```

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)!=0);
ADCSRA|=0x10;
return ADCW;
}
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
if(t==0)
{
jpulsa=jpulsa-1;
t++;}
y++;
if (read_adc(0)>150)

```

```

{
PORTC.1=1;
PORTC.0=1;
}
if(read_adc(0)<=150)
{
PORTC.1=0;
PORTC.0=0;
}

```

```

if(y==15000)
{
pps=jpulsa;
itoa(jpulsa,out);
puts(out);
jpulsa=0;
y=0;
}
}

```

```

// Timer 2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Place your code here
yek++;
if(yek==900)
{
yek=0;
}
if(yek <=servo_data)
{
mot_servo1=mot_servo2=mot_servo3=1;
}
else
{

```

```

mot_servo1=mot_servo2=mot_servo3=0;
}
}

```

// Declare your global variables here

```

void main(void)
{

```

// Declare your local variables here

```

jpulsa=0;

```

```

t=0;

```

```

a=0;

```

// Input/Output Ports initialization

// Port A initialization

```

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In

```

```

Func1=In Func0=In

```

```

// State7=T State6=T State5=T State4=T State3=T State2=T

```

```

State1=T State0=T

```

```

PORTA=0x00;

```

```

DDRA=0x00;

```

// Port B initialization

```

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In

```

```

Func1=In Func0=In

```

```

// State7=T State6=T State5=T State4=T State3=T State2=T

```

```

State1=T State0=T

```

```

PORTB=0x00;

```

```

DDRB=0xff;

```

// Port C initialization

```

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In

```

```

Func1=In Func0=In

```

```

// State7=T State6=T State5=T State4=T State3=T State2=T

```

```

State1=T State0=T

```

```

PORTC=0xff;

```



```
DDRC=0xff;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T  
State1=T State0=T
```

```
PORTD=0x04;
```

```
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 4000,000 kHz
```

```
// Mode: Normal top=FFh
```

```
// OC0 output: Disconnected
```

```
TCCR0=0x01;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 1 Stopped
```

```
// Mode: Normal top=FFFFh
```

```
// OC1A output: Discon.
```

```
// OC1B output: Discon.
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer 1 Overflow Interrupt: Off
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x00;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 4000,000 kHz
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x01;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x03;
MCUCSR=0x00;
GIFR=0x40;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x41;
```

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
```

```

// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x19;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AVCC pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x82;
SFIOR&=0xEF;

// Global enable interrupts
#asm("sei")

while (1)

//inisialisasi posisi servo awal
{
    kv++;
    if (kv==1) {
        servo_data=14;
        delay_ms(3000);
    }
}

```

```
b=pps;
};
```

```
//input fuzzy error
```

```
av=pps;
```

```
if (-13<av && av<=-9)
```

```
{
    y1=1;
};
```

```
if(-9<=av && av<-4)
```

```
{y1=-0.2*av-0.8;
};
```

```
if(-4<=av && av <=0)
```

```
{y1=0.25*av+1;
};
```

```
if(0<av && av<=4)
```

```
{y1=-0.25*av+1;
};
```

```
if(4<av && av<=9)
```

```
{y1=0.2*av-0.8;
};
```

```
//input fuzzy delta error
```

```
del=av-b;
```

```
if(-22<=del && del <-10)
```

```
{y2=-0.083*del-0.83;
};
```

```
if(-10<=del && del<=0)
```

```
{y2=0.1*del+1;
};
```

```
if(0<del && del<=10)
{y2=-0.1*del+1;
};
```

```
if(10<del && del<=22)
{y2=0.083*del-0.83;
};
```

```
if (y1>=y2) {
y3=y1;
}
else {
y3=y2;
};
```

```
if(av < -4 && del < -10)
{t1=0;
};
```

```
if(av < -4 && del >= -10 && del<= 10)
{t1=0;
};
```

```
if(av < -4 && del > 10 )
{t1=3;
};
```

```
if(av >=-4 && av<=4 && del < -10)
{t1=0;
};
```

```
if(av >=-4 && av<=0 && del >= -10 && del<= 0)
{t1=3;
```

```
};
```

```
if(av >0 && av<=4 && del >0 && del<= 10)
{t1=4;
};
```

```
if(av >=-4 && av<=0 && del >0 && del<= 10)
{t1=3;
};
```

```
if(av >0 && av<=4 && del >= -10 && del<= 0)
{t1=4;
};
```

```
if(av >=-4 && av<=4 && del > 10 )
{t1=2;
};
```

```
if(av > 4 && del <-10)
{t1=4;
};
```

```
if(av >4 && del >= -10 && del<= 10)
{t1=2;
};
```

```
if(av >4 && del > 10 )
{t1=2;
};
```

```
if(t1==0)
{ou=-2*y3-2;
};
```



```
if(t1==3)
{ou=2*y3-2;
};
```

```
if(t1==4)
{ou=-2*y3+2;
};
```

```
if(t1==2)
{ou=8*y3+32;
};
```

```
b=av;
servo_data=servo_data+ou;
```

```
};
}
```

**CodeVision AVR KLF2**

```

/*****
****

```

This program was produced by the  
 CodeWizardAVR V1.25.3 Professional  
 Automatic Program Generator  
 © Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Project :  
 Version :  
 Date : 25/06/2009  
 Author : adam  
 Company : its  
 Comments:

Chip type : ATmega8535  
 Program type : Application  
 Clock frequency : 4,000000 MHz  
 Memory model : Small  
 External SRAM size : 0  
 Data Stack size : 128

```

*****

```

```

****/

```

```

#include <mega8535.h>
#include <stdlib.h>
#include <delay.h>
#include <stdio.h>
#define mot_servo1 PORTB.1 //
#define mot_servo2 PORTB.3 //
#define mot_servo3 PORTB.5 //

```

```

#define ADC_VREF_TYPE 0x40
int kv,t1,a,t,ou,jpuls,a,pps;
float y1,y2,y3,del,b,av;
signed int y, servo_data;
unsigned int yek;
char *out;

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
jpuls=a+jpuls+1;

}

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)!=0);
ADCSRA|=0x10;
return ADCW;
}

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
if(t==0)
{
jpuls=a-jpuls-1;
t++;}
}

```

```

y++;
if (read_adc(0)>150)
{
PORTC.1=1;
PORTC.0=1;
}
if (read_adc(0)<=150)
{
PORTC.1=0;
PORTC.0=0;
}

```

```

if(y==15000)
{
pps=jpulsa;
itoa(jpulsa,out);
puts(out);
jpulsa=0;
y=0;
}
}

```

```

// Timer 2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Place your code here
yek++;
if(yek==900)
{
yek=0;
}
if(yek <=servo_data)
{
mot_servo1=mot_servo2=mot_servo3=1;

```

```

}
else
{
mot_servo1=mot_servo2=mot_servo3=0;
}
}

```

// Declare your global variables here

```

void main(void)
{
// Declare your local variables here
jpulsa=0;
t=0;
a=0;
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTB=0x00;
DDRB=0xff;

// Port C initialization

```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTC=0xff;
DDRC=0xff;
```

```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTD=0x04;
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 4000,000 kHz
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x01;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
```



```
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 4000,000 kHz
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x01;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x03;
MCUCSR=0x00;
GIFR=0x40;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x41;
```

```
// USART initialization
```

```
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: Off
```

```
// USART Transmitter: On
```

```
// USART Mode: Asynchronous
```

```
// USART Baud rate: 9600
```

```
UCSRA=0x00;
```

```
UCSRB=0x08;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x19;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1:  
Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// ADC initialization
```

```
// ADC Clock frequency: 1000,000 kHz
```

```
// ADC Voltage Reference: AVCC pin
```

```
// ADC High Speed Mode: Off
```

```
// ADC Auto Trigger Source: None
```

```
ADMUX=ADC_VREF_TYPE & 0xff;
```

```
ADCSRA=0x82;
```

```
SFIOR&=0xEF;
```

```
// Global enable interrupts
```

```
#asm("sei")
```

while (1)

//inisialisasi posisi servo awal

```
{  
kv++;  
if (kv==1) {  
servo_data=14;  
delay_ms(3000);  
b=pps;  
};
```

//input fuzzy error

```
av=pps;  
if (-13<av && av<=-9)  
{  
y1=1;  
};
```

```
if (-9<=av && av<=-4)  
{y1=-0.2*av-0.8;  
};
```

```
if (-4<=av && av <=0)  
{y1=0.25*av+1;  
};
```

```
if (0<av && av<=4)  
{y1=-0.25*av+1;  
};
```

```
if (4<av && av<=9)  
{y1=0.2*av-0.8;  
};
```

```

//input fuzzy delta error
del=av-b;

if(-22<=del && del <-10)
{y2=-0.083*del-0.83;
};

if(-10<=del && del<=0)
{y2=0.1*del+1;
};

if(0<del && del<=10)
{y2=-0.1*del+1;
};

if(10<del && del<=22)
{y2=0.083*del-0.83;
};

if (y1>=y2) {
y3=y1;
}
else {
y3=y2;
};

if(av < -4 && del <-10)
{t1=0;
};

if(av <-4 && del >= -10 && del<= 10)
{t1=0;
};

```

```
};
```

```
if(av < -4 && del > 10 )
```

```
{t1=3;
```

```
};
```

```
if(av >= -4 && av <= 4 && del < -10)
```

```
{t1=0;
```

```
};
```

```
if(av >= -4 && av <= 0 && del >= -10 && del <= 0)
```

```
{t1=3;
```

```
};
```

```
if(av > 0 && av <= 4 && del > 0 && del <= 10)
```

```
{t1=4;
```

```
};
```

```
if(av >= -4 && av <= 0 && del > 0 && del <= 10)
```

```
{t1=3;
```

```
};
```

```
if(av > 0 && av <= 4 && del >= -10 && del <= 0)
```

```
{t1=4;
```

```
};
```

```
if(av >= -4 && av <= 4 && del > 10 )
```

```
{t1=2;
```

```
};
```

```
if(av > 4 && del < -10)
```

```
{t1=4;
```

```
};
```

```
if(av > 4 && del >= -10 && del <= 10)
```

```
{t1=2;  
};
```

```
if(av >4 && del > 10 )  
{t1=2;  
};
```

```
if(t1==0)  
{ou=-3*y3-1;  
};
```

```
if(t1==3)  
{ou=y3-1;  
};
```

```
if(t1==4)  
{ou=-y3+1;  
};
```

```
if(t1==2)  
{ou=3*y3+1;  
};
```

```
b=av;  
servo_data=servo_data+ou;
```

```
};  
}
```