



**DESAIN SISTEM KONTROL POSISI DAN KESEIMBANGAN ROKET
DALAM MENJAGA SUDUT LUNCUR MENGGUNAKAN
PID KONTROLLER DAN *FUZZY LOGIC***

SKRIPSI

Oleh:

**Angger Panji Irwana
NIM 111910201076**

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



**DESAIN SISTEM KONTROL POSISI DAN KESEIMBANGAN ROKET
DALAM MENJAGA SUDUT LUNCUR MENGGUNAKAN
PID KONTROLLER DAN *FUZZY LOGIC***

SKRIPSI

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

Oleh:

**Angger Panji Irwana
NIM 111910201076**

1. Dosen Pembimbing Utama : Sumardi, S.T., M.T.
2. Dosen Pembimbing Anggota : Bambang Supeno, S.T., M.T.

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

PERSEMBAHAN

Syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Desain Sistem Kontrol Posisi Dan Keseimbangan Roket Dalam Menjaga Sudut Luncur Menggunakan PID Kontroller Dan *Fuzzy Logic*”. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Penulisan skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang tiada terhingga kepada:

1. Ayahku Imam Subandi, S.Pd., Ibunda Siti Samiati, Kakak Isti Pus Pita Wandansari, Adik Nikmatul Fitriyah, S.Pd., dan semua keluarga besar saya di Cluring yang selalu memberikan doa dan dukungan dari segi apapun, serta kasih sayang yang tidak pernah putus.
2. Pahlawan “Tanpa Tanda Jasa” ku di SDN 3 Cluring, SMPN 1 Cluring, SMKN 1 Glagah Banyuwangi, dan Universitas Jember, atas kesabaran dalam membimbing dan menyalurkan ilmunya.

MOTTO

“Sesungguhnya sesudah kesulitan itu ada kemudahan”

(QS, Al-Insyirah: 6)^{*)}

“Banyak kegagalan dalam hidup ini dikarenakan orang-orang tidak menyadari betapa dekatnya mereka dengan keberhasilan saat mereka menyerah”

(Thomas Alva Edison)^{**)}

“Semua Bisa Sampai Diatas Termasuk Yang Kecil”^{***)}

^{*)} Departemen Agama RI 2005. *Al-Quran dan Terjemahannya*. Jakarta: CV.

Penerbit J-ART

^{**)} Irwan, S. *Kumpulan Kata Bijak Thomas Alva Edison*. Jakarta: Rosdakarya.

^{***)} *Unknown*

PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Angger Panji Irwana

NIM : 111910201076

Menyatakan dengan sesungguhnya bahwa karya tulis yang berjudul “Desain Sistem Kontrol Posisi Dan Keseimbangan Roket Dalam Menjaga Sudut Luncur Menggunakan PID Kontroller Dan *Fuzzy Logic*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang saya sudah sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggungjawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 22 Juli 2016

Yang menyatakan,

Angger Panji Irwana

NIM 111910201076

SKRIPSI

**DESAIN SISTEM KONTROL POSISI DAN KESEIMBANGAN ROKET
DALAM MENJAGA SUDUT LUNCUR MENGGUNAKAN
PID KONTROLLER DAN *FUZZY LOGIC***

Oleh

**Angger Panji Irwana
NIM 111910201076**

Pembimbing :

Dosen Pembimbing Utama : Sumardi, S.T., M.T.

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

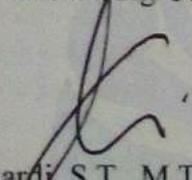
PENGESAHAN

Skripsi berjudul “**Desain Sistem Kontrol Posisi Dan Keseimbangan Roket Dalam Menjaga Sudut Luncur Menggunakan PID Kontroller Dan Fuzzy Logic**” telah diuji dan disahkan oleh Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember pada :

Hari : Rabu
Tanggal : 31 Agustus 2016
Tempat : Fakultas Teknik Universitas Jember

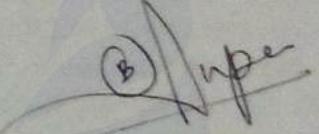
Menyetujui,

Dosen Pembimbing Utama


Sumardi, S.T., M.T.

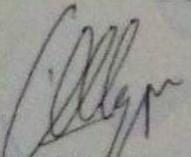
NIP. 19670113 199802 1 001

Dosen Pembimbing Anggota


Bambang Supeno, S.T., M.T.

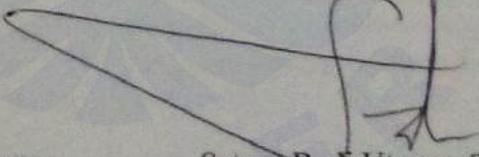
NIP. 19690630 199512 1 001

Dosen Penguji I


M. Agung Prawira Negara, S.T., M.T.

NIP. 19871217 201212 1 003

Dosen Penguji II


Satryo Budi Utomo, S.T., M.T.

NIP. 19850126 200801 1 002

Mengesahkan
Dekan Fakultas Teknik,



Dr. Ir. Entin Hidayah, M.UM.

NIP. 19661215 199503 2 001

Desain Sistem Kontrol Posisi Dan Keseimbangan Roket Dalam Menjaga Sudut Luncur Menggunakan PID Kontroller Dan Fuzzy Logic

Angger Panji Irwana

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Perkembangan teknologi roket di Indonesia sudah berkembang sejak lama. Pada saat ini teknologi roket mulai berkembang dikalangan mahasiswa melalui ajang kompetisi roket yang diadakan oleh DIKTI bersama LAPAN. Jenis roket yang berkembang dikalangan mahasiswa salah satunya adalah roket kecepatan rendah yang menggunakan daya dorong motor *brushless* EDF. Roket dengan daya dorong rendah memiliki tingkat akurasi keseimbangan yang rendah sehingga dibutuhkan suatu kontrol yang dapat menjaga posisi dan keseimbangan roket saat meluncur. Dalam pengaplikasiannya, roket mampu bergerak secara *autopilot* dengan meluncur keudara dengan mengontrol posisi dan keseimbangan sudut luncur melalui sirip roket. Metode yang digunakan yaitu PID kontroller dan *Fuzzy Logic* dilengkapi dengan sistem telemetri untuk memantau pergerakan roket melalui aplikasi *monitoring attitude* roket berdasarkan sudut *pitch*, *roll*, dan *yaw*. Data hasil pengujian sirip dapat bergerak bebas sebesar $+30^\circ$ sampai -30° dengan pergerakan sudut sebesar 5° . Data hasil pengujian awal roket dapat bertahan menjaga posisi dan keseimbangannya selama 2 detik di udara dengan sudut target yang terpenuhi sampai pada waktu 1000 *milisecond* pada sudut *pitch* dan sampai pada waktu 750 *milisecond* pada sudut *roll*. Data hasil pengujian akhir roket dapat bertahan menjaga posisi dan keseimbangannya selama 5 detik di udara dengan sudut target yang terpenuhi sampai pada waktu 3500 *milisecond* pada sudut *pitch* dan sampai pada waktu 2700 *milisecond* pada sudut *roll*. Roket gagal memenuhi target pada sudut *yaw* pada pengujian awal dan akhir karena kondisi roket yang melintir pada saat meluncur di udara.

Kata kunci: *Fuzzy Logic, kontrol PID, roll, roket, sirip roket, pitch, yaw*

Design Control System Of Position And Balance The Rocket In Maintaining Angle Sled Using PID Controller And Fuzzy Logic

Angger Panji Irwana

Department of Electrical Engineering, Faculty of Engineering. University of Jember

ABSTRACT

The development of the rocket technology in Indonesia had been growing for a long time. At this time the rocket technology will began to develop among students through a rocket competition that organized by the Higher Education with LAPAN. The type of rocket that develops among students is low-speed rockets that use the energy of EDF brushless motors. Rocket with lower energy levels has low accuracy of balance, so it takes a control that keep the rocket position and balancing while sliding. In its application, the rocket is capable of moving on autopilot by gliding into the air to control the position and angle of the glide through the balancing of the rocket fins. The method that used is a PID control and Fuzzy Logic were equipped with a telemetry system that will monitor the movement of the rocket through the rocket monitoring attitude applications based on the point pitch, roll, and yaw. The data of fins test result, can be moved freely from + 30° to -30° with the movement angle of 5°. The data early of the rocket test results, can survive to maintain position and balancing for 2 seconds in the air that will happen until the target angle has met at the time of 1000 milliseconds in the corner of the pitch and up to the time of 750 milliseconds on a rolling angle. The data final of the rocket test results, can survive to maintain position and balancing for 5 seconds in the air that will happen until the target angle has met at the time of 3500 milliseconds in the corner of the pitch and up to the time of 2700 milliseconds on a rolling angle. The rocket has failed to meet targets on yaw angle because of the condition of the rocket that spinned when its gliding through the air.

Key words: Fuzzy Logic, PID control, roll, rocket, rocket fin, pitch, yaw

RINGKASAN

Desain Sistem Kontrol Posisi Dan Keseimbangan Roket Dalam Menjaga Sudut Luncur Menggunakan PID Kontroller Dan *Fuzzy Logic*; Angger Panji Irwana, 111910201076; 2016; 134 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Perkembangan teknologi roket di Indonesia sudah berkembang sejak lama. Pada saat ini teknologi roket mulai berkembang dikalangan mahasiswa melalui ajang kompetisi roket yang diadakan oleh DIKTI bersama LAPAN. Jenis roket yang berkembang dikalangan mahasiswa salah satunya adalah roket kecepatan rendah yang menggunakan daya dorong motor *brushless* EDF. Roket dengan daya dorong rendah memiliki tingkat akurasi keseimbangan yang rendah sehingga dibutuhkan suatu kontrol yang dapat menjaga posisi dan keseimbangan roket saat meluncur. Pada sistem kendali roket digunakan sensor-sensor untuk mengukur parameter-parameter meteorologi dan perilaku gerak roket diudara.

Sistem kontrol dirancang agar roket mampu mengenali posisi dan kondisi terhadap gerakannya saat meluncur di udara, mampu mengoreksi keseimbangannya dalam meningkatkan akurasi dalam mencapai posisi dan keseimbangannya sesuai dengan perintah yang telah ditentukan. Sistem kontrol akan bekerja secara *autopilot* tanpa menggunakan *remote control*. Sistem yang digunakan adalah PID kontroller dan *Fuzzy Logic* untuk melakukan penyeimbangan posisi dan kondisi roket. Kontrol PID sering digunakan untuk membenahi kondisi nilai *error* dari suatu sistem seperti pada motor servo, sedangkan *Fuzzy Logic* memiliki penalaran logika yang mampu memberikan pengambilan keputusan dari logika dengan aturan-aturan yang diperintahkan dalam menangani suatu sistem. Metode ini cocok untuk sistem kontrol yang berbasis sudut kemiringan dalam sebuah wahana terbang. Berdasarkan hal tersebut maka dirancang sebuah sistem kontrol posisi dan keseimbangan *prototype* roket yang menggunakan daya dorong motor *brushless* EDF dilengkapi dengan

sistem telemetri sebagai pemantau data berupa aplikasi *ground segment monitoring attitude* roket.

Tujuan penelitian ini yaitu untuk merancang sistem kontrol yang dapat mengendalikan roket saat meluncur di udara dengan menjaga sudut posisi dan keseimbangan. Hal tersebut dilakukan untuk membuat roket bisa kembali ke posisi dan keseimbangannya semula ketika hilang kendali.

Hasil analisis data dalam pengujian roket saat belum uji luncur, roket dapat mengontrol posisi dan keseimbangannya dengan besar rata-rata kesalahan 6,29% pada sudut *pitch*, 4,26% pada sudut *roll*, dan 6,85% pada sudut *yaw*. Sedangkan pada hasil pengujian awal saat meluncur roket dapat bertahan di udara selama 2 detik dengan *range* sudut 10° dengan target yang terpenuhi sampai pada 1000 *milisecond* pada sudut *pitch* dengan target sudut target 80° , untuk sudut *roll* sampai pada 750 *milisecond* dengan sudut target 90° , dan roket gagal untuk memenuhi target pada sudut *yaw* dengan target sudut 0° (360°) yang diakibatkan oleh roket yang mengalami kondisi melintir saat meluncur. Dari data hasil pengujian awal menghasilkan nilai rata-rata *error* 32,25 %.

Data hasil pengujian akhir roket dapat bertahan di udara selama 5 detik dengan kondisi target sudut yang terpenuhi sampai pada 3500 *milisecond* pada sudut *pitch* dengan target sudut 80° , 2700 *milisecond* pada sudut *roll* dengan target sudut 90° , dan pada sudut *yaw* roket gagal untuk memenuhi target sudut 0° (360°). Dari data hasil pengujian akhir menghasilkan nilai rata-rata *error* 25,61 %. Berdasarkan analisis pengujian roket mengalami kondisi melintir saat meluncur, kondisi ini disebabkan oleh putaran motor *brushless* EDF yang besar. Sedangkan roket tidak dapat bertahan lama saat meluncur diakibatkan interferensi angin yang tidak dapat dikontrol oleh roket.

PRAKATA

Puji syukur kehadirat Allah SWT, atas hidayahnya dan rahmatnya sehingga penyusun dapat menyelesaikan skripsi ini sebagaimana mestinya. Shalawat serta salam semoga Allah SWT limpahkan kepada Nabi Muhammad SAW sebagai sumber inspirasi dan membuat penyusun lebih kuat dan menatap setiap hal yang penuh optimis dan berfikir positif, dalam menunjang kemampuan penyusun dalam menjalani persaingan globalisasi kerja nantinya.

Dalam pelaksanaannya kami tidak lepas dari kesulitan dan permasalahan dalam penyusunan skripsi ini, baik dari proses pembuatan proposal sampai penyusunan akhir skripsi, mengenai ilmu yang bermanfaat, moral dan sikap serta tanggung jawab dalam menyelesaikan skripsi ini. Dengan demikian kami mengucapkan terima kasih pada:

1. Ibu Dr.Ir. Entin Hidayah, M.UM. selaku Dekan Fakultas Teknik Universitas Jember.
2. Bapak Dr. Bambang Sri Kaloko, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.
3. Bapak Sumardi, S.T., M.T. selaku Dosen Pembimbing Utama sekaligus sebagai Dosen Pembimbing Akademik yang telah meluangkan banyak waktu, pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
4. Bapak Bambang Supeno, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan banyak waktu, pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
5. Bapak M. Agung Prawira N, S.T., M.T. selaku Dosen Penguji I yang telah meluangkan waktu, pikiran dan perhatiannya guna memberikan pengarahan demi terselesaikannya skripsi ini.
6. Bapak Satriyo Budi Utomo, S.T., M.T. selaku Dosen Penguji II yang telah meluangkan waktu, pikiran dan perhatiannya guna memberikan pengarahan demi terselesaikannya skripsi ini.

7. Keluarga Besar Teknik Elektro 2011 yang dengan bangga menggunakan identitas “GATEL (Gerombolan Teknik Elektro 2011)” berterima kasih sekali menjadi anggota dahsyat kalian.
8. Keluarga Besar Elektronika dan Sistem Kendali Elektro 2011 yang sudah mau diajak berdiskusi ilmu selama kuliah khususnya di bidang konsentrasi.
9. Keluarga besar kos Moersid 17, terimakasih atas kebersamaannya selama di Jember ini.
10. Pihak-pihak yang telah membantu penulis menyelesaikan skripsi ini yang tidak bisa penulis sebut satu persatu.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu Teknik Elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya.

Jember, 22 Juli 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
ABSTRAK	vii
ABSTRACT	viii
RINGKASAN	ix
PRAKATA	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR PERSAMAAN	xix
DAFTAR TABEL	xx
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 Raket	4
2.1.1 Bagian Dan Fungsi Raket.....	5
2.1.2 Prinsip Kerja Raket	6
2.1.3 Raket Kendali.....	7
2.2 Fuzzy Logic	8

2.2.1 Konsep <i>Fuzzy Logic</i>	10
2.2.2 Himpunan <i>Fuzzy</i>	10
2.2.3 Fungsi Keanggotaan(<i>Membership Function</i>)	11
2.3 Proportional Integral Derivative (PID)	13
2.3.1 Kontroler <i>Proportional</i> (P).....	14
2.3.2 Kontroler <i>Integral</i> (I).....	15
2.3.3 Kontroler <i>Derivative</i> (D)	17
2.4 Sensor	19
2.4.1 Sensor <i>Accelerometer</i>	19
2.4.2 Sensor <i>Gyroscope</i>	20
2.4.3 Sensor <i>Magnetometer</i>	22
2.5 Arduino	22
2.5.1 Spesifikasi Arduino Uno	23
2.5.2 Bahasa Pemrograman Arduino	24
2.6 Komunikasi Serial	24
2.7 Sistem Telemetri	25
2.8 Software Processing	26
2.9 Motor Brushless	27
2.10 Motor Servo	28
2.10.1 Jenis Motor Servo.....	29
2.10.2 Pensinyalan Motor Servo	30
2.11 Modul Komunikasi Wireless Xbee Pro S2B	31
BAB 3. METODOLOGI PENELITIAN	33
3.1 Waktu Dan Tempat Penelitian	33
3.2 Alat dan Bahan	33
3.2.1 <i>Hardware</i>	33
3.2.2 <i>Software</i>	34
3.3 Tahapan Penelitian	34
3.4 Perancangan Sistem	36

3.5 Perancangan <i>Prototype</i> Roket	37
3.6 Perancangan Elektronik	40
3.7 Perancangan Perangkat Lunak	47
3.7.1 <i>Flow Chart</i> Perancangan Sistem.....	47
3.7.2 Perancangan <i>Software Monitor Attitude</i> Roket.....	52
3.7.3 Perancangan Sistem Kontrol PID dan <i>Fuzzy Logic</i>	56
BAB 4. HASIL DAN PEMBAHASAN	65
4.1 Pengujian dan Analisa <i>Hardware</i> Sistem Kontrol Posisi dan Keseimbangan Roket	65
4.1.1 Kalibrasi dan Pengujian Sensor.....	66
4.1.2 Pengujian dan Analisa Pergerakan Sayap Roket.....	75
4.2 Pengujian dan Analisa Sistem <i>Telemetry</i> Roket dan <i>Interface Ground Segment</i> Roket	79
4.2.1 Konfigurasi Komunikasi <i>Wireless</i> Modul Xbee Pro S2B.....	79
4.2.2 Pengujian dan Analisa Sistem <i>Telemetry</i>	79
4.2.3 Pengujian dan Analisa <i>Interface Ground Segment</i>	81
4.3 Pengujian dan Analisa Sistem Kontrol Posisi dan Keseimbangan Roket	82
4.3.1 Pengujian Sistem Kontrol Tanpa Uji Terbang.....	82
4.3.2 Pengujian Sistem Kontrol Roket Uji Awal	89
4.3.3 Pengujian Sistem Kontrol Roket Akhir.....	94
BAB 5. PENUTUP	105
5.1 Kesimpulan	105
5.2 Saran	106
DAFTAR PUSTAKA	107
LAMPIRAN	109

DAFTAR GAMBAR

2.1	Roket	4
2.2	Representasi linear naik	11
2.3	Representasi linear turun.....	11
2.4	Kurva segitiga	12
2.5	Kurva trapesium.....	12
2.6	Blok diagram kontroler PID.....	13
2.7	Diagram blok kontroler <i>proportional</i>	14
2.8	Diagram blok kontroler <i>integral</i>	15
2.9	Kurva sinyal kesalahan terhadap sinyal kontrol	16
2.10	Diagram blok kontroler <i>derivative</i>	18
2.11	Sikap sudut orientasi <i>gyroscope</i>	21
2.12	Arduino Uno	22
2.13	Letak rotor di dalam dan di luar rotor.....	27
2.14	Motor servo.....	28
2.15	Sistem mekanik motor servo	29
2.16	Pengaruh lebar pulsa terhadap posisi.....	30
2.17	Modul <i>wireless</i> Xbee Pro S2B.....	31
3.1	Perancangan sistem roket kendali	36
3.2	Perancangan roket kendali	37
3.3	Perancangan parasut.....	38
3.4	Hidung (<i>nose</i>).....	38
3.5	Bagian tengah roket	39
3.6	Penyambungan servo dengan sirip.....	39
3.7	Bagian bawah roket.....	39
3.8	Diagram blok perancangan sistem kendali roket	40
3.9	Blok diagram <i>power supply</i>	41
3.10	ESC <i>brushless</i> motor	42

3.11	Motor <i>brushless</i> EDF 70 mm	42
3.12	Modul CMPS11	43
3.13	Konfigurasi I2C sensor CMPS11	43
3.14	Rangkaian Arduino dengan CMPS11.....	45
3.15	IMU (<i>inertial measuring unit</i>) arduino pro mini	46
3.16	Xbee Pro S2B dan pemasangan pada IMU.....	47
3.17	<i>Flow chart</i> program kontrol posisi dan keseimbangan.....	47
3.18	<i>Flow chart</i> program kontrol posisi dan keseimbangan (lanjutan)	48
3.19	<i>Inisialisasi</i> masukan dan keluaran roket.....	50
3.20	Proses pengoahan data CMPS11	51
3.21	Program masukan dan keluaran pada <i>setup</i>	51
3.22	Program kontrol keluaran servo	52
3.23	Program sistem pengiriman data	52
3.24	Perancangan aplikasi data <i>monitoring attitude</i> roket EDF	53
3.25	<i>Inisialisasi</i> pada <i>software</i> Processing	54
3.26	Pengaturan <i>instrument visual</i>	54
3.27	Komunikasi serial	55
3.28	<i>Listing</i> program <i>instrument</i> roket	55
3.29	Pengaturan <i>instrument visual</i> roket	55
3.30	Blok diagram sistem kontrol PID	56
3.31	Grafik respon sistem PID.....	57
3.32	Fungsi keanggotaan <i>pitch</i>	59
3.33	Fungsi keanggotaan <i>roll</i>	60
3.34	Fungsi keanggotaan <i>yaw</i>	61
3.35	Fungsi keanggotaan keluaran kondisi roket	62
4.1	Grafik hasil pengujian sensor kompas	69
4.2	Grafik hasil pengujian sudut kemiringan <i>pitch</i>	72
4.3	Grafik hasil pengujian sudut kemiringan <i>roll</i>	74
4.4	Grafik hasil pengujian sudut kemiringan <i>yaw</i>	75

4.5	Kondisi kemiringan sayap roket	78
4.6	Pengujian aplikasi GS dengan roket	81
4.7	Kondisi roket saat digantung	83
4.8	Kondisi roket saat diberi sumber angin	83
4.9	Grafik hasil pengujian tanpa uji terbang sudut kemiringan <i>pitch</i>	85
4.10	Grafik hasil pengujian tanpa uji terbang sudut kemiringan <i>roll</i>	87
4.11	Grafik hasil pengujian tanpa uji terbang sudut kemiringan <i>yaw</i>	88
4.12	Grafik hasil pengujian awal sudut kemiringan <i>pitch</i>	90
4.13	Grafik hasil pengujian awal sudut kemiringan <i>roll</i>	92
4.14	Grafik hasil pengujian awal sudut kemiringan <i>yaw</i>	93
4.15	Kondisi roket saat lepas kendali	94
4.16	Grafik hasil pengujian akhir sudut kemiringan <i>pitch</i>	96
4.17	Grafik hasil pengujian akhir sudut kemiringan <i>roll</i>	98
4.18	Grafik hasil pengujian akhir sudut kemiringan <i>yaw</i>	100
4.19	Kondisi sirip roket saat roket miring	100
4.20	Perancangan roket pengujian awal	101
4.21	Perancangan roket pengujian akhir.....	102

DAFTAR PERSAMAAN

2.1 Rumus fungsi keanggotaan representasi linear naik	11
2.2 Rumus fungsi keanggotaan representasi linear turun	12
2.3 Rumus fungsi keanggotaan kurva segitiga.....	12
2.4 Rumus fungsi keanggotaan kuva trapesium.....	12
2.5 Persamaan kontroler proporsional	15
2.6 Persamaan $u(t)$ kontroler integral.....	17
2.7 Persamaan u kontroler integral	17
2.8 Persamaan derivatif.....	19
2.9 Rumus sudut <i>pitch</i>	21
2.10 Rumus sudut <i>roll</i>	21
2.11 Persamaan besar momentum	22
2.12 Persamaan momentum sudut.....	22

DAFTAR TABEL

3.1	<i>Datasheet</i> sensor CMPS11	44
3.2	<i>Datasheet</i> sensor CMPS11 (lanjutan).....	45
4.1	Data hasil kalibrasi sensor <i>magnetometer</i>	67
4.2	Data hasil pengujian sensor kompas roket.....	68
4.3	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>pitch</i>	70
4.4	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>pitch</i> (lanjutan)	71
4.5	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>roll</i>	72
4.6	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>roll</i> (lanjutan)..	73
4.7	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>yaw</i>	74
4.8	Data hasil pengujian kemiringan roket terhadap sudut gerak <i>yaw</i> (lanjutan)..	75
4.9	Data hasil pengujian kemiringan sayap roket bagian bawah	76
4.10	Data hasil pengujian kemiringan sayap roket bagian atas	77
4.11	Data hasil pengujian jarak komunikasi modul Xbee Pro S2B.....	80
4.12	Data hasil pengujian sistem kontrol roket tanpa uji terbang terhadap sudut <i>pitch</i>	84
4.13	Data hasil pengujian sistem kontrol roket tanpa uji terbang terhadap sudut <i>roll</i>	85
4.14	Data hasil pengujian sistem kontrol roket tanpa uji terbang terhadap sudut <i>roll</i> (lanjutan)	86
4.15	Data hasil pengujian sistem kontrol roket tanpa uji terbang terhadap sudut <i>yaw</i>	87
4.16	Data hasil pengujian sistem kontrol roket tanpa uji terbang terhadap sudut <i>yaw</i> (lanjutan)	88
4.17	Data hasil pengujian sistem kontrol awal terhadap sudut kemiringan <i>pitch</i>	90
4.18	Data hasil pengujian sistem kontrol awal terhadap sudut kemiringan <i>roll</i> ...	91
4.19	Data hasil pengujian sistem kontrol awal terhadap sudut kemiringan <i>roll</i> (lanjutan).....	92

4.20	Data hasil pengujian sistem kontrol awal terhadap sudut kemiringan <i>yaw</i> ..	93
4.21	Data hasil pengujian sistem kontrol posisi dan keseimbangan roket terhadap sudut <i>pitch</i>	95
4.22	Data hasil pengujian sistem kontrol posisi dan keseimbangan roket terhadap sudut <i>roll</i>	96
4.23	Data hasil pengujian sistem kontrol posisi dan keseimbangan roket terhadap sudut <i>roll</i> (lanjutan).....	97
4.24	Data hasil pengujian sistem kontrol posisi dan keseimbangan roket terhadap sudut <i>yaw</i>	98
4.25	Data hasil pengujian sistem kontrol posisi dan keseimbangan roket terhadap sudut <i>yaw</i> (lanjutan).....	99

BAB I. PENDAHULUAN

1.1 LATAR BELAKANG

Indonesia sedang mengembangkan teknologi untuk mendukung sistem informasi dan pertahanan. Salah satunya melalui teknologi penerbangan dan antariksa merupakan teknologi yang selalu berkembang dari zaman ke zaman. Salah satunya tentang teknologi kendali roket yang dikembangkan untuk mengorbit satelit ataupun untuk alutsita (alat utama sistem senjata) sebagai pertahanan.

Saat ini perkembangan teknologi roket sudah mulai berkembang dikalangan mahasiswa melalui ajang kompetisi yang diadakan oleh DIKTI bersama LAPAN. Sebagai wujud untuk mengembangkan teknologi pada mahasiswa kegiatan ini selalu diadakan setiap tahunnya dengan nama KOMURINDO yang diselenggarakan pertama kali pada tahun 2009 dengan kategori Roket Muatan dan Roket EDF.

Roket EDF merupakan salah satu obyek elektronika yang saat ini tengah berkembang dalam kompetisi komurindo. Pada roket hal yang diutamakan adalah tentang sistem kendalinya dalam menjaga sikap meluncurnya ke udara. Pada sistem kendali roket digunakan sensor-sensor untuk mengukur parameter-parameter meteorologi dan perilaku gerak roket diudara. Untuk itu, roket dilengkapi dengan sensor seperti: sensor *accelerometer*, sensor *gyroscope*, dan sensor *magnetometer*. Untuk menjamin kehandalan dari sistem kendali dari roket maka sistem kendali harus bisa memberikan data sudut orientasi secara akurat.

Dalam menghasilkan luaran sudut orientasi terhadap bumi digunakan sensor *accelerometer*, namun pada sensor ini rentang terhadap noise getaran (Alfiansyah, 2012). Sensor kompas digital dan sensor *magnetometer* pada hakikatnya memiliki prinsip kerja yang sama. Sensor kompas digital merupakan modul sensor

magnetometer dengan keluaran berupa sudut yang menyatakan arah hadap. Sedangkan sensor *magnetometer* keluarannya berupa besar medan magnet bumi yang diukur dalam tiga sumbu yang dapat digunakan untuk menentukan sudut arah hadap dengan rumusan tertentu (Noor, 2013). Untuk pemenuhan kontrol navigasi, luaran dari sensor dibutuhkan sudut global dengan urutan rotasi *roll*, *pitch*, dan *yaw*.

Implementasi roket pada sistemnya tidak hanya daya dorong yang ada pada roket saja yang diperhatikan, *sensor tracking*, *satelite controlling*, dan sirip kendali roket juga dikembangkan sebagai kontrol utama roket dalam menjaga sudut posisi dan keseimbangan roket saat di udara. Dengan data kemiringan sudut yang diirinkan dari *autopilot* roket diharapkan roket dapat bergerak dengan menjaga posisi dan keseimbangannya.

Berdasarkan penelitian sebelumnya yang berjudul “Implementasi Sensor Magnetometer dan Akselerometer Untuk Memonitor Arah Muatan Roket” dan “Rancang Bangun Prototipe Sistem Aktuator Sirip Roket Menggunakan Motor Servo”. Penelitian skripsi ini adalah untuk mendesain dan membangun sistem kontrol keseimbangan pada roket kecepatan rendah yang mampu menjaga sudut luncur pada saat meluncur hingga dapat mencapai waktu yang sudah ditentukan tanpa harus mengalami gagal meluncur karena pergerakan roket tidak terkendali. Dengan menggunakan kontroler *Proportional Integral Derivative* (PID) dan *Fuzzy Logic* dilengkapi dengan sistem telemetri untuk memantau pergerakan roket melalui *ground segment* yang menampilkan informasi dari pembacaan sensor terhadap sudut orientasi *roll*, *pitch*, dan *yaw* yang dikirimkan melalui komunikasi tanpa kabel secara *real-time*.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang, dapat diidentifikasi beberapa masalah yang harus diselesaikan antara lain:

- a. Bagaimana sistem kontrol roket agar dapat dikendalikan melalui *fin* atau sirip pada roket?

b. Bagaimana sistem kontrol posisi dan keseimbangan sudut luncur roket?

1.3 BATASAN MASALAH

Permasalahan dalam tugas akhir ini dibatasi oleh beberapa hal sebagai berikut:

- a. Roket yang digunakan adalah roket kecepatan rendah yang menggunakan daya dorong dari motor EDF.
- b. Sistem kendali yang dirancang hanya untuk menjaga posisi dan keseimbangan roket.
- c. Roket meluncur di udara dengan batas waktu 10 detik dalam mengikuti lintasannya.
- d. Sudut kemiringan pada *pitch* sebesar 80° , pada *roll* 90° , dan pada *yaw* 0° .
- e. *Range* sudut kemiringan dalam kondisi seimbang sebesar 10° .

1.4 TUJUAN PENELITIAN

Tujuan dalam tugas akhir ini adalah sebagai berikut :

- a. Merancang sistem kontrol roket agar dapat dikendalikan melalui *fin* atau sirip pada roket.
- b. Merancang sistem kontrol posisi dan keseimbangan sudut luncur roket.

1.5 MANFAAT

Hasil dari tugas akhir ini diharapkan bermanfaat untuk menjadi dasar penelitian tentang kontrol posisi dan keseimbangan pada roket dan perkembangan teknologi dibidang penerbangan khususnya tentang roket kecepatan rendah.

BAB 2. TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan beberapa bagian-bagian yang menjadi dasar dalam pembuatan *prototype* dan sistem kontrol posisi dan keseimbangan roket.

2.1 ROKET

Roket adalah teknologi penerbangan yang sering digunakan dalam wahana luar angkasa, peluru kendali, atau kendaraan terbang, yang mendapatkan dorongan melalui reaksi terhadap bahan fluida dari keluaran mesin roket atau dari daya dorong dari energi kinetik. Aksi dari keluaran dalam ruang bakar dan *nozzle* pengembang itu mampu membuat gas mengalir dengan kecepatan supersonik, sehingga menimbulkan dorongan reaktif yang besar bagi roket untuk melaju.



Gambar 2.1 Roket

(Sumber : <http://id.wikipedia.org/wiki/Roket>)

Nama roket berasal dari bahasa Italia *rocchetta* yang artinya sekering kecil, sebutan untuk petasan kecil yang diciptakan Ludovico Antonio Muratori, seorang ilmuwan Italia, pada tahun 1379. Asal mula roket berawal pada abad ke-9, ketika para ahli kimia Cina menemukan bubuk hitam (*mesiu*) ketika sedang berusaha

menemukan obat awet muda (*elixir of immortality*). Penemuan yang terjadi secara kebetulan itu kemudian mengarah pada percobaan dalam bentuk senjata seperti bom, meriam, dan panah api pembakar, yang menggunakan bubuk hitam tersebut sebagai daya picunya.

Roket pada dasarnya adalah mesin untuk alat transportasi seperti mesin jet, diesel, dan lain-lain. Tapi, berbeda dengan mesin transportasi lain, roket bersifat anaerob. Untuk melakukan pembakaran bahan bakar ia membawa oksigen sendiri, sehingga praktis tak membutuhkan oksigen dari luar. Karenanya, roket dapat digunakan sebagai mesin transportasi ke ruang angkasa yang tak beroksigen. Pada umumnya, roket terdiri dari tiga bagian. Bagian pembawa muatan, pengendali, dan bagian mesin. Bagian pembawa muatan berfungsi untuk mengangkut barang satelit, objek lainnya, hingga bahan peledak. Bagian pengendali merupakan bagian di mana terdapat piranti untuk mengendalikan roket. Dan, bagian mesin, merupakan bagian di mana terdapat mesin serta bahan bakar roket. Sebagai catatan, mesin roket ini terbagi dalam dua kelompok, tergantung dari jenis bahan bakarnya: cair dan padat.

2.1.1 Bagian dan Fungsi Roket

Roket merupakan wahana luar angkasa, peluru kendali, atau kendaraan terbang yang menghasilkan dorongan melalui reaksi pembakaran dari mesin roket. Dorongan ini terjadi karena reaksi cepat pembakaran/ledakan dari satu atau lebih bahan bakar yang dibawa dalam roket. Seringkali definisi roket digunakan untuk merujuk kepada mesin roket. Bagian dan fungsi mesin roket, sebagai berikut.

1. Hidung (*Nose*)

Bagian paling depan yang biasanya diisi hulu ledak muatan ilmiah atau peralatan indera atau kendali yang berbentuk tabung silindris (*cylinder*) yang berguna dalam muatan. Badan utama roket yang biasanya diisi bahan bakar dan peralatan bakarnya.

2. Ekor (*Tail*)

Bagian paling belakang berisi saluran sumber pembakaran (*nozzle*) mekanisme pengendalian atau bagian dimana sistem energi pendorong ditempatkan. Seperti menggunakan teknologi jet atau motor. Pada ekor ditempatkan sirip yang berada bagian ujung ekor.

3. Sirip (*Fin* atau *Stabilizer*)

Alat kendali aerodinamik, yang berfungsi sebagai pemberi kemudi maupun kestabilan roket dalam meluncur ke udara. Sirip dapat ditambahkan dengan bentuk memanjang lebih jauh ke belakang, berguna dalam membantu menjamin stabilitas dari roket dalam kestabilan meluncur. Namun, sirip yang panjang menyebabkan roket jatuh dengan kecepatan lebih tinggi, sehingga dapat menyebabkan benturan keras saat bertabrakan dengan obyek lain. Roket yang tidak memiliki parasut dalam mendarat akan mengalami kejadian kerusakan tersebut. Dalam menghindarinya roket dilengkapi sistem parasut untuk mendarat di bumi dengan dilengkapi sistem kontrol yang dimiliki roket. Sirip berfungsi juga sebagai kendali dari roket dalam melakukan manuver di udara untuk melakukan belokan ataupun melakukan pergerakan berputar.

2.1.2 Prinsip Kerja Roket

Perkembangan teknologi sekarang ini semakin berkembang sangat pesat, salah satunya teknologi roket yang saat ini terus berkembang. Roket adalah sejenis sistem propulsi yang membawa bahan bakar dan oksigennya sendiri. Dorongan pada roket merupakan penerapan yang menarik dari hukum III Newton dan Kekekalan momentum yaitu dengan memancarkan aliran massa hasil pembakaran propelan. Rocket dapat meluncur secara vertikal keatas karena roket memiliki mesin pendorong yang dapat memberikan kecepatan pada roket. Gaya dorong roket bekerja berdasarkan impuls yang diberikan oleh mesin roket.

Prinsip propulsi roket akan dianalogikan dengan menggunakan roket air sederhana. Prinsipnya adalah botol akan meluncur bila botol diberi tekanan udara yang tinggi dari pompa dan di dalamnya diberi sedikit air untuk menghasilkan tenaga

semburan yang lebih besar. Prinsip kerja roket ini merupakan penerapan dari hukum ketiga Newton dan kekekalan momentum. Dasar hukum roket air adalah hukum Newton 3 yang berbunyi “Apabila sebuah benda memberikan gaya kepada benda lain, maka benda kedua memberikan gaya kepada benda yang pertama. Kedua gaya tersebut memiliki besar yang sama tapi berlawanan arah”.

Teori dasar peluncuran roket air, sama dengan percobaan balon yang meluncur ke atas. Roket air memberikan gaya aksi yang sangat besar kepada gas, dengan mendorong gas keluar, dan gas tersebut memberikan gaya reaksi yang sama besar, dengan mendorong roket air ke atas. Roket air mendorong gas ke bawah, gas mendorong roket air ke atas. Inilah yang disebut hukum aksi-reaksi atau Newton 3. Berdasarkan kekekalan momentum, kelajuan akhir yang dapat dicapai sebuah roket bergantung pada banyaknya bahan bakar yang dapat dibawa oleh roket dan kelajuan pancaran gas. Ketika bahan bakar tahap pertama telah terbakar habis, roket ini dilepaskan begitu seterusnya, sehingga pesawat-pesawat antariksa yang pergi ke luar angkasa dapat terbang tinggi meninggalkan bumi. Banyaknya *stage* atau tahapan tergantung kebutuhan kelajuan pada misi roket itu sendiri.

2.1.3 Roket Kendali

Roket dikendalikan oleh sistem yang mengatur pergerakan sirip. Untuk mengatur kecepatannya, roket menggunakan pendorong atau *booster*. Ketika roket diluncurkan, saat itu juga arah roket diatur. Walaupun kecepatan awal roket ditingkatkan dengan menggunakan pendorong, saat kecepatan terbangnya masih rendah, sulit untuk mendapatkan hasil yang cukup signifikan dalam menggerakkan roket akibat pergerakan sirip. Pergerakan sirip dapat mengakibatkan adanya gaya aerodinamis yang dapat mengubah arah dari roket tersebut (Ariel, 2011).

Target yang dituju adalah target diam yang data posisinya akan digunakan sebagai acuan untuk *autopilot* dalam mengarahkan roket. *Autopilot* adalah alat atau berupa kontrol yang akan mengirimkan perintah untuk menggerakkan sirip. Ada 4 sirip yang berada pada roket yang dikendalikan oleh aktuator, dan masing-masing

aktuator dikendalikan secara independen akibat penyimpangan pada permukaan yang terjadi, penyimpangan sirip pada umumnya bergerak kurang dari 10° . Simpangan sirip sebesar 1° mampu membuat perubahan arah roket cukup signifikan, sehingga pergerakan sirip pada roket tidak lebih dari 10° . Pergerakan sirip harus mampu menahan beban angin, gravitasi, dan kecepatan yang dihasilkan oleh pendorong (Anwar, 2012).

Pengembangan wahana untuk roket kendali terdiri dari beberapa aspek terkait, diantaranya:

1. Penentuan misi. Misi terbang roket, seperti terbang jelajah, terbang dari darat ke udara, terbang dari udara ke udara, dan dari udara ke darat.
2. Desain Aerodinamik. Penentuan bentuk sirip, posisi, dan hal-hal yang berkaitan dengan bentuk aerodinamik untuk menaikkan performa roket.
3. Pembuatan struktur. Pemilihan material sesuai dengan berat yang diinginkan, struktur roket yang akan dibuat, penentuan bentuk *nose* serta sirip, penyambungan sirip, dan mekanisme kendali sirip.
4. Instrumentasi sistem kendali. Perangkat keras dan perangkat lunak yang digunakan.
5. Sistem indra dinamik. Pada tahap pengujian, sistem ini diperlukan untuk mendapatkan data dari gerak roket yang akan digunakan untuk memantau kinerja roket, seperti menggunakan sistem *telemetry* dan *ground segment*.
6. Alat dorong. Roket menggunakan alat dorong yang digunakan untuk mendorong gerak roket. Alat dorong yang digunakan bisa berupa bahan bakar seperti propulsi atau menggunakan non bahan bakar berupa motor *brushless* EDF.
7. Uji validasi. Untuk memastikan semua sistem bekerja dengan baik.

2.2 Fuzzy Logic

Konsep logika *fuzzy* pertama kali diperkenalkan oleh Professor Lotfi A.Zadeh dari Universitas California, Berkeley pada bulan Juni 1965. Dalam kamus Oxford,

istilah *fuzzy* didefinisikan sebagai *blurred* (kabur atau remang-remang), *indistinct* (tidak jelas), *imprecisely defined* (didefinisikan secara tidak presisi), *confused* (membingungkan), *vague* (tidak jelas). Yang dimaksud sistem *fuzzy* adalah sebuah sistem yang dibangun dengan definisi, cara kerja, dan deskripsi yang jelas berdasar pada teori logika *fuzzy*.

Secara umum, *fuzzy logic* adalah sebuah metodologi berhitung dengan variabel kata-kata (*linguistic variable*), sebagai pengganti berhitung dengan bilangan. Kata-kata yang digunakan dalam *fuzzy* memang tidak sepresisi bilangan, namun kata-kata jauh lebih dekat dengan intuisi manusia. Manusia bisa langsung merasakan nilai dari variabel kata-kata yang sudah dipakainya sehari-hari.

Dengan *fuzzy logic*, sistem kepakaran manusia bisa diimplementasikan kedalam bahasa mesin secara mudah dan efisien. Bahasa presisi yang diperlukan mesin sulit dirasakan oleh manusia yaitu kurang bermakna dari sudut pandang bahasa manusia. Sebaliknya, variabel kata-kata bisa lebih simpel, singkat, dan langsung dapat dirasakan manusia, namun kurang presisi dari sudut pandang bahasa mesin. Peran sistem *fuzzy* yaitu untuk menjembatani komunikasi sehingga menjadi lebih efektif dan efisien antara mesin dan manusia.

Ada beberapa alasan digunakannya *fuzzy logic*, antara lain (Pudjo Prabowo, 2012):

1. Konsep *fuzzy logic* mudah dimengerti, karena konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. *Fuzzy logic* sangat fleksibel
3. *Fuzzy logic* memiliki toleransi terhadap data-data yang tidak tepat.
4. *Fuzzy logic* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. *Fuzzy logic* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.

6. *Fuzzy logic* dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
7. *Fuzzy logic* didasarkan pada bahasa alami.

2.2.1 Konsep *Fuzzy Logic*

Konsep *fuzzy logic* adalah memetakan sebuah ruang *input* ke dalam ruang *output* dengan menggunakan *IF-THEN rules*. Pemetaan *fuzzy* merupakan suatu nilai yang dapat bernilai benar atau salah secara bersamaan. Namun seberapa besar nilai kebenaran dan kesalahannya tergantung pada derajat keanggotaan yang dimilikinya. Derajat keanggotaan dalam *fuzzy* memiliki rentang nilai 0 (nol) hingga 1 (satu). Hal ini berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak). *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Logika *fuzzy* merupakan logika dalam pengambilan keputusan yang digunakan untuk memecahkan masalah dengan sistem yang sulit untuk dimodelkan, teori tentang *fuzzy set*. Dalam kehidupan banyak masalah dengan informasi yang sulit direpresentasikan kedalam sebuah model rumus atau angka yang pasti karena informasi tersebut bersifat kualitatif (tidak bisa dihitung secara kuantitatif).

2.2.2 Himpunan *Fuzzy*

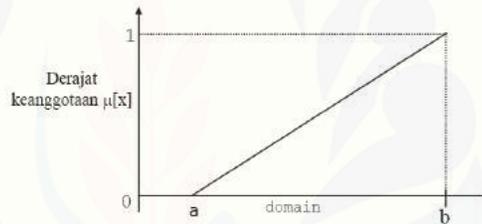
Himpunan *fuzzy* adalah himpunan-himpunan yang akan dibicarakan pada suatu variabel dalam suatu sistem *fuzzy* dimana keanggotaan dari tiap elemennya tidak mempunyai batas yang jelas. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item dalam suatu himpunan dapat memiliki dua kemungkinan, yaitu satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau nol (0), yang berarti suatu item tidak menjadi anggota dalam suatu himpunan. Sedangkan pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1, yang berarti himpunan *fuzzy* dapat mewakili interpretasi tiap nilai berdasarkan pendapat atau keputusan dan probabilitasnya.

2.2.3 Fungsi Keanggotaan (*Membership Function*)

Fungsi Keanggotaan (*Membership Function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan (Pudjo Prabowo, 2012) :

1. Representasi Linear

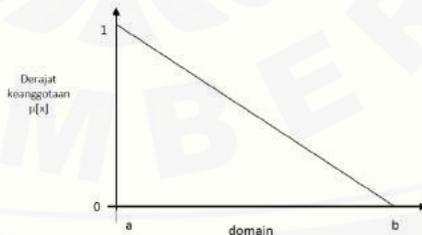
Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Dimana bentuk ini adalah bentuk yang sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.



Gambar 2.2 Representasi linear naik

Rumus fungsi keanggotaan representasi linear naik:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots 2.1$$



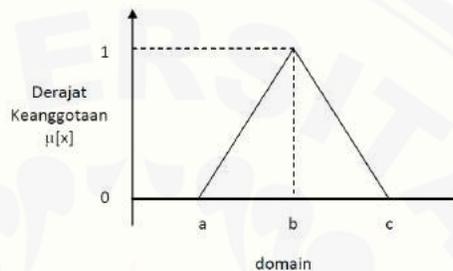
Gambar 2.3 Representasi linear turun

Rumus fungsi keanggotaan representasi linear turun:

$$\mu[x] = \begin{cases} 0; & x \geq b \\ \frac{b-x}{b-a}; & a < x < b \\ 1; & x \leq a \end{cases} \dots\dots\dots 2.2$$

2. Representasi Kurva Segitiga

Kura segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) seperti pada gambar 2.4.



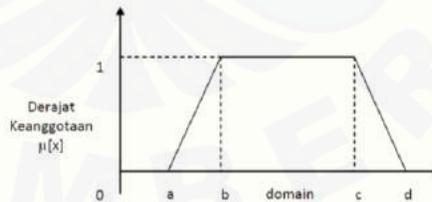
Gambar 2.4 Kurva segitiga

Rumus fungsi keanggotaan kurva segitiga:

$$\mu(x, a, b, c) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases} \dots\dots\dots 2.3$$

3. Representasi Kurva Trapezium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja adatitik yang memiliki nilai keanggotaan 1.



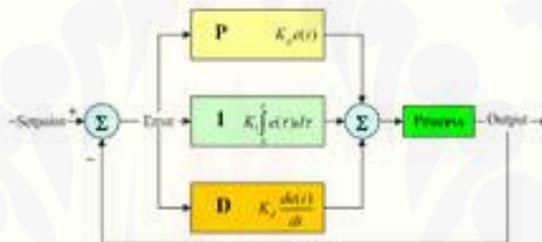
Gambar 2.5 Kurva trapesium

Rumus fungsi keanggotaan kurva trapesium:

$$\mu[x] = \begin{cases} 0; & x \geq d \text{ atau } x \leq a \\ \frac{x-a}{b-a}; & a < x < b \\ \frac{d-x}{d-c}; & c < x < d \\ 1; & b \leq x \leq c \end{cases} \dots\dots\dots 2.4$$

2.3 Proportional Integral Derivative (PID)

PID dari singkatan bahasa *Proportional Integral Derivative controller* merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. PID selalu didasari atas tinjauan terhadap karakteristik yang diatur oleh (*plant*) pengenalan *plant* tersebut harus diketahui terlebih dahulu sebelum penalaran. Komponen kontrol PID ini terdiri dari tiga jenis yaitu *Proportional*, *Integratif* dan *Derivative*. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*.



Gambar 2.6 Blok diagram kontroler PID

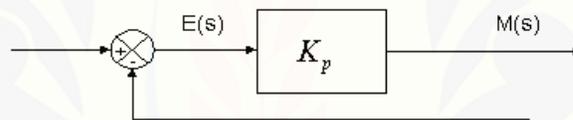
(Sumber : <http://id.wikipedia.org/wiki/PID>)

Salah satu metode kontroler PID yang banyak digunakan yakni metode *Ziegler Nicholas*. Keberadaan suatu kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem pada prinsipnya hal ini disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut yang artinya *plant* harus diterima sebagaimana adanya sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan sub sistem, salah satu tugas dari kontrol adalah mereduksi sinyal kesalahan yaitu perbedaan antara sinyal *setting* dan sinyal aktual. Hal ini sesuai dengan tujuan sistem kontrol adalah mendapatkan sinyal aktual (yang diinginkan) sama dengan sinyal *setting* dimana semakin cepat reaksi sistem mengikuti sinyal aktual sehingga semakin kecil kesalahan yang akan terjadi dan semakin baiklah sistem kinerja *control* yang diterapkan.

2.3.1 Kontroller *Proportional* (P)

Kontroller *Proportional* memberi pengaruh langsung sebanding pada *error*. Semakin besar *error*, semakin besar sinyal kendali yang dihasilkan kontroller. Pengaruh pada sistem dapat menambah atau mengurangi kestabilan, dapat memperbaiki respon transien khususnya *rise time* dan *settling time*, dan mengurangi bukan menghilangkan *Error steady state*.

Control proportional memiliki keluaran yang sebanding proporsional dengan besarnya kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya), secara sederhana bahwa keluaran kontroller *proportional* merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung akan mengubah keluaran sebesar konstanta pengalinya.



Gambar 2.7 Diagram blok kontroller *Proportional*

(Sumber : elektro indonesia, 2012)

Pada gambar 2.7 menunjukkan diagram blok yang menggambarkan hubungan antara besaran *setting*, besaran aktual dengan besaran kontroller proporsionalnya. Yang dimana keluaran dari sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya, dari hasil selisih ini akan mempengaruhi kerja dari kontroller untuk mengeluarkan sinyal positif yang digunakan untuk (mempercepat harga *setting*) atau negatif juga yang digunakan untuk (memperlambat tercapainya harga yang diinginkan).

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika

$$u = K_p \cdot e \dots\dots\dots 2.5$$

dengan K_p adalah Konstanta *Proportional*. K_p berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun

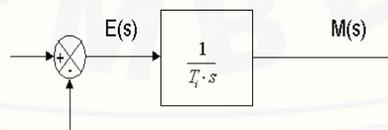
demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Secara pengguna controller *proportional* harus memperhatikan ketentuan sebagai berikut :

1. Bila nilai K_p kecil, controller *proportional* hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. Bila nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan yang diinginkan.
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosilasi.

2.3.2 Kontroller *Integral* (I)

Pengaruh kontroler *integral* pada sistem adalah menghilangkan *error steady state*, respon lebih lambat dibandingkan dengan P, dapat menambah ketidak stabilan karena menambah orde pada sistem. Perubahan sinyal kontrol sebanding dengan perubahan *error*. Semakin besar *error*, semakin cepat sinyal kontrol bertambah atau berubah.

Kontroller *integral* memiliki karakteristik seperti halnya sebuah integral. Keluarannya controller sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan, keluaran controller ini merupakan jumlah yang terus menerus dari perubahan masukannya, kalau sinyal kesalahan tidak mengalami perubahan maka keluaran akan menjaga seperti sebelum terjadinya perubahan masukan.

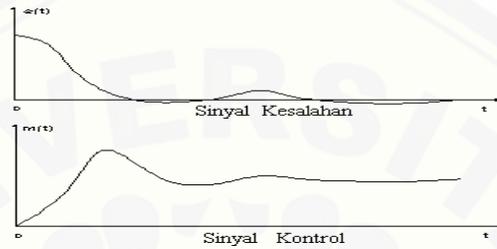


Gambar 2.8 Diagram blok kontroller *integral*

(Sumber : elektro indonesia, 2012)

Sinyal keluaran kontrol *integral* merupakan luas bidang yang dibentuk oleh kurva kesalahan penggerak lihat konsep numerik, sinyal keluaran akan berharga sama

dengan harga sebelumnya, ketika sinyal kesalahan berharga nol. Pada gambar 2.9 menunjukkan contoh sinyal kesalahan yang digambarkan ke dalam kontrol *integral* dan keluaran kontrol *integral* terhadap perubahan sinyal kesalahan tersebut sehingga timbul perbedaan grafik antara sinyal kesalahan dan sinyal kontrol terhadap kurva dari pembangkit nol.



Gambar 2.9 Kurva sinyal kesalahan terhadap sinyal kontrol

(Sumber : elektro indonesia, 2012)

Pengaruh perubahan konstanta integral terhadap keluaran integral ketika sinyal kesalahan berlipat ganda, maka nilai laju perubahan keluaran kontrol berubah menjadi dua kali dari semula. Jika nilai konstanta *integrator* berubah menjadi lebih besar, sinyal kesalahan yang relatif kecil dapat mengakibatkan laju keluaran menjadi besar.

Pengontrol *Integral* berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*Error Steady State = 0*). Jika sebuah pengontrol tidak memiliki unsur *integrator*, pengontrol *proportional* tidak mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai

$$u(t) = \int e^t dt \cdot K_i \dots\dots\dots 2.6$$

dengan K_i adalah konstanta integral, dan dari persamaan di atas, $G(s)$ dapat dinyatakan sebagai

$$u = K_d \cdot \left(\frac{\alpha e}{\alpha t} \right) \dots\dots\dots 2.7$$

Jika $e(T)$ mendekati konstan atau bukan nol maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki *error*. Jika $e(T)$ mendekati nol maka

efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde system.

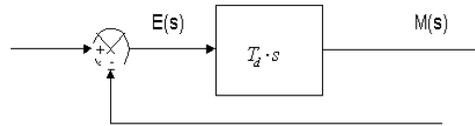
Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol *integral* merupakan luas bidang yang dibentuk oleh kurva kesalahan atau *error*. Ketika digunakan kontrol *integral* mempunyai beberapa karakteristik berikut ini :

1. Keluaran kontrol membutuhkan selang waktu tertentu, sehingga kontrol *integral* cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .

Konstanta integral K_i yang berharga besar akan mempercepat hilangnya *offset*. Tetapi akan semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran kontrol.

2.3.3 Kontroller *Derivative* (D)

Pengaruh kontroller *derivative* pada sistem adalah memberikan efek redaman pada sistem yang berosilasi sehingga bisa memperbesar pemberian nilai K_p , memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*, dan *derivative* hanya berubah saat ada perubahan *error*, sehingga saat ada *error* statis *derivative* tidak beraksi, sehingga *derivative* tidak boleh digunakan sendiri.



Gambar 2.10 Diagram blok kontroler derivative
(Sumber : elektro indonesia, 2012)

Keluaran kontrol *derivative* memiliki sifat seperti halnya suatu operasi derivatif perubahan yang mendadak pada masukan kontrol, yang akan mengakibatkan perubahan yang sangat besar dan cepat. Besarnya sinyal kontrol sebanding dengan perubahan *error* (*e*) semakin cepat *error* berubah, semakin besar aksi kontrol yang ditimbulkan.

Keluaran pengontrol *derivative* memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengontrol juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*), keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan (fungsi *ramp*), keluarannya justru merupakan fungsi *step* yang besar magnitudenya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan factor konstanta *Kd*.

Sinyal kontrol *U* yang dihasilkan oleh kontrol *D* dapat dinyatakan sebagai

$$G(s) = s \cdot K_d \dots \dots \dots 2.8$$

Dari persamaan di atas, nampak bahwa sifat dari kontrol *D* ini dalam konteks kecepatan atau *rate* dari *error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi *error* yang akan terjadi. Kontrol *derivative* hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler *derivative* tidak dapat dipakai sendiri. Karakteristik kontroler *derivative* adalah sebagai berikut :

1. kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukannya (berupa sinyal kesalahan).

2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan controller tergantung nilai T_d dan laju perubahan sinyal kesalahan.
3. Controller *derivative* mempunyai satu karakter untuk mendahului sehingga controller ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi controller *derivative* dapat mengantisipasi pembangkit kesalahan dan cenderung meningkatkan stabilitas sistem.

Berdasarkan karakteristik controller tersebut, controller *derivative* umumnya dipakai untuk mempercepat respon awal suatu sistem tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja controller *derivative* hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan, sehingga controller *derivative* tidak pernah digunakan tanpa ada controller lain sebuah sistem.

2.4 Sensor

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut Transduser. Pada saat ini, sensor tersebut telah dibuat dengan ukuran sangat kecil dengan orde nanometer. Ukuran yang sangat kecil ini sangat memudahkan pemakaian dan menghemat energi.

2.4.1 Sensor *Accelerometer*

Sensor *accelerometer* adalah sebuah transduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan dan mesin. Sensor *accelerometer* juga sering digunakan untuk mengukur sudut kemiringan. Untuk mendapatkan sudut kemiringan yang akurat, *accelerometer* sering dikombinasikan dengan satu atau lebih sensor *gyroscope* dan kombinasi data yang digunakan untuk menghitung sudut.

Accelerometer digital akan memberikan informasi berupa data hasil pembacaan menggunakan rotokol serial seperti menggunakan jalur I2C, SPI dan USART. Sementara *accelerometer* analog akan menampilkan tingkat tegangan dalam kisaran yang telah ditentukan dengan mengkonversi ke nilai digital menggunakan ADC (*analog to digital converter*) modul. Dalam mendapatkan sudut *pitch* dan *roll* pada sensor *3-axis accelerometer* menggunakan rumus sebagai berikut:

$$Pitch = \alpha = \arctan\left(\frac{ax}{\sqrt{(ay)^2+(az)^2}}\right) \times \left(\frac{180}{\pi}\right) \dots\dots\dots 2.9$$

$$Roll = \beta = \arctan\left(\frac{ay}{(az)^2}\right) \times \left(\frac{180}{\pi}\right) \dots\dots\dots 2.10$$

Dimana : ax = nilai dari *accelerometer* pada sumbu x.

ay = nilai dari *accelerometer* pada sumbu y.

az = nilai dari *accelerometer* pada sumbu z.

π = phi (3,14159).

Prinsip kerja dari tranduser ini bahwa apabila suatu konduktor digerakkan melalui suatu medan magnet, atau jika suatu medan magnet digerakkan melalui suatu konduktor, maka akan timbul suatu tegangan induksi pada konduktor tersebut. *Accelerometer* akan menangkap perubahan yang terjadi pada dimensi x, y, dan z. Sumbu y akan selalu berada menuju ke atas-bawah dari *device*, tegak lurus terhadap *hardware*, sumbu x selalu berada dari sisi kesisi searah dengan *hardware* dan sumbu z merupakan sumbu maya menembus *device* jika kita memegang *device* dan melihat ke arah *device*.

2.4.2 Sensor *Gyroscope*

Gyroscope dalam dunia instrumentasi digunakan untuk mengukur orientasi berdasarkan prinsip momentum sudut. *Gyroscope* ini juga digunakan untuk mengukur kecepatan sudut dari suatu rotasi yang satuannya adalah *radians* per detik (rad/s). Momentum sudut merupakan momentum yang dimiliki benda-benda yang melakukan gerak rotasi. Momentum sudut sebuah pertikel yang berputar terhadap sumbu putar

didefinisikan sebagai hasil kali momentum linier partikel tersebut terhadap jarak partikel ke sumbu putarnya.

Besar momentum sudut dinyatakan dengan persamaan :

$$L = m \cdot v \cdot R \dots\dots\dots 2.11$$

Dari persamaan diatas didapat $p = m \cdot v$ (momentum linear). Sehingga didapat persamaan :

$$L = p \cdot R \dots\dots\dots 2.12$$

Dimana : L = momentum sudut ($\text{Kg} \cdot \text{m}^2/\text{s}$).

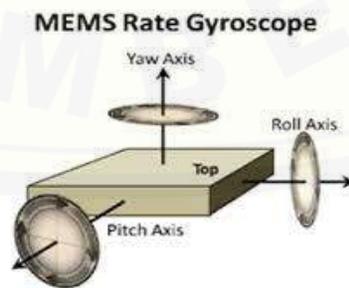
m = massa (Kg).

v = kecepatan (m/s).

R = Jari-jari lintasan (m).

p = momentum linear (kg.m/s).

Gyroscope adalah roda berat yang berputar pada jari-jarinya. Sebuah *gyroscope* mekanis terdiri dari sebuah roda yang terletak pada sebuah bingkai. Roda ini berada di sebuah batang besi yang disebut dengan poros roda. Ketika *gyroscope* digerakkan, maka ia akan bergerak mengitari poros tersebut. Poros tersebut terhubung dengan lingkaran-lingkaran yang disebut gimbal. Jadi saat piringan itu berputar, unit *gyroscope* itu akan tetap menjaga posisinya saat pertama kali dia diputar. *Gyroscope* memiliki *output* yang peka terhadap kecepatan sudut dari arah sumbu x yang nantinya akan menjadi sudut *roll*, dari sumbu y nantinya akan menjadi sudut *pitch*, dan sumbu z nantinya menjadi sudut *yaw*.



Gambar 2.11 Sikap sudut orientasi *gyroscope*

(Sumber: Adli, 2014)

2.4.3 Sensor Magnetometer

Sensor *magnetometer* merupakan sebuah modul sensor akselerasi dan medan magnet yang terintegrasi. Dimana *magnetometer* sering difungsikan untuk mengukur besarnya medan magnet bumi. Modul ini mampu menghasilkan masing-masing 3 sumbu pengukuran. Modul ini juga dapat berfungsi sebagai kompas dengan kompensasi kemiringan, deteksi posisi dan fungsi-fungsi yang diaktifkan dengan gerakan. Dengan menggunakan *3-axis magnetometer*, maka medan magnet bumi dapat diukur dengan mengetahui 3 sumbu utama bumi yaitu x, y, dan z.

2.5 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri.

Secara *software*, arduino *Open source* IDE yang digunakan untuk mendevelop aplikasi mikrokontroler yang berbasis arduino *platform*. Secara *Hardware Single board* mikrokontroler yang bersifat *open source hardware* yang dikembangkan untuk arsitektur mikrokontroler AVR 8 bit dan ARM 32 bit.



Gambar 2.12 Arduino Uno

(Sumber : <http://www.arduino.cc/en/Main/ArduinoBoardUno>)

Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para *hobbyist* atau profesional pun ikut senang mengembangkan

aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan.

Kelebihan Arduino dari *platform hardware* mikrokontrol lain adalah:

1. IDE Arduino merupakan *multiplatform*, yang dapat dijalankan di berbagai sistem operasi, seperti Windows, Macintosh dan Linux.
2. IDE Arduino dibuat berdasarkan pada IDE *processing*, yang sederhana sehingga mudah digunakan.
3. Pemrograman arduino menggunakan kabel yang terhubung dengan *port* USB, bukan *port serial*. Fitur ini berguna karena banyak komputer yang sekarang ini tidak memiliki *port serial*.
4. Arduino adalah *hardware* dan *software open source* pembaca bisa *download software* dan gambar rangkaian arduino tanpa harus membayar ke pembuat arduino.
5. Biaya *hardware* cukup murah, sehingga tidak terlalu menakutkan untuk membuat kesalahan.
6. Proyek arduino ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.
7. Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi.

2.5.1 Spesifikasi Arduino Uno

Berikut spesifikasi Arduino Uno:

- Mikrokontroler ATmega328
- Catu Daya 5V
- Tegangan *Input* (rekomendasi) 7-12V
- Tegangan *Input* (batasan) 6-20V
- Pin I/O Digital 14 (*of which 6 provide PWM output*)

- Pin *Input Analog* 6
- Arus DC per Pin I/O 40 mA
- Arus DC per Pin I/O untuk PIN 3.3V 50 mA
- *Flash Memory* 32 KB (ATmega328) dimana 0.5 KB digunakan oleh *bootloader*
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- *Clock Speed* 16 MHz

2.5.2 Bahasa pemrograman Arduino

Bahasa pemrograman yang digunakan pada arduino adalah bahasa *C* dimana bahasa *C* adalah bahasa pemrograman yang dapat dikatakan berada di antara bahasa berbasas rendah dan berbasas tinggi. Maksud dari bahasa berbasas rendah artinya bahasa yang berorientasi pada mesin dan disebut bahasa berbasas tinggi yaitu beorientasi pada manusia. Pencipta bahasa *C* adalah Brian W. Kernighan dan Denis M. Ritchi sekitar pada tahun 1972 dimana penulisan program didalam bahasa *C* dilakukan dengan membagi dalam *block – block*, sehingga bahasa *C* disebut dengan bahasa terstruktur. Bahasa *C* dapat digunakan di berbagai mesin dengan mudah mulai dari *PC* sampai *mainframe* dengan berbagai macam sistem operasi

2.6 Komunikasi Serial

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus

komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan.

Komunikasi serial ada dua macam, *asynchronous* serial dan *synchronous* serial. *Synchronous* serial adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan *clock* dan mengirimkan *clock* tersebut bersama-sama dengan data. Contoh penggunaan *synchronous* serial terdapat pada transmisi data *keyboard*. *Asynchronous* serial adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan *clock* namun hanya data yang ditransmisikan, tanpa *clock*. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi *clock* harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi *clock* pengirim dan penerima akan membaca data sesuai dengan frekuensi *clock* penerima. Contoh penggunaan *asynchronous* serial adalah pada *Universal Asynchronous Receiver Transmitter* (UART) yang digunakan pada serial *port* (COM) komputer.

2.7 Sistem Telemetry

Telemetry adalah suatu teknologi yang biasa digunakan untuk pengukuran dari jarak jauh. Hasil pengukuran yang dilakukan di lapangan akan dikirim ke tempat lain melalui proses pengiriman data. Proses pengiriman data tersebut bisa dilakukan dalam 2 (dua) cara yaitu, dengan menggunakan media perantara kabel atau tanpa menggunakan media perantara kabel yang biasa disebut dengan nirkabel (*wireless*). Hasil pengiriman data dapat dimanfaatkan secara langsung atau bisa juga diolah kembali dengan menganalisa hasil pengukurannya. Secara umum sistem telemetry memiliki beberapa bagian pendukung, yaitu objek ukur, sensor, pemancar, saluran transmisi, penerima, dan tampilan (*user interface*).

Sistem Telemetry Wireless merupakan pengukuran yang dilakukan dari jarak jauh dengan memanfaatkan media radio frekuensi sebagai proses pengiriman data.

Pada sistem telemetri terdapat dua bagian utama dalam sistem telemetri wireless. Bagian pertama adalah *measuring end* yang terdiri dari sensor untuk membaca proses pengukuran pada alat yang akan dipantau dan pemancar nirkabel untuk mengirim data pengukuran ke penerima. Sedangkan bagian kedua adalah *receiving end* yang terdiri dari penerima nirkabel untuk menerima data pengukuran yang dikirim oleh transmitter dari bagian *measuring end* dan komputer atau sejenisnya untuk memproses, menampilkan, dan menyimpan data hasil pengukuran.

2.8 Software Processing

Processing adalah bahasa pemrograman *open source* dengan pengembangan terpadu (IDE) yang digunakan untuk memprogram gambar, animasi dan interaksi. Digunakan oleh pelajar, seniman, desainer, peneliti, dan *hobbyist* untuk belajar, membuat prototipe, dan produksi. *Processing* digunakan untuk mengajarkan dasar-dasar pemrograman komputer dalam konteks rupa dan berfungsi sebagai buku sketsa perangkat lunak (*software*) dan *tool* produksi profesional.

Processing adalah suatu proyek terbuka yang diinisiasi oleh Ben Fry dan Casey Reas. Berkembang dari ide-ide yang dieksplorasi di *Aesthetics and Computation Group* (ACG) di MIT Media Lab. Proyek ini kini terus diperbaiki dan dikelola oleh sejumlah kecil tim relawan. *Processing* mengaitkan konsep *software* pada prinsip-prinsip bentuk rupa, gerak, dan interaksi. *Processing* mengintegrasikan suatu bahasa pemrograman, lingkungan pemrograman, dan metodologi pengajaran ke dalam sistem terpadu.

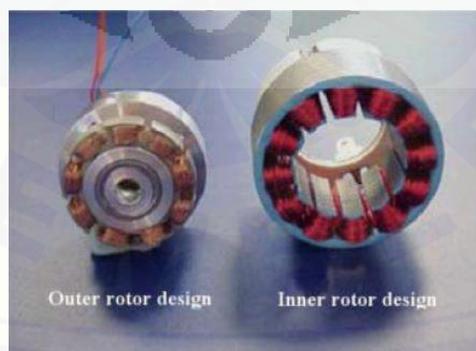
Lingkungan pemrograman *Processing* terdiri dari teks editor terintegrasi dan jendela tampilan untuk menampilkan program. Jika tombol “*run*” ditekan, program akan mengkompilasi dan berjalan di jendela grafis (jendela tampilan). Dari jendela lingkungan utama, bisa menjalankan, menghentikan, menyimpan, membuka, dan mengekspor file.

Processing menggunakan sistem koordinat kartesian dengan titik asal terletak di sudut kiri-atas. Bila program berukuran lebar 320 piksel dan lebar 240 piksel, maka koordinat [0, 0] terletak di kiri-atas dan koordinat [320, 240] terletak di kanan bawah.

2.9 Motor *Brushless*

Motor *brushless* adalah sebuah motor yang bekerja pada AC 3 fasa yang cukup optimal untuk mengendalikan beberapa alat-alat aeromodeling maupun alat-alat sederhana lainnya. *Brushless* motor tidak menggunakan sikat untuk menggerakkan rotornya. Sistem 3 fasa yang diberikan pada *brushless* motor, dapat menggerakkan dan membuatnya berputar akibat magnet permanen yang ada pada motor tersebut dengan pemberian sinyal 3 fasa.

Brushless motor tidak termasuk motor AC karena tetap menggunakan sumber DC pada sumber tegangannya dan mengubah menjadi tegangan AC menggunakan inverter 3 fasa. *Brushless* motor terdiri dari dua bagian, yaitu rotor dan stator. Rotor adalah bagian yang bergerak, terbuat dari magnet permanen dan stator merupakan bagian yang diam, terdiri dari kumparan. Elektromagnet pada *brushless* motor tidak bergerak, hanya dipindahkan dari stator ke rotor. Letak rotor dapat berada di dalam stator maupun di luar stator.



Gambar 2.13 Letak rotor di dalam dan di luar stator

(Sumber : Anwar, 2012)

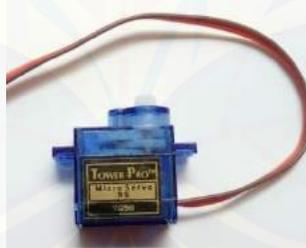
Motor *brushless* memiliki efisiensi yang tinggi dan kecepatan yang tinggi sehingga sering digunakan dalam dunia aeromodeling.

2.10 Motor Servo

Motor servo adalah sebuah motor dengan sistem *closed feedback* dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang diirim melalui kaki sinyal dari kabel motor.

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor *stepper*. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

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 in kontronnya.



Gambar 2.14 Motor servo

Motor servo merupakan sebuah motor DC yang memiliki rangkaian kontrol elektronik dan *internal gear* untuk mengendalikan pergerakan dan sudut angularnya. Sistem mekanik motor servo tampak pada gambar 2.15.



Gambar 2.15 Sistem mekanik motor servo

(Sumber : <http://elektronika-dasar.web.id/motor-servo/bagian-motor-servo/>)

Motor servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal gearnya*. Di dalam motor servo memiliki komponen yang diantaranya :

1. Memiliki 3 jalur kabel dimana kabel warna merah sebagai *power*, warna hitam sebagai *ground*, dan warna kuning sebagai kontrol signal.
2. Sinyal kontrol mengendalikan posisi.
3. Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0,5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.
4. Kontruksi didalamnya meliputi *internal gear*, potensiometer, dan *feedback control*.

2.10.1 Jenis Motor Servo

Motor servo dibagi dalam 2 jenis, yaitu:

1. Motor Servo Standar 180°

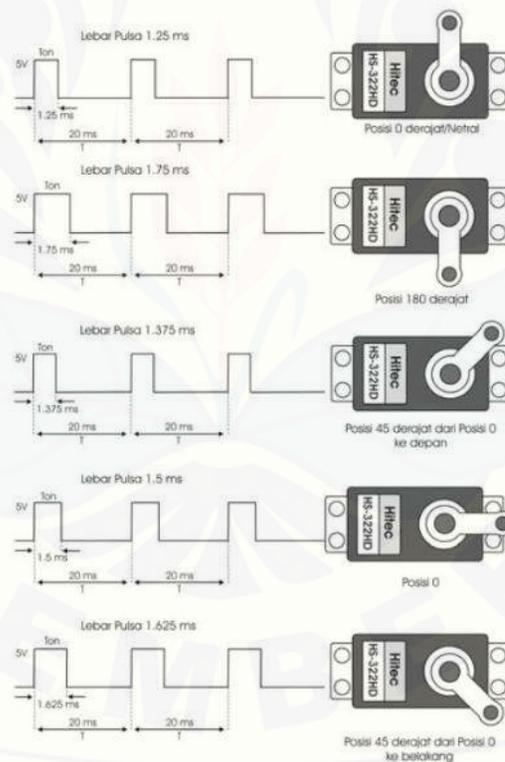
Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

2. Motor Servo *Continuous*

Motor servo ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

2.10.2 Pensinyalan Motor Servo

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya. Mode pensinyalan motor servo tampak pada gambar 2.16.



Gambar 2.16 Pengaruh lebar pulsa terhadap posisi

(Sumber : <http://elektronika-dasar.web.id/motor-servo/bagian-motor-servo/>)

Contoh dimana bila diberikan pulsa dengan lebar 1,5 ms mencapai gerakan 90°, maka bila diberikan data kurang dari 1,5 ms maka posisi mendekati 0° dan bila diberi

data lebih dari 1,5 ms maka posisi mendekati 180°. Beberapa hal yang harus diperhatikan dalam pensinyalan motor servo adalah :

1. Motor servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz.
2. Pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *Ton duty cycle* 1,5 ms, maka rotor dari motor servo akan berhenti tepat di tengah-tengah (sudut 0° atau *netral*).
3. Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1 ms, maka rotor akan berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle*, dan akan bertahan diposisi tersebut.
4. Dan sebaliknya, jika *Ton duty cycle* dari sinyal yang diberikan lebih dari 1,5 ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan di posisi tersebut.

2.11 Modul Komunikasi *Wireless* XBee PRO S2B

Radio Frequency Transceiver atau pengirim dan penerima frekuensi radio ini berfungsi untuk komunikasi secara *full duplex*. Salah satu modul komunikasi *wireless* dengan frekuensi 2.4 GHz adalah Xbee-PRO OEM ZigBee / IEEE 802.15.42.4 GHz. Radio frekuensi ini merupakan sebuah modul yang terdiri dari RF *receiver* dan RF *transmitter* dengan sistem *interface* serial UART *asynchronous*.



Gambar 2.17 Modul *Wireless*XBee PRO S2B

(Sumber: Xbee®/Xbee-PRO® ZB RF Modules, *Manual Book*)

Untuk *input* atau *output interface* XBee PRO S2B memiliki 3,3 V CMOS serial UART. Komunikasi serial adalah komunikasi yang pengiriman datanya per bit secara berurutan dan bergantian. Teknologi CMOS digunakan di mikroprosesor, pengontrol mikro, RAM statis, dan sirkuit logikan digital lainnya. Teknologi CMOS juga digunakan dalam banyak sirkuit analog, seperti sensor gambar, pengubah data, dan trimancar terintegrasi untuk berbagai jenis komunikasi UART atau *Universal Asynchronous Receiver Transmitter* adalah protokol komunikasi yang umum digunakan dalam pengiriman data serial antara device satu dengan yang lainnya. Sebagai contoh komunikasi antara sesama mikrokontroler ke PC.

BAB 3. METODOLOGI PENELITIAN

Pada bab ini menjelaskan perancangan sistem kontrol posisi dan keseimbangan roket dalam menjaga sudut luncur menggunakan kontrol PID dan *Fuzzy Logic*. Perancangan sistem ini meliputi perancangan mekanik (*prototype*) roket dengan daya dorong menggunakan motor EDF, perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*) berupa program.

3.1 Waktu dan Tempat Penelitian

Tahap penelitian ini dimulai pada bulan Mei 2015 sampai bulan Mei 2016. Sedangkan untuk tempat pelaksanaannya akan dilakukan di dua tempat yaitu, tempat tinggal pelaksana dan Universitas Jember, yaitu meliputi Laboratorium Elektronika Terapan Jurusan Teknik Elektro Universitas Jember, lingkungan stadion Universitas Jember dan ditempat tinggal pelaksana untuk mempersingkat waktu kegiatan dan penelitian.

3.2 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

3.2.1 *Hardware*

Berikut rincian *hardware* yang digunakan dalam penelitian roket.

1. Arduino Pro Mini
2. Sensor CMPS11
3. Modul *Telemetry* Xbee S2B 2.5Ghz dan Xbee *Shield Breakout*
4. Xbee USB *adapter*
5. *Power supply* modul 3.3 Volt
6. Motor servo

7. ESC 50A
8. *Battery* Lippo 4 cell 14.8 Volt
9. UBEC 5 Volt
10. *Remot Controle*
11. Motor EDF 70 mm
12. Laptop

3.2.2 Software

Berikut rincian *software* yang digunakan dalam penelitian roket.

1. Arduino IDE
2. Processing 3.0.1
3. Matlab R2009b
4. *Eagle Layout Editor*
5. X-CTU
6. Microsoft Office 2007

3.3 Tahapan Penelitian

Dalam penelitian sistem kontrol posisi dan keseimbangan roket dalam menjaga sudut luncur menggunakan kontrol PID dan *fuzzy logic* dibutuhkan langkah-langkah sebagai berikut :

1. Studi literatur.

Tahap ini merupakan awal dari proses keseluruhan penelitian, diharapkan dengan menemukan literatur dari penelitian sebelumnya dapat memberikan keyakinan bahwa penelitian ini dapat dilaksanakan dan memberikan arahan untuk mengurangi kesalahan dalam penelitian. Studi literatur pada penelitian ini meliputi pemahaman mengenai mekatronika roket dan konsep keseimbangan roket dengan daya dorong rendah, sistem yang digunakan yakni kontrol PID dan *Fuzzy Logic*, serta pemahaman komponen yang digunakan.

2. Pembelian alat dan bahan.

Tahap kedua adalah pembelian material yang dibutuhkan untuk membuat mekanik (*prototype*) roket dan komponen-komponen elektronika aktif dan pasif yang dibutuhkan.

3. Pengerjaan perangkat keras.

Tahap ketiga adalah proses pembuatan sistem kontrol posisi dan keseimbangan roket yang meliputi pembuatan perangkat mekanik dan elektronik dari bahan pembuatan yang telah dibeli.

4. Penghubungan perangkat keras.

Tahap keempat adalah menghubungkan perangkat keras yang telah dirancang sebelumnya sehingga menjadi perangkat keras yang mampu bekerja secara menyeluruh dalam sebuah sistem.

5. Pembuatan perangkat lunak.

Tahap kelima adalah pembuatan perangkat lunak dengan *software* Processing sebagai media *monitoring* yang menghasilkan tiga buah data yaitu *pitch*, *roll*, dan *yaw*. Dimana tiga data tersebut diperoleh dari pembacaan sensor CMPS11.

6. Pengoperasian perangkat lunak.

Tahap keenam adalah pengoperasian perangkat lunak yang telah dirancang sebelumnya, sehingga perangkat lunak dapat digunakan dengan baik.

7. Sinkronisasi perangkat keras dengan perangkat lunak.

Tahap ketujuh adalah menghubungkan antara perangkat keras dan perangkat lunak yang telah dirancang sebelumnya, yaitu mengkoordinasikan antara keduanya agar mampu bekerja dengan baik.

8. Pengujian alat

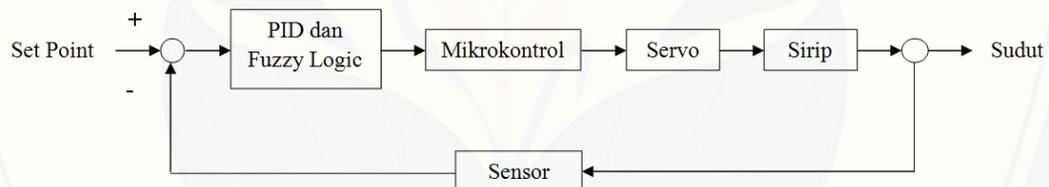
Tahap kedelapan adalah pengujian alat, diharapkan dengan adanya tahap ini perangkat keras dan perangkat lunak mampu berfungsi dengan baik dengan menguji sistem yang terintegrasi secara menyeluruh, sehingga nantinya alat yang telah dibuat dapat bekerja dengan maksimal.

9. Analisa sistem

Tahap terakhir adalah menganalisa kinerja alat, baik yang berupa perangkat keras maupun perangkat lunak sistem yang diuji. Sehingga diharapkan dengan adanya tahap ini perangkat keras dan perangkat lunak mampu bekerja dengan baik setelah diuji. Sehingga diharapkan dari analisa keseluruhan sistem perangkat dapat ditemukan suatu pembahasan atas analisa tersebut dan dapat mengambil suatu kesimpulan.

3.4 Perancangan Sistem

Dalam perancangan suatu sistem, dibutuhkan suatu blok diagram yang dapat menerangkan sistem secara keseluruhan agar sistem yang dibuat dapat berjalan dengan baik. Berikut adalah blok diagram dalam perancangan kontrol aktuator sirip roket kendali dalam mengatur sudut kemiringan dalam mengatur posisi dan keseimbangan.



Gambar 3.1 Perancangan sistem roket kendali

Pada gambar 3.1 dapat dijelaskan bahwa sistem yang terdapat pada roket tersusun dari beberapa subsistem yaitu : subsistem akuisisi oleh sensor yakni sensor *accelerometer*, sensor *gyroscope*, dan sensor *magnetometer*, subsistem pengatur arah putaran motor servo dalam menentukan sikap *roll*, *pitch*, dan *yaw*, subsistem kontrol PID dan *Fuzzy Logic*.

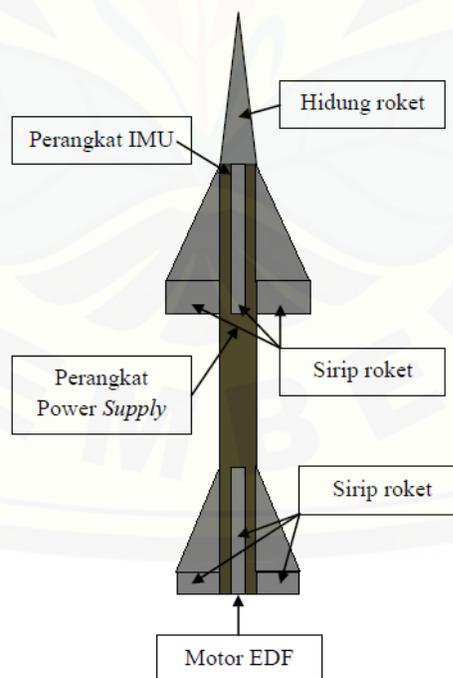
Dalam pengaplikasiannya, roket meluncur dengan menjaga sikap posisi dan keseimbangannya dengan sudut luncur yang telah ditentukan. Mikrokontrol akan memberi perintah pada sirip roket ketika posisi roket tidak seimbang, dalam

pergerakan sirip akan dipengaruhi oleh pembacaan sensor dengan kontrol PID dan *Fuzzy Logic* dalam mengembalikan roket dalam posisi seimbang.

3.5 Perancangan *Prototype* Roket

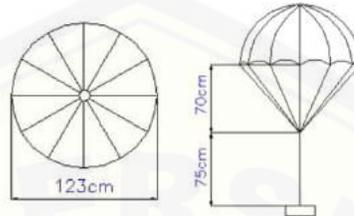
Sistem mekanik merupakan bagian yang sangat penting pada sirip roket kendali, untuk itu sistem mekanik harus sangat diperhatikan mulai dari desain *prototype* roket kendali, peletakan aktuator dalam menggerakkan sirip roket kendali dan peletakan komponen-komponen elektronika didalam *prototype* agar didapat berat seimbang.

Pada perancangan *prototype* sirip roket menggunakan 4 buah servo yang digunakan untuk menggerakkan sirip bagian atas roket dan bagian bawah roket. Sirip ini bergerak sesuai dengan kontrol yang sudah dibuat berdasarkan pembacaan sensor. Roket yang dibuat memiliki panjang 75 cm, lebar 25cm, dan diameter badan roket 7 cm. Berikut gambar 3.2 adalah perancangan roket kendali kecepatan rendah.



Gambar 3.2 Perancangan roket kendali

Pada *prototype* roket terdapat sistem parasut yang terletak diujung atas roket. Parasut ini berguna agar *prototype* tidak rusak pada saat jatuh setelah diluncurkan dalam waktu 10 detik. Untuk lebih jelas dapat dilihat pada gambar 3.3 berikut.



Gambar 3.3 Perancangan parasut

Perancangan *prototype* roket kendali terdapat beberapa bagian utama. Pada bagian atas terdapat hidung (*nose*) yang didalamnya terdapat parasut. Parasut ini akan keluar ketika dibuka oleh sebuah servo yang mengunci hidung roket. Gambaran mulut roket sebagai berikut pada gambar 3.4.



Gambar 3.4 Hidung (*nose*) beserta parasut

Pada bagian tengah *prototype* roket terdapat dua sirip utama dengan lebar sirip 25 cm. Dua sirip ini berguna untuk mengatur sudut *pitch* yang menggerakkan posisi roket ke arah depan dan belakang. Sirip ini digerakkan dengan aktuator servo 2 buah dengan pergerakan yang sama antara kiri dan kanan. Didalam *prototype* dibagian tengah terdapat beberapa komponen elektronika yang meliputi sensor, mikrokontrol, battery, Xbee Pro S2b, ESC dan Ubec. Berikut *prototype* roket bagian tengah.



Gambar 3.5 Bagian tengah roket

Pergerakan motor servo langsung digunakan sebagai keluaran untuk menggerakkan sirip roket, untuk penghubung dari servo dengan sirip roket digunakan penghubung berupa *clevies* sirip.



Gambar 3.6 Penyambungan servo dengan sirip

Sistem pendorong *prototype* roket kendali berada pada bagian bawah atau ekor (*tail*). Sistem pendorong *prototype* roket menggunakan aktuator berupa motor EDF dengan diameter 70 mm. Pada ekor *prototype* roket terdapat 2 buah sirip kecil yang bersifat pasif atau tidak bergerak dengan lebar 20 cm dan 2 buah sirip besar yang digunakan untuk mengatur sudut *roll* dan *yaw* dengan lebar 25 cm. Berikut gambar ekor roket.

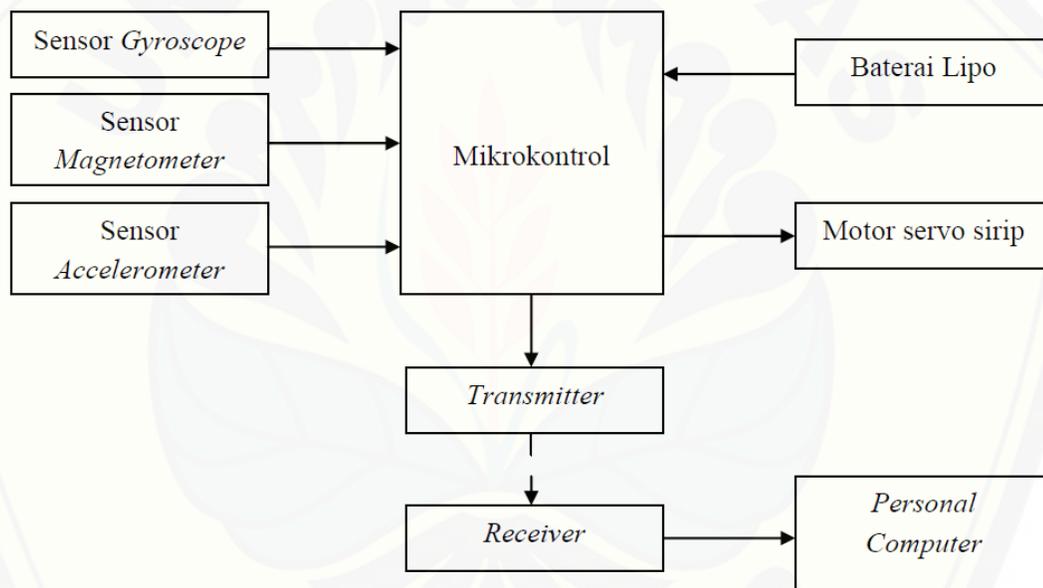


Gambar 3.7 Bagian bawah roket

Dalam menggerakkan pendorong roket menggunakan motor brushless EDF dengan daya dorong (*trust*) 1.2Kg. untuk mengontrol kecepatan putar motor EDF ini menggunakan ESC yang dikontrol dari *remote controle*.

3.6 Perancangan Elektronik

Perancangan elektronik yang digunakan disini sebagai komponen yang digunakan untuk mengendalikan sirip *prototype* roket kendali. Berikut merupakan blok diagram perancangan elektronik yang meliputi mikrokontrol, sensor, servo dan sistem *telemetry*.



Gambar 3.8 Diagram blok perancangan sistem kendali roket

Pengendali sistem kontrol posisi dan keseimbangan *prototype* roket dipengaruhi oleh pembacaan sensor-sensor yang digunakan dalam menunjukkan sudut orientasi dari posisi roket. Sensor membaca sudut orientasi dan arah yang diproses pada mikrokontrol untuk menentukan tindakan yang akan dilakukan oleh motor servo, dalam menjaga sudut orientasi yang ditentukan. Disini roket diprogram agar dapat meluncur dengan menjaga sudut luncur yang sudah ditentukan dengan percobaan 3

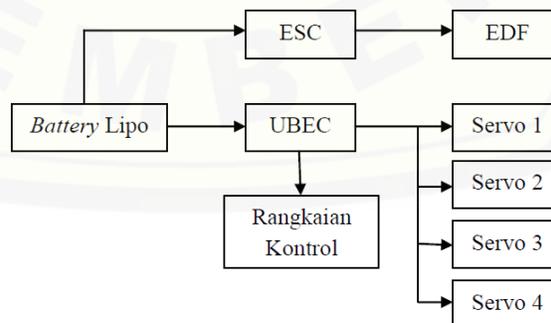
sudut orientasi yang. Sudut orientasi yang diprogram mengacu pada sudut *roll*, *pitch*, dan *yaw*.

Didalam perangkat elektronik, pemroses utama menjadi otak dalam sistem kendali dikerjakan oleh mikrokontrol Arduino Pro Mini, mikrokontrol ini bertugas untuk mengatur semua masukan, menjalankan fungsi perangkat lunak dan mengatur seluruh keluaran. Masukan pada mikrokontrol berupa pembacaan sudut yang diolah oleh sensor CMPS11 dimana keluaran sensor ini berupa sudut *pitch*, *roll* dan *yaw*. Keluaran dari sistem kendali dari mikrokontrol berupa gerak servo yang sudah terkontrol dan juga keluaran data yang dikirim melalui perangkat sistem *telemetry* berupa hubungan tanpa kabel menggunakan sepasang modul Xbee Pro S2B yang dikirim ke *Ground Segment* untuk memantau pergerakan dari *prototype* roket.

Berikut spesifikasi perancangan elektronika *prototype* roket kendali kecepatan rendah.

1. Power Supply

Untuk memberi *supply* tegangan ke seluruh perangkat elektronik, perlu sebuah battery dengan tegangan 14.8 volt 4 *cell* yang dirangkai pada gambar menggunakan modul UBEC dengan tegangan input 8 volt – 26 volt dan output yang dihasilkan sebesar 5 volt 5 amper. Berikut alur pemasangan *power supply* melalui blok diagram yang ditunjukkan pada gambar 3.9.



Gambar 3.9 Blok diagram *power supply*

2. Driver Motor EDF (ESC)

Pada bagian driver motor brushless EDF menggunakan ESC (*electronic speed control*) yang berfungsi untuk mengatur kecepatan motor EDF dengan menggunakan remot kontrol sebagai pengatur pulsa PWM pada ESC. Modul ESC mengkonversi tegangan DC dari *battery* menjadi tegangan 3 phase.



Gambar 3.10 ESC brushless motor

ESC yang digunakan memiliki arus output 50 ampere dengan menggunakan *battery* 2 cell – 4 cell lipo. Penggunaan ESC ini disesuaikan dengan kebutuhan dari motor EDF yang digunakan dengan kebutuhan tegangan 14.8 volt 4 cell dengan arus 50 ampere. Motor EDF yang digunakan memiliki diameter 70mm dengan jumlah baling-baling 6 *blade*.



Gambar 3.11 Motor brushless EDF 70mm

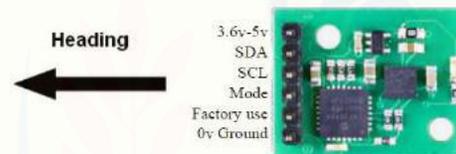
3. Rangkaian Sensor CMPS11

Penggunaan sensor diperlukan untuk pembacaan sudut kemiringan dan sudut orientasi dari roket. Sensor yang digunakan dalam sistem kontrol posisi dan keseimbangan roket menggunakan sensor CMPS11 yaitu sensor yang dilengkapi dengan sensor *accelerometer*, *gyroscope* dan *digital compass (magnetometer)* yang sudah tergabung dalam satu modul.



Gambar 3.12 Modul CMPS11

Dalam perancangan elektronik, modul sensor CMPS11 dikonfigurasi untuk mendapatkan pembacaan sensor berupa data olahan sudut yang dioperasikan melalui jalur I2C (*inter-integrated circuit*). Penggunaan jalur operasi I2C dihubungkan melalui 2 pin jalur analog input pada pin SDA (pin *serial data*) dan SCL (pin *serial clock*) dalam mengirimkan dan menerima data.



Gambar 3.13 Konfigurasi I2C sensor CMPS11

(Sumber : CMPS11 *Datasheet*)

Penggunaan mode operasi I2C, data dikirim per-bit melalui alamat register 0x60 yang merupakan data register dari CMPS11. Penggunaan *supply* tegangan pada CMPS11 menggunakan tegangan 3.6 volt – 5 volt. Jalur pin analog input yang digunakan pada Arduino menggunakan pin A4 (pin data SDA) dan A5 (pin data SCL). Penggunaan pin *serial clock* (SCL) yang pada Arduino memberikan pulsa *clock interval* pengiriman data, dan pin *serial data* (SDA) yang memuat data yang akan dikirimkan antar kedua perangkat Arduino dan CMPS11.

Dalam mengakses data keluaran dari sensor CMPS11 menggunakan data register yang sudah ditentukan dalam *data sheet*.

Tabel 3.1 *Data sheet* sensor CMPS11

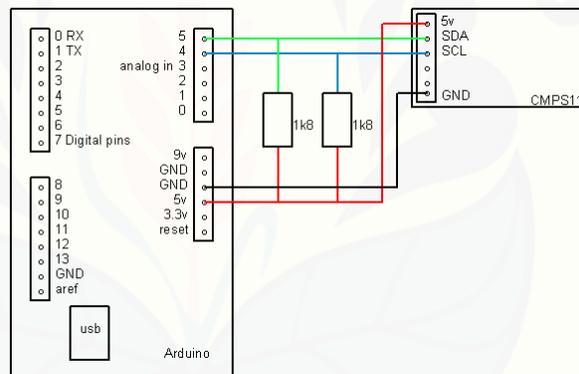
Register	Function
0	<i>Command register (write) / Software version (read)</i>
1	<i>Compass Bearing 8 bit, i.e. 0-255 for a full circle</i>
2, 3	<i>Compass Bearing 16 bit, i.e. 0-3599, representing 0-359.9 degrees. register 2 being the high byte</i>
4	<i>Pitch angle - signed byte giving angle in degrees from the horizontal plane, Kalman filtered with Gyro</i>
5	<i>Roll angle - signed byte giving angle in degrees from the horizontal plane, Kalman filtered with Gyro</i>
6, 7	<i>Magnetometer X axis raw output, 16 bit signed integer with register 6 being the upper 8 bits</i>
8, 9	<i>Magnetometer Y axis raw output, 16 bit signed integer with register 8 being the upper 8 bits</i>
10, 11	<i>Magnetometer Z axis raw output, 16 bit signed integer with register 10 being the upper 8 bits</i>
12, 13	<i>Accelerometer X axis raw output, 16 bit signed integer with register 12 being the upper 8 bits</i>
14, 15	<i>Accelerometer Y axis raw output, 16 bit signed integer with register 14 being the upper 8 bits</i>
16, 17	<i>Accelerometer Z axis raw output, 16 bit signed integer with register 16 being the upper 8 bits</i>
18, 19	<i>Gyro X axis raw output, 16 bit signed integer with register 18 being the upper 8 bits</i>
20, 21	<i>Gyro Y axis raw output, 16 bit signed integer with register 20 being the upper 8 bits</i>
22, 23	<i>Gyro Z axis raw output, 16 bit signed integer with register 22 being the upper 8 bits</i>

Tabel 3.2 *Data sheet* sensor CMPS11 (lanjutan)

Register	Function
24, 25	Temperature raw output, 16 bit signed integer with register 24 being the upper 8 bits
26	Pitch angle - signed byte giving angle in degrees from the horizontal plane (no Kalman filter)
27	Roll angle - signed byte giving angle in degrees from the horizontal plane (no Kalman filter)

Sumber : *User Guides* CMPS11

Dalam penggunaan data yang diolah dalam pembacaan sensor menggunakan nomor register yang telah tersedia pada tabel di atas.



Gambar 3.14 Rangkaian Arduino dengan CMPS11

(Sumber : http://www.robot-electronics.co.uk/htm/arduino_examples)

Pada perancangan rangkaian hubungan Arduino dengan CMPS11 pada jalur SDA dan SCL ada penambahan komponen yang dipasang secara paralel berupa komponen resistor. Resistor yang digunakan bernilai 1K8 ohm yang dihubungkan ke VCC (tegangan *input* sensor).

4. Arduino Pro Mini

Mikrokontrol yang digunakan dalam perancangan roket kendali menggunakan Arduino Pro Mini yang merupakan mikrokontrol keluarga Arduino. Pin yang digunakan dalam rancang kontrol menggunakan jalur

analog input, pin serial dan digital PWM. Pin yang digunakan dalam Arduino Pro Mini sebagai berikut.

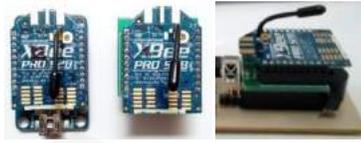
1. Analog pin *input* menggunakan pin A4 dan pin A5 sebagai jalur I2C. Jalur I2C ini digunakan untuk hubungan *input* dengan sensor.
2. Pin 3 digital PWM dihubungkan dengan sensor IR.
3. Pin serial menggunakan jalur RX dan TX yakni pin 0 dan pin 1 pada jalur digital. Pin RX Arduino Pro Mini dihubungkan dengan pin TX modul Xbee Pro S2B dan pin TX Arduino Pro Mini dihubungkan dengan pin RX modul Xbee Pro S2B.
4. Dengan 4 pin pada pin digital PWM untuk *output* pada motor servo menggunakan pin 5, pin 6, pin 10 dan pin 11.



Gambar 3.15 IMU (*Inertial Measuring Unit*) Arduino Pro Mini Board Arduino Pro Mini dibentuk dalam satu modul IMU dengan komponen lain agar mempermudah proses pemrograman juga meringkas pengkabelan.

5. Rancangan Sistem *Telemetry*

Rancangan sistem pengiriman data sistem kontrol posisi dan keseimbangan pada roket menggunakan modul Xbee Series 2B. Modul ini digunakan untuk mengirim data dari mikrokontrol Arduino Pro Mini menuju PC (*personal computer*). Modul ini bekerja di frekuensi 2.5 Ghz dengan tegangan 3.3 volt. Pengaturan modul Xbee Pro S2B pada IMU menggunakan *Coordinator AT* sedangkan pada Xbee Pro S2B pada *ground segment (PC)* menggunakan pengaturan *Router AT*



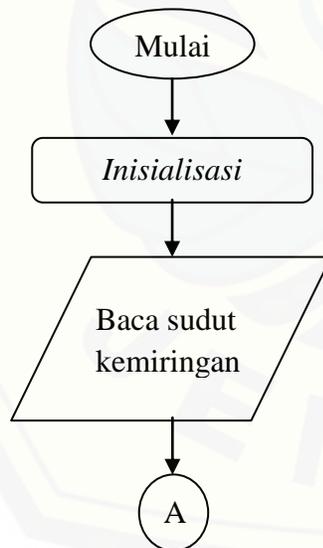
Gambar 3.16 Xbee Pro S2B dan pemasangan pada *board* IMU

3.7 Perancangan Perangkat Lunak

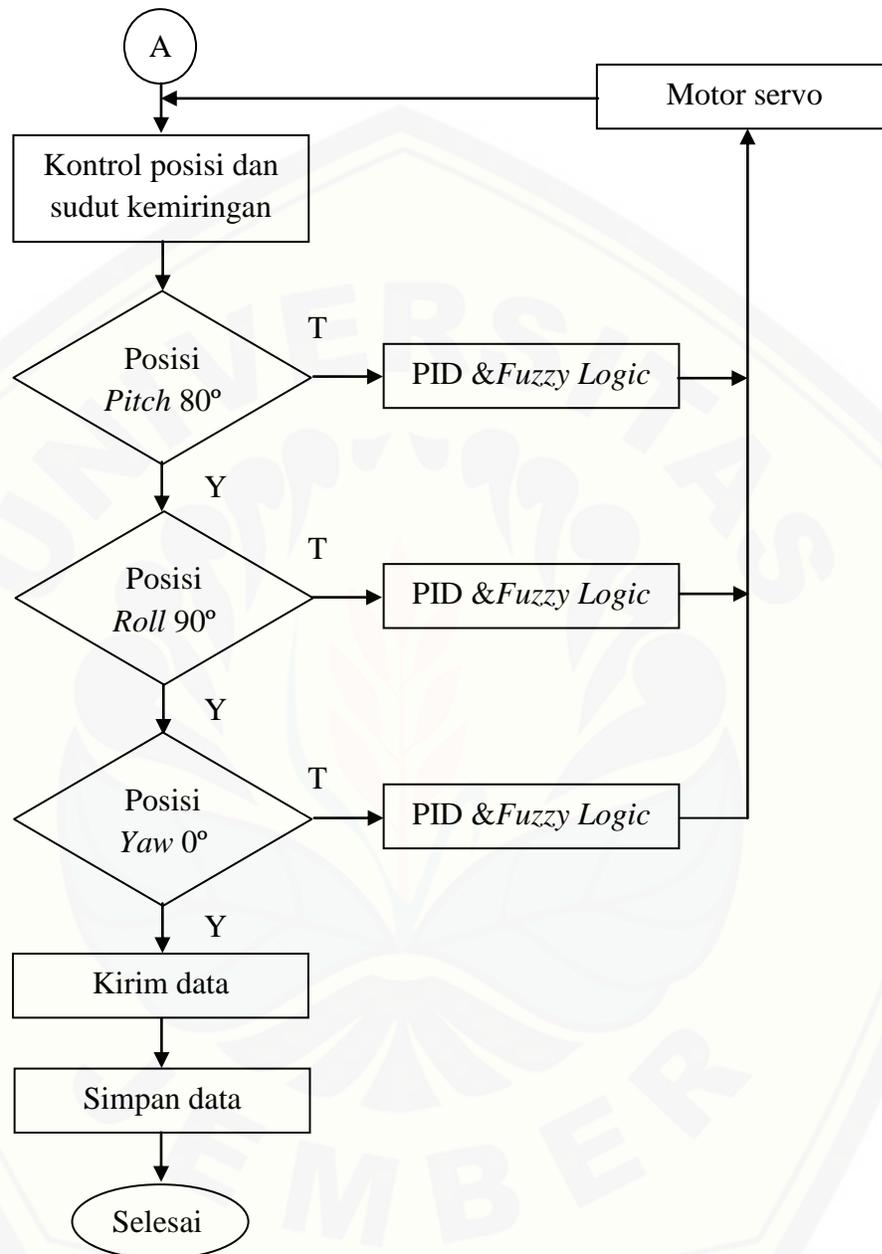
Perancangan perangkat lunak dari sistem kontrol posisi dan keseimbangan roket meliputi perancangan sistem keseluruhan berupa *flow chart* sistem, perancangan program pada Arduino Pro Mini sebagai pemroses utama dalam mengolah masukan dan keluaran, dan perancangan *software monitor attitude* roket yang berupa *ground segment*, perancangan sistem kontrol menggunakan kontrol PID dan *Fuzzy Logic*.

3.7.1 Flow Chart Perancangan Sistem

Berikut diagram alir yang digunakan dalam pemrograman arduino.



Gambar 3.17 *Flow chart* program kontrol posisi dan keseimbangan



Gambar 3.18 *Flow chart* program kontrol posisi dan keseimbangan (lanjutan)

Flow chart di atas merupakan urutan pemrograman maupun urutan kerja dari program yang dibuat dalam pemrograman arduino. Program yang dibuat adalah sistem kontrol posisi dan keseimbangan roket menggunakan PID (*proportional integral derivative*) dan *Fuzzy Logic*.

Perancangan program arduino berupa *inisialisasi input* dan *output*, proses pembacaan *input*, kontrol keluaran dan sistem pengiriman data. Berikut adalah penjelasan dari perancangan program arduino. Yang pertama dilakukan adalah *inisialisasi* masukan dan keluaran yang digunakan dalam *prototype* roket. Mulai dari masukan berupa *library* sensor CMPS11, PID dan *Fuzzy Logic*. Pembacaan sensor terhadap sensor CMPS11 adalah untuk mendapatkan sudut kemiringan, yakni sudut yang berorientasi pada sudut *pitch*, *roll*, dan *yaw*. Sehingga data yang didapatkan mikrokontrol sudah diolah menjadi data sudut.

Sistem kerja roket diawali dengan mengontrol sudut kemiringan terhadap posisi dan keseimbangan roket yang dituju dengan memberi perintah yang telah diprogram dalam arduino sehingga secara otomatis roket akan bekerja sesuai dengan perintah yang sudah diperintahkan.

Dalam mengatur kontrol posisi dan keseimbangan roket terdapat tiga sudut kemiringan yang menjadi acuan sudut kemiringan, yaitu sudut *pitch* dengan besar sudut kemiringan 80° terhadap sudut tegak roket, sudut *roll* dengan besar sudut kemiringan 90° terhadap sudut lurus roket, dan sudut *yaw* dengan besar sudut 0° dimana sudut ini merupakan sudut yang berorientasi terhadap arah mata angin.

Konsep kontrol posisi dan keseimbangan roket menggunakan sayap yang akan mempengaruhi sudut kemiringan gerak roket dalam meluncur di udara. Sayap roket akan digerakkan oleh motor servo yang diperintahkan oleh mikrokontrol. Pengendali sistem keseimbangan roket dipengaruhi oleh pembacaan sudut kemiringan yang digunakan dalam menunjukkan sudut orientasi dari posisi roket. Sensor membaca sudut orientasi dan arah yang diproses pada mikrokontrol untuk menentukan tindakan yang akan dilakukan oleh motor servo dalam menjaga sudut orientasi yang ditentukan.

Sistem kontrol PID dirancang untuk membenahi kesalahan sudut kemiringan antara pembacaan sensor dengan sudut yang dihasilkan dari gerak motor servo. *Fuzzy logic* dirancang untuk membaca kondisi dari setiap pergerakan roket, sehingga dari pembacaan ini akan diproses sesuai dengan aturan-aturan yang dibuat untuk

menentukan lebar pulsa yang akan diberikan pada motor servo. Sehingga diharapkan roket dapat bekerja dengan baik saat meluncur di udara.

Data yang diperoleh setiap pergerakan roket yang telah diolah oleh mikrokontrol akan dikirim ke *ground segment* melalui sistem *telemetry* berupa komunikasi *serial*. Komunikasi *serial* pada roket menggunakan *software* Processing untuk pembuatan *ground segment* dalam menampilkan informasi berupa data-data pembacaan sensor dan posisi roket saat meluncur berupa sudut kemiringan. Dari *ground segment* ini dapat diketahui seberapa baik sistem kontrol posisi dan keseimbangan roket yang dibuat untuk menjaga sudut luncurnya.

Berikut merupakan alur penulisan program kontrol posisi dan keseimbangan roket pada Arduino.

1. *Inisialisasi* masukan dan keluaran pada program arduino

Pada bagian ini merupakan bagian dalam *inisialisasi* masukan-masukan dan keluaran pada pemrograman sistem kontrol posisi dan keseimbangan roket. Hal ini penting karena pada proses pemrograman pada Arduino dibutuhkan *inisialisasi* komponen masukan dan keluaran yang digunakan sehingga proses yang ditulis pada program dapat difungsikan dengan maksimal sesuai dengan kegunaan masing-masing.

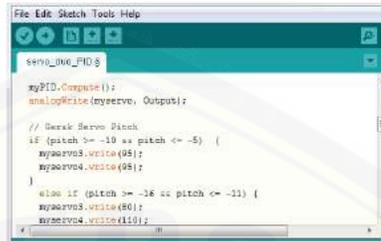


```
1 // Include the libraries we need
2 #include <PID.h>
3 #include <Servo.h>
4 #include <XBee.h>
5 #include <XBeeSerial.h>
6 #include <XBeeModule.h>
7 #include <XBeeModule.h>
8 #include <XBeeModule.h>
9 #include <XBeeModule.h>
10 #include <XBeeModule.h>
11 #include <XBeeModule.h>
12 #include <XBeeModule.h>
13 #include <XBeeModule.h>
14 #include <XBeeModule.h>
15 #include <XBeeModule.h>
16 #include <XBeeModule.h>
17 #include <XBeeModule.h>
18 #include <XBeeModule.h>
19 #include <XBeeModule.h>
20 #include <XBeeModule.h>
21 #include <XBeeModule.h>
22 #include <XBeeModule.h>
23 #include <XBeeModule.h>
24 #include <XBeeModule.h>
25 #include <XBeeModule.h>
26 #include <XBeeModule.h>
27 #include <XBeeModule.h>
28 #include <XBeeModule.h>
29 #include <XBeeModule.h>
30 #include <XBeeModule.h>
31 #include <XBeeModule.h>
32 #include <XBeeModule.h>
33 #include <XBeeModule.h>
34 #include <XBeeModule.h>
35 #include <XBeeModule.h>
36 #include <XBeeModule.h>
37 #include <XBeeModule.h>
38 #include <XBeeModule.h>
39 #include <XBeeModule.h>
40 #include <XBeeModule.h>
41 #include <XBeeModule.h>
42 #include <XBeeModule.h>
43 #include <XBeeModule.h>
44 #include <XBeeModule.h>
45 #include <XBeeModule.h>
46 #include <XBeeModule.h>
47 #include <XBeeModule.h>
48 #include <XBeeModule.h>
49 #include <XBeeModule.h>
50 #include <XBeeModule.h>
51 #include <XBeeModule.h>
52 #include <XBeeModule.h>
53 #include <XBeeModule.h>
54 #include <XBeeModule.h>
55 #include <XBeeModule.h>
56 #include <XBeeModule.h>
57 #include <XBeeModule.h>
58 #include <XBeeModule.h>
59 #include <XBeeModule.h>
60 #include <XBeeModule.h>
61 #include <XBeeModule.h>
62 #include <XBeeModule.h>
63 #include <XBeeModule.h>
64 #include <XBeeModule.h>
65 #include <XBeeModule.h>
66 #include <XBeeModule.h>
67 #include <XBeeModule.h>
68 #include <XBeeModule.h>
69 #include <XBeeModule.h>
70 #include <XBeeModule.h>
71 #include <XBeeModule.h>
72 #include <XBeeModule.h>
73 #include <XBeeModule.h>
74 #include <XBeeModule.h>
75 #include <XBeeModule.h>
76 #include <XBeeModule.h>
77 #include <XBeeModule.h>
78 #include <XBeeModule.h>
79 #include <XBeeModule.h>
80 #include <XBeeModule.h>
81 #include <XBeeModule.h>
82 #include <XBeeModule.h>
83 #include <XBeeModule.h>
84 #include <XBeeModule.h>
85 #include <XBeeModule.h>
86 #include <XBeeModule.h>
87 #include <XBeeModule.h>
88 #include <XBeeModule.h>
89 #include <XBeeModule.h>
90 #include <XBeeModule.h>
91 #include <XBeeModule.h>
92 #include <XBeeModule.h>
93 #include <XBeeModule.h>
94 #include <XBeeModule.h>
95 #include <XBeeModule.h>
96 #include <XBeeModule.h>
97 #include <XBeeModule.h>
98 #include <XBeeModule.h>
99 #include <XBeeModule.h>
100 #include <XBeeModule.h>
```

Gambar 3.19 *Inisialisasi* masukan dan keluaran roket

Pada bagian *inisialisasi* masukan dan keluaran terdapat beberapa komponen yang didefinisikan pada *inisialisasi*, kontrol PID dan *fuzzy logic*, motor servo, sensor CMPS11 dan Xbee Pro S2B merupakan komponen yang digunakan seperti pada gambar di atas.

Arduino Pro Mini yang digunakan. Seperti pin yang digunakan pada servo, besar nilai *baud rate* yang digunakan pada Xbee Pro S2B, dan nilai *set point*.



```
File Edit Sketch Tools Help
servo_duo_PID $
myPID.Controller();
analogWrite(myServo, Output);

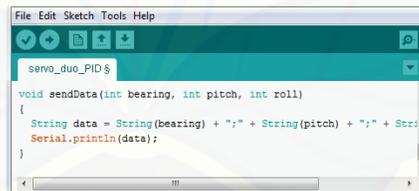
// Gerak Servo Pitch
if (pitch >= -10 && pitch <= -5) {
  myServo3.write(95);
  myServo4.write(95);
}
else if (pitch >= -16 && pitch <= -11) {
  myServo3.write(80);
  myServo4.write(110);
}
```

Gambar 3.22 Program kontrol keluaran servo

Pada gambar dapat dijelaskan bahwa sistem kontrol posisi dan keseimbangan diatur dari pergerakan motor servo yang dihubungkan dengan sayap roket sehingga dari pergerakan sayap roket dapat mengontrol sudut kemiringan roket.

4. Sistem pengiriman data

Pada sistem *telemetry* untuk mengirim data yang menggunakan modul Xbee Pro S2B, modul ini berfungsi hanya untuk mengirim data yang dihasilkan dari controller roket menuju *ground segment* pada komputer.



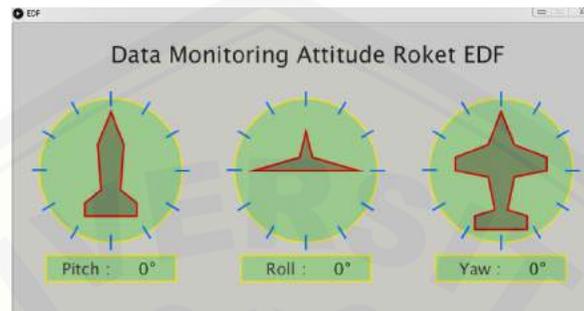
```
File Edit Sketch Tools Help
servo_duo_PID $
void sendData(int bearing, int pitch, int roll)
{
  String data = String(bearing) + ";" + String(pitch) + ";" + Str
  Serial.println(data);
}
```

Gambar 3.23 Program sistem pengiriman data

3.7.2 Perancangan *Software Monitor Attitude* Roket

Perancangan *software monitor attitude* roket berupa aplikasi yang dibuat untuk melihat (*monitoring*) sudut dan gerakan *visual* dari roket kendali. Sudut dan gerakan ini didapat dari pembacaan besar sudut *pitch*, *roll*, dan *yaw* yang dikirim melalui sistem *telemetry* dari *prototype* roket kendali yang dikirim ke *ground segment* berupa komputer. Pada perancangan aplikasi *monitoring* ini menggunakan *software* Processing 3.0.1 dengan bahasa pemrograman *Java* dan jalur komunikasi berupa

komunikasi *serial*. Didalam aplikasi yang dibuat ini hanya terdapat fungsi komunikasi serial dan menampilkan data secara *visual*. Berikut adalah gambar aplikasi *minitor attitude* roket.



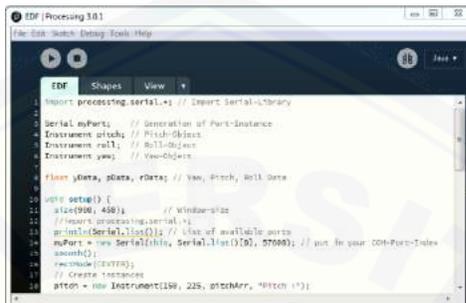
Gambar 3.24 Perancangan aplikasi data *monitoring attitude* roket EDF

Pada aplikasi ini fungsi nilai dari *pitch*, *roll*, dan *yaw* yaitu untuk mengetahui sudut kemiringan yang terjadi pada roket. Sudut kemiringan ini digunakan sebagai parameter untuk mengetahui kondisi roket dalam keadaan terkontrol atau tidak. Dari aplikasi ini dapat diketahui jika roket dapat terkontrol posisi dan keseimbangannya. Pada aplikasi ini fungsi komunikasi *serial* adalah digunakan untuk mendeteksi *serial port* piranti komunikasi *serial* dari perangkat keras yang digunakan sebagai *system telemetry* dan juga untuk mekanisme pengiriman data. Fungsi pengiriman data adalah untuk mengetahui data hasil pergerakan roket terhadap sudut kemiringan dengan kontrol posisi dan keseimbangan yang sudah diprogram, sehingga roket dapat dipantau hasilnya saat di udara.

Pada perancangan aplikasi *monitoring attitude* roket ini, hanya menerima data dari *serial* komunikasi *system telemetry* dengan data yang dikirim dari *prototype* roket yang kemudian ditampilkan lagi dalam bentuk *visual* berupa gerak roket dan nilai kemiringan roket. Pada perancangan aplikasi ini pembacaan data komunikasi *serial* arduino dan tampilan *visual* yang dihasilkan dari *software* processing tidak dapat digunakan secara bersamaan. Hal ini dikarenakan sistem komunikasi *serial* pada *system telemetry* hanya bisa diakses pada satu jalur saja.

Seperti pada umumnya pemrograman *visual* pada aplikasi ini juga terdapat beberapa langkah dalam menyusun program, seperti dimulai dengan *inisialisasi*

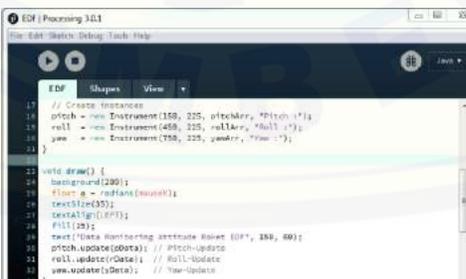
sistem perangkat dan *serial*, komponen yang digunakan untuk menampilkan data dan juga pengaturan bentuk meliputi desain gambar *visual*, tulisan, dan nilai yang akan dimunculkan.



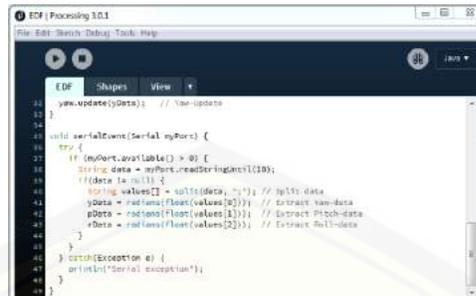
Gambar 3.25 Inisialisasi pada *software* Processing

Pada gambar merupakan proses awal pemrograman aplikasi data *monitoring attitude* roket EDF dalam hal *inisialisasi* yang meliputi data *serial* komunikasi dan tipe data yang diterima dari arduino pro mini. Data yang diterima dari pengiriman data berupa bilangan desimal yang mengacu pada nilai sudut kemiringan *pitch*, *roll* dan *yaw*.

Selanjutnya adalah pembuatan tampilan *visual* dengan menggunakan *instrument* yang akan digunakan sebagai *icon* pada tampilan *visual* yang pergerakan dari *instrument* ini disesuaikan dengan data yang diterima agar setiap pergerakan dan nilai yang ditampilkan sesuai dengan keadaan roket pada saat di udara. Dengan demikian hal tersebut membutuhkan data secara *real time* agar data yang diterima dalam setiap waktu selalu sama dengan keadaan roket.



Gambar 3.26 Pengaturan *instrument visual*



```

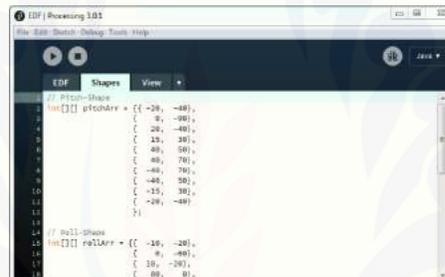
EDF [Processing 3.0.1]
File Edit Sketch Debug Tools Help

EDF Shapes View
// var-update
11 void update(Data) { // var-update
12
13
14 void serialEvent(Serial myPort) {
15   try {
16     if (myPort.available() > 0) {
17       String data = myPort.readStringUntil(10);
18       if (data != null) {
19         // Split data
20         String values[] = split(data, ","); // split data
21         xData = radians(float(values[0])); // Extract var-data
22         yData = radians(float(values[1])); // Extract Pitch-data
23         zData = radians(float(values[2])); // Extract Roll-data
24       }
25     } catch (Exception e) {
26       println("Serial exception");
27     }
28   }
29 }

```

Gambar 3.27 Komunikasi serial

Selanjutnya adalah proses dalam membentuk *instrument* yang digunakan kedalam bentuk *visual* yang dapat menggambarkan setiap bentuk dari roket. Berikut merupakan potongan program untuk membentuk desain *visual instrument*.



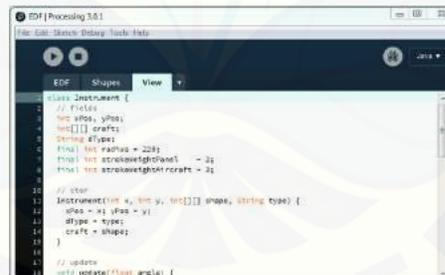
```

EDF [Processing 3.0.1]
File Edit Sketch Debug Tools Help

EDF Shapes View
// Pitch-shape
1 // Pitch-shape
2 int[][] pitchArr = {{ -20, -40},
3                   { 0, -90},
4                   { 20, -40},
5                   { 35, 30},
6                   { 40, 50},
7                   { 40, 70},
8                   { -40, 50},
9                   { -35, 30},
10                  { -20, -40}
11 };
12
13
14 // Roll-shape
15 int[][] rollArr = {{ -10, -20},
16                  { 0, -40},
17                  { 10, -20},
18                  { 0, 0}
19 };

```

Gambar 3.28 Listing program instrument roket



```

EDF [Processing 3.0.1]
File Edit Sketch Debug Tools Help

EDF Shapes View
// class Instrument
1 class Instrument {
2   // fields
3   int xPos, yPos;
4   int[][] craft;
5   String shape;
6   float int radius = 220;
7   float int strokeWeight = 3;
8   float int strokeWeightRate = 3;
9
10  // ctor
11  Instrument(int x, int y, int[][] shape, String type) {
12    xPos = x; yPos = y;
13    shape = type;
14    craft = shape;
15  }
16
17  // update
18  void update(float angle) {

```

Gambar 3.29 Pengaturan instrument visual roket

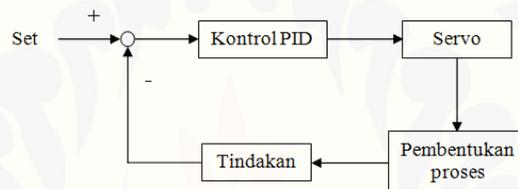
Pada program *view* terdapat beberapa pengaturan untuk mengubah nilai desimal untuk digunakan dalam bentuk radian s yaitu data berbentuk sudut (derajat) sehingga nantinya data yang ditampilkan dalam aplikasi data monitoring dapat digambarkan berupa data *radians*.

3.7.3 Perancangan Sistem Kontrol PID dan *Fuzzy Logic*

Pada perancangan sistem kontrol menggunakan dua sistem yaitu kontrol PID (*Proporsional Integral Derivative*) dan *Fuzzy Logic*. Pada kontrol PID digunakan untuk membenahi nilai pembacaan dari sensor ke motor servo sedangkan *fuzzy logic* digunakan untuk menentukan dan mengambil keputusan dari sistem yang dibuat.

1. Perancangan sistem kontrol PID

Perancangan sistem kontrol PID pada roket kendali menggunakan diagram blok berikut. Diagram blok berikut merupakan diagram sistem dari aktuator yang akan dikontrol.



Gambar 3.30 Blok diagram sistem kontrol PID

Dari gambar dapat dilihat bahwa pada kondisi awal motor servo akan bergerak sesuai dengan nilai *set pint* yang diberikan, tetapi selanjutnya nilai *input* dari motor servo diperoleh dari nilai pembentukan proses dan tindakan. Nilai error ($e(t)$) diperoleh dari hasil pengurangan nilai sensor sebagai nilai *feedback* dengan nilai *set point*.

Seperti yang dijelaskan sebelumnya bahwa persamaan-persamaan dari kontrol PID dapat dipecah menjadi kontrol *proporsional*, kontrol *integral* dan kontrol *derivatif*.

a. Kontrol *proporsional*

Berikut persamaan *proporsional*

$$u(k) = K_p e(k)$$

b. Kontrol *Integral*

Berikut persamaan *integral*

$$K_i = \frac{K_p}{T_i}$$

Dalam persamaan *integral* ini nilai error yang sebelumnya dijumlahkan dengan error-error yang sebelumnya hingga error yang sekarang.

c. Kontrol *derivatif*

Berikut persamaan *derivatif*

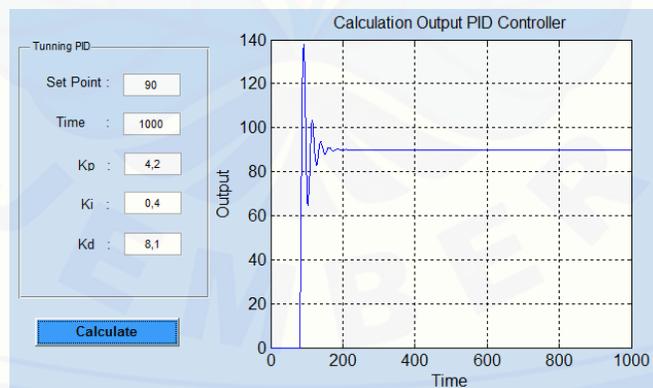
$$Kd = Kp \times Td$$

Berdasarkan persamaan di atas variabel *error* (e) yang diderivatiskan, dimana error yang sekarang dikurangi error sebelumnya. Dimana nilai-nilai persamaan variabel sebagai berikut.

$$Error = SP - PV$$

Dimana *set poin* (SP) merupakan suatu parameter nilai acuan atau nilai yang diinginkan oleh *controller*, sedangkan *Present Value* (PV) merupakan nilai pembacaan sensor pada saat itu atau dapat disebut juga dengan variabel terukur yang diumpankan balikkan oleh sensor.

Dari persamaan dari kontrol P, I dan D diatas, maka persamaan-persamaan tersebut dapat diaplikasikan ke kontrol PID. Kontrol PID merupakan gabungan dari ketiga kontrol di atas. Dimana nilai Kp, Ki, dan Kd didapat melalui tuning dengan menggunakan metode yang sudah ada dan melalui metode *try and error* (coba-coba).



Gambar 3.31 Grafik respon sistem PID

Grafik respon yang dihasilkan merupakan nilai parameter yang diambil secara *try and error* (coba-coba) dengan mendapatkan grafik diatas yang proses pengolahannya menggunakan *software* Matlab R2009b agar dapat melihat hasil

grafik respon. Fungsi dari nilai K_p , K_i dan K_d yaitu untuk mengetahui seberapa besar perubahan dari grafik *output* PID controller dari *set poin* yang telah ditentukan yang selanjutnya digunakan untuk dijadikan parameter *tuning* pada aktuator servo. Pada perancangan grafik respon controller PID hanya menerima data yang telah dihitung sebelumnya yang kemudian ditampilkan dalam bentuk *visual*.

Berikut ini terminologi tentang definisi dalam PID antara lain sebagai berikut :

1. *Error* : merupakan perbedaan target *position* dan *measured position*.
2. *Propotional* : untuk mengukur seberapa jauh roket dalam kondisi tidak seimbang dimana *propotional* merupakan dasar untuk membaca posisi roket dengan menggunakan sensor, semakin banyak data yang didapatkan maka semakin akurat kita mengukur posisi dari roket .
3. *Integral* : untuk mengakumulasi *error* terhadap waktu nilai *integral* ini bisa naik ketika roket tidak berada pada posisi seimbang, semakin lama roket dalam keadaan tidak seimbang nilai *integralnya* juga semakin tinggi.
4. *Derevative* : mengukur seberapa sering roket bergerak dari kanan ke kiri atau sebaliknya.
5. Faktor P- K_p : merupakan konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh dari *propotional*.
6. Faktor I- K_i : merupakan konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh *integral*.
7. Faktor D- k_d : merupakan konstanta yang digunakan untuk memperbesar dan memperkecil pengaruh dari *derevative*.

Dari terminologi diatas kita dapat mengetahui maksud dari penggunaan masing-masing controller PID mulai dari controller P, controller I dan controller D serta efek dari masing – masing penggunaan kontsnta dari K_p , K_i dan K_d .

2. Perancangan Fuzzy Logic

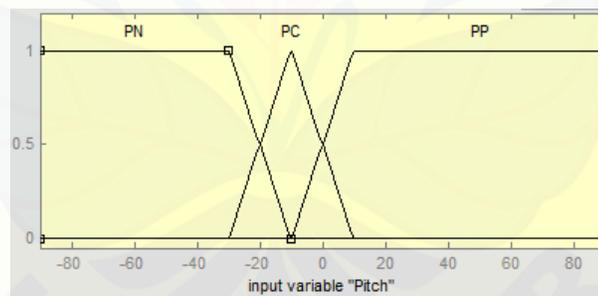
Logika *fuzzy* digunakan sebagai pengolah data masukan dari sensor CMPS11, dengan menghasilkan suatu keluaran berupa pengambil keputusan dari hasil proses

pengolahan logika *fuzzy*. Hal ini bertujuan agar proses kontrol posisi dan keseimbangan bisa berjalan lancar.

a. *Fuzzyfication*

Data hasil pembacaan sensor CMPS11 merupakan data masukan bernilai tegas (*crisp*) kemudian diubah menjadi himpunan *fuzzy* menurut fungsi keanggotannya. Proses awal dari *fuzzyfikasi* adalah membuat *membership function* (fungsi keanggotaan) dari masukan dan keluaran, serta menentukan banyaknya variable *linguistic* dalam fungsi keanggotaan tersebut. Dari fungsi keanggotaan yang dibuat akan diketahui nilai derajat keanggotaan dari masing-masing variable dalam himpunan *fuzzy* berdasarkan masukan tegas (*crisp*).

Pada perancangan sistem ini terdapat tiga masukan dari sensor CMPS11, dimana masukan dari sensor ini merupakan data sudut kemiringan pada sudut *pitch*, *roll*, dan *yaw*. Masukan dari sensor CMPS11 ini memiliki masing-masing tiga nilai linguistik untuk masing-masing masukan dari sensor. Berikut fungsi keanggotaan masukan sudut kemiringan *pitch* dengan nilai linguistik PN, PC, dan PP.



Gambar 3.32 Fungsi keanggotaan *pitch*

Pada nilai fungsi keanggotaan sudut kemiringan *pitch* nilai tengah (*center*) bernilai -10 bukan pada nilai 0, hal ini bertujuan agar roket dapat meluncur pada sudut kemiringan 80° pada orientasi sudut *pitch*. Pada input sudut kemiringan *pitch* dibagi menjadi tiga *range*.

1. PN ($-90^\circ - 10^\circ$)
2. PC ($-30^\circ - 30^\circ$)
3. PP ($-10^\circ - 90^\circ$)

Secara matematis fungsi keanggotaan *pitch* dapat dituliskan sebagai berikut.

1. Himpunan PN

$$\mu_{PN}(x) = \begin{cases} 1 & , x \leq -30 \\ \frac{-x}{30} & , -30 < x < 0 \\ 0 & , x \geq -10 \end{cases}$$

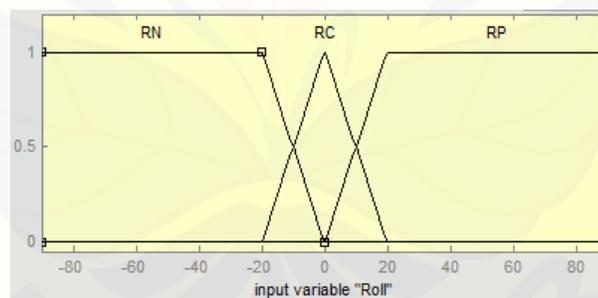
2. Himpunan PC

$$\mu_{PC}(x) = \begin{cases} 0 & , x \leq -30 \text{ atau } x \geq 10 \\ \frac{x-30}{5} & , -30 < x \leq -10 \\ \frac{-10-x}{5} & , -10 < x \leq 10 \\ 1 & , x = -10 \end{cases}$$

3. Himpunan PP

$$\mu_{PP}(x) = \begin{cases} 0 & , x \leq -10 \\ \frac{x}{10} & , 0 < x < 10 \\ 1 & , x \geq 10 \end{cases}$$

Berikut fungsi keanggotaan sudut kemiringan *roll* dengan nilai linguistik RN, RC, dan RP



Gambar 3.33 Fungsi keanggotaan *roll*

Dari fungsi keanggotaan *roll*, dapat diketahui nilai tengah yaitu pada nilai 0 yang merupakan sudut kemiringan 90° dari roket. Dari sudut ini ditentukan untuk roket berada pada kondisi lurus. Pada input sudut kemiringan *roll* dibagi menjadi tiga *range*.

1. RN ($-90^\circ - 0^\circ$)
2. RC ($-20^\circ - 20^\circ$)
3. RP ($0^\circ - 90^\circ$)

Secara matematis fungsi keanggotaan *roll* dapat dituliskan sebagai berikut.

1. Himpunan RN

$$\mu_{RN}(x) = \begin{cases} 1 & , x \leq -20 \\ \frac{-x}{20} & , -20 < x < 0 \\ 0 & , x \geq 0 \end{cases}$$

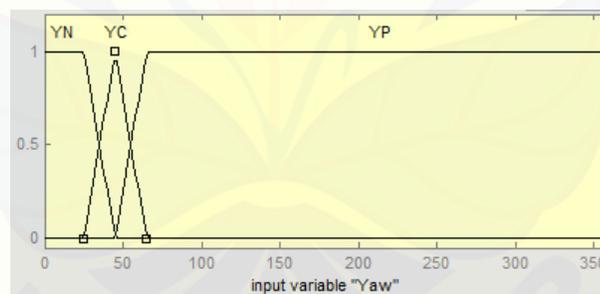
2. Himpunan RC

$$\mu_{RC}(x) = \begin{cases} 0 & , x \leq -20 \text{ atau } x \geq 20 \\ \frac{x+20}{20} & , -30 < x \leq -10 \\ \frac{20-x}{20} & , 0 < x \leq 20 \\ 1 & , x = 0 \end{cases}$$

3. Himpunan RP

$$\mu_{RP}(x) = \begin{cases} 0 & , x \leq 0 \\ \frac{x}{20} & , 0 < x < 20 \\ 1 & , x \geq 20 \end{cases}$$

Berikut fungsi keanggotaan sudut kemiringan *yaw* dengan nilai linguistik YN, YC dan YP.



Gambar 3.34 Fungsi keanggotaan *yaw*

Dari fungsi keanggotaan *yaw*, dapat diketahui nilai tengah yaitu pada nilai 45 yang merupakan sudut kemiringan 45° dari roket. Berbeda dengan masukan *pitch* dan *roll* yang memiliki *range* sampai 180°, pada masukan *yaw* memiliki *range* 360°. Pada input sudut kemiringan *yaw* dibagi menjadi tiga *range*.

1. YN (0° - 45°)
2. YC (25° - 65°)
3. RP (45° - 360°)

Secara matematis fungsi keanggotaan *yaw* dapat dituliskan sebagai berikut.

1. Himpunan YN

$$\mu_{YN}(x) = \begin{cases} 1 & , x \leq 25 \\ \frac{x}{25} & , 25 < x < 45 \\ 0 & , x \geq 45 \end{cases}$$

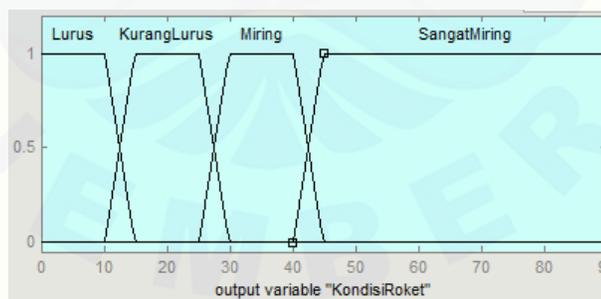
2. Himpunan YC

$$\mu_{YC}(x) = \begin{cases} 0 & , x \leq 25 \text{ atau } x \geq 65 \\ \frac{x+25}{20} & , 25 < x \leq 45 \\ \frac{65-x}{20} & , 45 < x \leq 65 \\ 1 & , x = 45 \end{cases}$$

3. Himpunan YP

$$\mu_{YP}(x) = \begin{cases} 0 & , x \leq 45 \\ \frac{x}{65} & , 65 < x < 360 \\ 1 & , x \geq 65 \end{cases}$$

Selanjutnya pada perancangan fungsi keanggotaan keluaran hanya digunakan untuk mengambil keputusan berupa hasil dari kondisi roket dan menentukan lebar pulsa yang diberikan pada motor servo. Berikut fungsi keanggotaan keluaran yang berupa kondisi dari roket dengan nilai linguistik Lurus, Kurang Lurus, Miring, dan Sangat Miring.



Gambar 3.35 Fungsi keanggotaan keluaran kondisi roket

Dari fungsi keanggotaan keluaran kondisi terdapat empat nilai kondisi, dari kondisi tersebut menggambarkan kondisi roket pada saat bekerja. *Range* kondisi roket sebesar 90° , berikut range kondisi dalam setiap nilai kondisi roket.

1. Lurus ($0^\circ - 15^\circ$)
2. Kurang Lurus ($10^\circ - 30^\circ$)
3. Miring ($25^\circ - 45^\circ$)
4. Sangat Miring ($40^\circ - 90^\circ$)

Secara matematis fungsi keanggotaan kondisi dapat dituliskan sebagai berikut.

1. Himpunan Lurus

$$\mu_{Lurus}(x) = \begin{cases} 1 & , x \leq 10 \\ 15 - x & , 10 < x < 15 \\ 0 & , x \geq 15 \end{cases}$$

2. Himpunan Kurang Lurus

$$\mu_{Kurang\ Lurus}(x) = \begin{cases} 0 & , x \leq 10 \\ x - 10 & , 10 < x \leq 15 \\ 1 & , 15 < x \leq 25 \\ 30 - x & , 25 < x \leq 30 \end{cases}$$

3. Himpunan Miring

$$\mu_{Miring}(x) = \begin{cases} 0 & , x \leq 25 \\ x - 25 & , 25 < x \leq 30 \\ 1 & , 30 < x \leq 40 \\ 45 - x & , 40 < x \leq 45 \end{cases}$$

4. Himpunan Sangat Miring

$$\mu_{Sangat\ Miring}(x) = \begin{cases} 0 & , x \leq 40 \\ 45 - x & , 40 < x < 45 \\ 1 & , x \geq 45 \end{cases}$$

b. Rule Base Inference

Pada *rule base inference*, terjadi proses pengolahan data masukan *fuzzyfikasi* dengan keluaran berupa kondisi yang dikehendaki dengan aturan-aturan (*rule*) tertentu. Dari aturan-aturan yang dibentuk inilah yang nantinya akan menentukan respon dari sistem terhadap berbagai kondisi yang terjadi ketika roket bekerja. Berikut *rule base inference* pada keluaran Kondisi yang memiliki 27 aturan *fuzzy*.

1. Jika Pitch = PC dan Roll = RC dan Yaw = YC maka Kondisi = Lurus
2. Jika Pitch = PN dan Roll = RC dan Yaw = YC maka Kondisi = Kurang Lurus

3. Jika Pitch = PP dan Roll = RC dan Yaw = YC maka Kondisi = Kurang
Lurus

Sampai dengan

27. Jika Pitch = PP dan Roll = RN dan Yaw = YP maka Kondisi = Sangat
Miring

c. *Defuzzyfication*

Defuzzyfication merupakan pemetaan bagi nilai-nilai *fuzzy* keluaran yang dihasilkan pada tahap *rule inference* ke nilai-nilai keluaran kuantitatif. Pada perancangan sistem kontrol posisi dan keseimbangan roket ini proses *defuzzyfikasi* menggunakan metode *Weighted Average*. Keluaran dari proses *defuzzyfikasi* berupa Kondisi yang nantinya digunakan untuk mengontrol motor servo menggunakan lebar pulsa yang diberikan sesuai dengan hasil keluaran Kondisi.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, diperoleh kesimpulan sebagai berikut:

1. Berdasarkan data pengujian kontrol sirip roket menghasilkan sudut bebas sebesar 60° dalam mengontrol posisi dan keseimbangan roket (dilihat pada tabel 4.8 dan tabel 4.9).
2. Berdasarkan pengujian sistem kontrol posisi dan keseimbangan, maka dapat diambil kesimpulan sebagai berikut :
 - a. Pengujian awal roket dapat bertahan di udara dalam menjaga posisi dan keseimbangannya selama 2 detik dengan sudut target yang terpenuhi sampai pada 1000 *milisecond* pada sudut *pitch* dengan nilai rata-rata *error* 25,63% dan sampai pada 750 *milisecond* pada sudut *roll* dengan nilai rata-rata *error* 25,14% (dilihat pada tabel 4.17 dan tabel 4.18). Serta pada pengujian akhir roket dapat bertahan menjaga posisi dan keseimbangannya selama 5 detik di udara dengan sudut target yang terpenuhi sampai pada waktu 3500 *milisecond* pada sudut *pitch* dengan nilai rata-rata *error* 13,56 % (dilihat pada tabel 4.21) dan sampai pada waktu 2700 *milisecond* pada sudut *roll* dengan nilai rata-rata *error* 14,44 % (dilihat pada tabel 4.22 dan tabel 4.23).
 - b. Roket gagal memenuhi target pada sudut *yaw* karena kondisi roket yang melintir pada saat meluncur di udara dengan nilai rata-rata *error* 32,35 % pada pengujian awal (dilihat pada tabel 4.19 dan tabel 4.20) dan nilai rata-rata *error* 25,61 % pada pengujian akhir (dilihat pada tabel 4.24 dan tabel 4.25). Roket melintir disebabkan oleh putaran dari motor *brushless* EDF yang besar dan interferensi angin.

- c. Perancangan roket uji akhir menghasilkan keberhasilan sistem lebih baik dari perancangan roket uji awal dengan hasil pengujian pada uji awal menghasilkan nilai rata-rata *error* sebesar 32,25 % dengan lama meluncur selama 2 detik dan pada roket uji akhir menghasilkan nilai rata-rata *error* sebesar 25,61 % dengan lama meluncur selama 5 detik.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran terkait penelitian ini adalah sebagai berikut:

1. Diperlukan penelitian lebih lanjut tentang aerodinamik perancangan mekanik roket, sehingga dapat membenahi kondisi roket yang melintir saat meluncur di udara.
2. Diperlukan motor *brushless* dengan daya dorong (*trus*) yang lebih besar agar roket mendapatkan daya dorong yang lebih besar dari berat roket itu sendiri.
3. Diperlukan *Ground Control System* untuk mengontrol roket secara langsung melalui aplikasi *Ground Segment* pada PC.

DAFTAR PUSTAKA

- Adli, Muhammad. 2014. Desain dan Implementasi Sistem Telemetry Wireless Untuk Mendeteksi Kinerja Alat Berat Menggunakan Accelerometer Berbasis Arduino dan Zigbee. *Jurnal Jurusan Teknik Elektro Universitas Telkom*.
- Alfiansyah, Muhammad. et all. 2012. Rancang Bangun Inertial Measurement Unit Untuk Unmanned Aerial Vehicles “Quadrotor”. *Jurnal Teknik ITS Vol. 1, No 1*.
- Anonim Arduino <http://www.arduino.cc/en/Main/ArduinoBoardUno>, (Diakses pada 16 Maret 2015).
- Anonim CMPS11 http://www.robot-electronics.co.uk/htm/arduino_examples (Diakses 25 Maret 2016).
- Anonim Contoh Pemrograman Arduino : <http://github.com/>, (Diakses 20 Maret 2016).
- Anonim Kumpulan Teori *Fuzzy Logic* <http://sonoku.com/>, (Diakses 20 Maret 2016).
- Anonim Teori Kontrol PID : <http://menanamilmu.blogspot.com/2010/09/teori-kontrol-pid-proportionalintegrald.html>, (Diakses pada 5 Maret 2015).
- Anonim PID dan Roket <http://id.wikipedia.org/wiki/>, (Diakses pada 2 Maret 2015).
- Anonim PID Tuning : <http://Elektroindonesia-tutorial.html>, (diakses 15 Desember 2015).
- Bobby Grace, 2015. Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontrol. *Jurnal ELKOMIKA Teknik Elektro Itenas, No. 2 Vol. 3*.
- Hidayat Nur. 2009. *Rancang Bangun Sistem Kendali Quadrotor Untuk Keseimbangan Posisi dengan PID*. Depok, Indonesia : Skripsi.
- Mehran Kamyar. 2008. *Takagi Sugeno Fuzzy Modeling for Process Control*. School of Electrical Electronic and Computer Engineering. Newcastle University.
- Mulyana, Agus., Subhan, Andriyana. 2012. Rancang Bangun Sistem Kendali Posisi, Keseimbangan, dan Navigasi Untuk Prototipe Nano Satelit. *Jurnal Sistem Komputer Unikom, Komputika Volume 1, No. 1*.

- Naba Agus, 2009. *Belajar Cepat Fuzzy Logic Menggunakan MATLAB*. Yogyakarta: CV. ANDI OFFSET.
- Noor, E. S. 2013. Implementasi Sensor Magnetometer dan Akselerometer Untuk Memonitor Arah Muatan Roket. *Jurnal Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya*.
- Pudjo Widodo Prabowo, *et all*, 2012. *Penerapan Soft Computing Dengan MATLAB*. Bandung: Rekayasa Sains.
- Shiddiq Anwar, 2012. *Rancang Bangun Prototipe Sistem Aktuator Kendali Sirip Menggunakan Brushless DC Motor*. Skripsi. Depok: Universitas Indonesia.
- Yagusandri Ariel, 2011. *Rancang Bangun Prototipe Sistem Aktuator Sirip Roket Menggunakan Motor Servo*. Skripsi. Depok, Universitas Indonesia.
- Yusuf, Z., dan Fatoni, A. 2012. Perancangan dan Implementasi Kontroler PID Untuk Autonomus Moving Forward Pada Quadcopter. *Jurnal Teknik POMITS Vol.1, No. 1, 1-6*.

LAMPIRAN

1. Roket EDF



Perancangan rangka prototipe roket



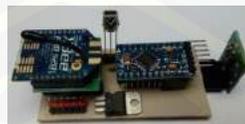
Perancangan komponen roket di dalam rangka roket



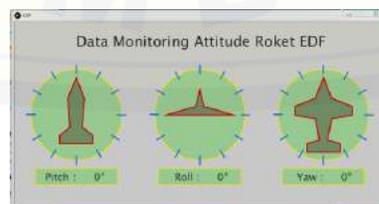
Proses pemrograman kontroler roket



Uji coba kemiringan sirip roket



Board IMU roket



Perancangan aplikasi *Ground Segment* Roket



Pengujian sistem telemetri pada aplikasi *Ground Segment*

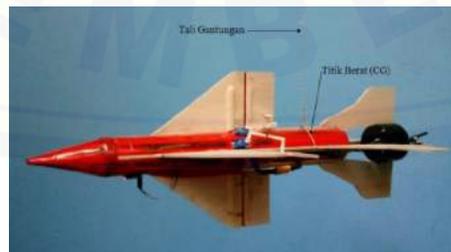
2. Roket EDF Pada Pengujian Tanpa Uji Terbang



Pengujian roket dengan posisi digantung



Pengambilan data pada pengujian tanpa uji terbang



Kondisi roket saat digantung

3. Roket EDF Pada Pengujian Awal

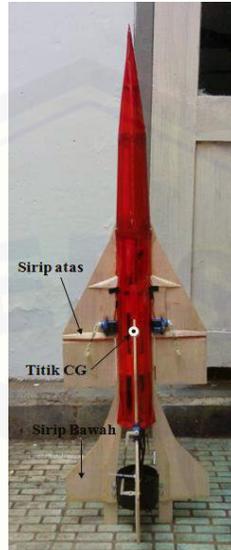


Perancangan akhir roket pada pengujian awal



Kondisi roket saat meluncur di udara

4. Roket EDF Pada Pengujian Akhir



Perancangan akhir roket pada pengujian akhir



Kondisi roket saat meluncur di udara



Kondisi roket saat kehilangan keseimbangan