



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**BUKU LAPORAN  
PROYEK AKHIR**

## **PROYEK PAYLOAD DAN PELUNCUR ROKET EZ RX-100**

### **DESAIN DAN SISTEM PENGENDALIAN PELUNCUR ROKET CERDAS BERDASARKAN KECEPATAN DAN ARAH ANGIN**

**EDWIN RAMADHANI S**  
NRP. 7506 040 010

Dosen Pembimbing I:

Dr. Ir. Endra Pitowarno, M.Eng  
NIP. 196206301987011001

Dosen Pembimbing II:

Nu Rhahida Arini, ST, MT  
NIP. 197401202008122002

**JURUSAN TEKNIK MEKATRONIKA  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2011**



## **BUKU LAPORAN PROYEK AKHIR**

### **PROYEK PAYLOAD DAN PELUNCUR ROKET EZ RX-100**

**DESAIN DAN SISTEM PENGENDALIAN PELUNCUR ROKET  
CERDAS BERDASARKAN KECEPATAN DAN ARAH ANGIN**

Oleh:

**EDWIN RAMADHANI S**  
**NRP. 7506 040 010**

**Dosen Pembimbing I:**

**Dr. Endra Pitowarno, M Eng**  
**NIP. 196206301987011001**

**Dosen Pembimbing II:**

**Nu Rhahida Arini, ST, MT**  
**NIP. 197401202008122002**

**JURUSAN TEKNIK MEKATRONIKA  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2011**

**PROYEK PAYLOAD DAN PELUNCUR ROKET EZ RX-100  
DESAIN DAN SISTEM PENGENDALIAN PELUNCUR ROKET  
CERDAS BERDASARKAN KECEPATAN DAN ARAH ANGIN**

**Edwin Ramadhani Sampurna**  
**NRP. 7506.040.010**

**Proyek Akhir ini Digunakan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar  
Sarjana Sains Terapan (S.ST)**

**di**  
**Politeknik Elektronika Negeri Surabaya**  
**Institut Teknologi Sepuluh Nopember Surabaya**

**Disetujui oleh**

**Dosen Penguji :**

1.

**Dr. Indra Adji Sulistijono, ST, M.Eng**  
**NIP. 196705271994031018**

2.

**Endah Suryawati N, ST, MT**  
**NIP. 197501122000122001**

3.

**Eko Henfri Binugroho, S.ST, M.Sc**  
**NIP. 197912232003121002**

**Dosen Pembimbing :**

1.

**Dr. Ir. Endra Pitowarno, M.Eng**  
**NIP. 196206301987011001**

2.

**Nu Rhahida Arini, ST, MT**  
**NIP. 197401202008122002**

**Mengetahui,**  
**Ketua Jurusan Teknik Mekatronika**

**Dr. Ir. Endra Pitowarno, M.Eng**  
**NIP. 196206301987011001**

## ABSTRAK

Pada Tugas Akhir ini akan dianalisa dan diimplementasikan kontrol posisi dengan menggunakan PID (*Proportional, Integratif dan Derivative*) melalui sebuah plant peluncur roket dengan dua derajat kebebasan konfigurasi sendi-peluncur roket yang dikendalikan oleh dua buah motor DC. *Input* pergerakan sendi-peluncur roket akan dikendalikan oleh sebuah Mikrokontroler dan *Simulink* MATLAB. Masukan yang diberikan adalah berupa posisi tujuan dan konstanta PID. Kontrol peluncur roket sendiri menggunakan motor DC yang dihitung berdasarkan arah pergerakan tanpa memperhatikan aspek dinamik motor. Dalam menuju proses implementasi ini maka peluncur roket dua derajat kebebasan dirancang untuk dapat mengerjakan fungsi khusus melakukan gerakan naik-turun dan berputar hingga 360 derajat. Peluncur roket dilengkapi dengan sensor penentuan arah dan kecepatan mata angin untuk dapat membuat peluncur rocket yang dapat mengetahui arah angin dan derajat pergeseran arah angin sehingga dapat diketahui dimana arah luncur roket yang tepat. Sensor arah angin tersebut akan dijadikan sebagai input ke *system* untuk nilai parameter kontrol PID. Sensor angin ini didalamnya terdapat serangkaian potensio linear *continuous* yang jika diputar 1 kali putaran penuh nilai outputnya akan kembali ke nilai awal, output sensor ini adalah resistor variabel yang difungsikan sebagai pembagi tegangan agar dapat diambil data ke ADC mikrokontroler. Dari hasil tersebut Maka didapat hasil 5 Volt per 1 kali putaran penuh, yang dibutuhkan sensor posisi angin didalam menentukan posisi derajat arah angin. Dari hasil pengujian disimpulkan bahwa nilai potensio 0.01 Volt per 1 derajat. Nilai tersebut dimasukkan sebagai nilai parameter kontrol PID yang dapat digunakan untuk merancang sistem peluncur roket yang lebih kompleks dengan *input* tujuan.

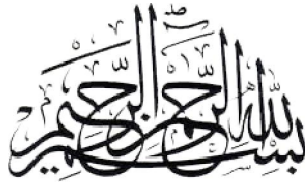
**Kata kunci :** Peluncur rocket, PID, Sensor arah angin, ADC, Mikrokontroler, Resistor variable, Potensiometer

## **ABSTRACT**

*At this final task will be analyzed and implemented position control using PID (Proportional, Integratif and Derivative) controller in a plant of two degree-of-freedom Rocket Launcher with joint-Launcher configuration controlled by two DC motor. Movement input of rocket launcher will be controlled by a Mikrokontroler using C program and Matlab Simulink application. The input has given by the form of target position and PID constanta. Controlling the rocket launcher use the DC motor which calculated movement direction and regardless dynamic motor constanta. This implementation process border with special purpose of rocket launcher two degrees freedom is designed to be able to perform the specific function rise-down movement and rotates up to 360 degrees. Rocket launcher equipped with sensor environment wind velocity and direction to make rocket launchers that could see the wind direction changes so that know where the rocket launching at the right time. Sensor environment wind velocity and direction will be used as input to the system for the value of PID control parameters. This wind sensor circuit in which there is a continuous linear potensio 1 time round if played full value of its output will return to their original values, output of this sensor is used as a variable resistor voltage divider can be taken for the data to the microcontroller ADC. So the result is 5 Volt full rotations per 1 time, wind position sensor is needed in determining the position of the degree of wind direction. From the test results concluded that the value potensio 0.01 Volt / 1 degree. This value is entered as the value of PID control parameters that can be used to design a rocket launcher system that is more complex with the input goal.*

**Keywords :** *Rocket Launcher, PID, sensor environment wind velocity and direction*

# KATA PENGANTAR



Syukur Alhamdulillah saya panjatkan kepada Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya sehingga dapat terselesaikannya Proyek Akhir ini yang berjudul :

## **“ DESAIN DAN SISTEM PENGENDALIAN PELUNCUR ROKET CERDAS BERDASARKAN KECEPATAN DAN ARAH ANGIN ”**

Proyek Akhir ini dibuat dengan maksud untuk memenuhi salah satu persyaratan guna menyelesaikan *studi* di Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember.

Dalam menyelesaikan Proyek Akhir ini, kami melaksanakan berdasarkan teori-teori yang telah kami peroleh dalam perkuliahan, *literature* dan bimbingan dari dosen pembimbing serta pihak-pihak lain yang telah banyak memberikan semangat dan bantuan.

Penyusun sadar bahwasanya masih banyak kesalahan dan kekurangan dalam penyusunan buku laporan Proyek Akhir ini, untuk itu penyusun mohon maaf dan mengharapkan kritik dan saran guna penyempurnaan pada buku selanjutnya. Selain itu juga diharapkan pembaca dapat mengembangkan Proyek Akhir ini. Semoga buku ini memberikan manfaat dalam menghasilkan sumber daya manusia yang berkualitas.

Surabaya, Februari 2011

Penyusun

## UCAPAN TERIMA KASIH

Puji syukur yang tak terhingga saya sampaikan kepada Allah SWT Yang Maha Berkuasa Atas Segalanya, karena hanya dengan ridho, hidayah dan anugerah-Nya saya dapat menyelesaikan proyek akhir ini. Ucapan terima kasih saya sampaikan juga ke berbagai pihak yang turut membantu memperlancar penyelesaian proyek akhir ini, yaitu kepada: Ibu, hanya air mata yang bisa menetes setiap kali ingin ku ucapkan rasa syukur ini. Terimakasih atas doa, cinta kasih sayang dan dukungannya, Untuk saudara sedaraku, Angga Aditya S, Reza Fahrel S , Ibu Nu Rhahida Arini, ST, MT selaku dosen pembimbing, yang telah sabar dan arif dalam membimbing dan mendampingi saya. Maaf jika selalu meropotkan saat demo, Bu Arini baik sekali mau datang ke lab 😊. Bapak Dr. Endra Pitowarno, M Eng, selaku dosen pembimbing yang banyak memberikan masukan, arahan, semangat, Bapak Ir. Dadet Pramadihanto, M.Eng. PhD, selaku Direktur Politeknik Elektronika Negeri Surabaya, Bapak Dr.Ir. Son Kuswadi, Dr.Indra Adji Sulistijono,ST.M.Eng, bu Endah Suryawati Ningrum, ST.,MT dan bapak Didik Setyo Purnomo, ST, M.Eng selaku dosen penguji Terima kasih telah membantu menyempurnakan proyek akhir ini, dan teman-teman yang lain tetep semangat pantang menyerah. Terimakasih untuk sharing-sharing ilmunya, Seluruh keluarga besarku D4 Mekatronika. Perjuangan kalian untuk menyelesaikan project. Forever we will be together a family.

Segala ucapan terima kasih tentunya belum cukup, semoga Allah SWT senantiasa membalas segala kebaikan Anda. Amin.

# DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
KATA PENGANTAR.....	v
UCAPAN TERIMA KASIH.....	vi
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	xi

## BAB I PENDAHULUAN

1.1 LATAR BELAKANG .....	1
1.2 PERUMUSAN MASALAH .....	2
1.3 BATAS MASALAH .....	2
1.4 TUJUAN.....	2
1.5 METODOLOGI .....	2
1.6 SISTEMATIKA STUDI .....	5

## BAB II STUDI PENDAHULUAN

2.1 ROCKET LAUNCHER.....	7
2.2 KONFIGURASI MANIPULATOR .....	9
2.3 SISTEM KONTROL ROCKET LAUNCHER .....	10
2.4 SENSOR ARAH DAN POSISI ANGIN .....	20
2.5 TRANSMISI GEAR .....	22
2.6 MOTOR DC.....	24

## BAB III DESAIN DAN PROSES PEMBUATAN PROYEK AKHIR

3.1 DESAIN MEKANIK.....	27
3.2 PEMBUATAN MEKANIK.....	31
3.2.1 MEKANIK ROCKET LAUNCHER BAGIAN BAWAH ...	31
3.2.2 MEKANIK ROCKET LAUNCHER BAGIAN ATAS.....	32
3.3 DESAIN SISTEM.....	35
3.4 DESAIN ELEKTRONIKA.....	35
3.5 PEMBUATAN SISTEM ELEKTRONIK.....	39
3.6 DESAIN SENSOR ANGIN.....	40



3.7 PEMBUATAN SENSOR ANGIN.....	42
---------------------------------	----

#### BAB IV PENGUJIAN DAN ANALISA SISTEM ROKET LAUNCHER

4.1 UJI COBA PERANGKAT PELUNCUR ROKET .....	43
4.1.1 PENGUJIAN KONTROL PID.....	43
4.1.2 PENGUJIAN DRIVER MOTOR DC.....	48
4.1.3 PENGUJIAN DATA ADC .....	51
4.1.4 PENGUJIAN SENSOR WIND VANE DIRECTION.....	57
4.1.5 PENGUJIAN SENSOR ANEMOMETER.....	62
4.2 UJI COBA KESELURUHAN SISTEM .....	66

#### BAB V KESIMPULAN DAN SARAN

5.1 KESIMPULAN.....	75
5.2 SARAN .....	76

## DAFTAR GAMBAR

Gambar 1.1	Perencanaan Penyelesaian TA dalam Flowchar.....	4
Gambar 2.1	Anatomi rocket launcher untuk RUM(LAPAN)....	7
Gambar 2.2	Sistem rocket launcher.....	8
Gambar 2.3	Konfigurasi sendi-lengan.....	9
Gambar 2.4	Kontrol lengan loop terbuka.....	10
Gambar 2.5	Kontrol lengan loop tertutup.....	11
Gambar 2.6	Penggunaan transformasi Laplace.....	13
Gambar 2.7	Ilustrasi Rocket launcher dengan satu sendi.....	13
Gambar 2.8	Rocket launcher dengan satu sendi.....	14
Gambar 2.9	Diagram kontrol rocket Launcher satu sendi.....	14
Gambar 2.10	Kontrol rocket launcher tertutup.....	16
Gambar 2.11	Kontrol Proporsional, P.....	16
Gambar 2.12	Kontrol Integral, I.....	17
Gambar 2.13	Kontrol Proporsional Integral,P-I.....	17
Gambar 2.14	Kontrol Derivatif, D.....	18
Gambar 2.15	Kontrol PID.....	19
Gambar 2.16	Potensio RS spedtrol no 173-596.....	21
Gambar 2.17	Incremental Rotary Ecoder.....	22
Gambar 2.18	Transmisi Gear.....	23
Gambar 2.19	Motor DC.....	24
Gambar 3.1	Foto peluncur Rocket.....	27
Gambar 3.2	Desain 3D peluncur rocket.....	28
Gambar 3.3	Rancangan rocket launcher bagian atas.....	28
Gambar 3.4	Rancangan rocket launcher bagian bawah.....	29
Gambar 3.5	Desain peluncur roket tampak 2D.....	29
Gambar 3.6	Desain kaki dan badan peluncur roket.....	30
Gambar 3.7	Desain kaki penyangga peluncur roket.....	30
Gambar 3.8	Foto mekanik rocket launcher bagian bawah.....	32
Gambar 3.9	Foto mekanik rocket launcher bagian atas.....	32
Gambar 3.10	Gear bevel merubah posisi motor dari horisontal ke vertikal.....	33
Gambar 3.11	Foto Bagian atas Rocket Launcher.....	33
Gambar 3.12	Mekanik rocket launcher bagian bawah menggunakan Worm gear.....	34
Gambar 3.13	Foto rocket launcher bagian bawah.....	34
Gambar 3.14	Desain Sistem.....	35

Gambar 3.15	Rangkaian hibridge satu arah.....	36
Gambar 3.16	Rangkaian hibridge dua arah .....	38
Gambar 3.17	Rangkaian sistem Mikrokontroler.....	39
Gambar 3.18	Foto rangkaian sistem mikrokontroler dan rangkaian driver motor .....	40
Gambar 3.19	Mekanik Sensor penunjuk angin.....	41
Gambar 3.20	Mekanik Sensor kecepatan angin.....	41
Gambar 3.21	Foto mekanik sensor angin.....	42
Gambar 4.1	Blok Diagram kontrol PID .....	45
Gambar 4.2	Blok Diagram kontrol P .....	45
Gambar 4.3	Kontrol P dengan respon sistem terhadap gangguan.....	46
Gambar 4.4	Blok Diagram kontrol PD.....	46
Gambar 4.5	Kontrol PD dengan respon sistem terhadap gangguan.....	47
Gambar 4.6	Blok Diagram kontrol PID .....	47
Gambar 4.7	Kontrol PID dengan respon sistem terhadap gangguan.....	48
Gambar 4.8	Diagram pengujian motor DC .....	49
Gambar 4.9	Blok Diagram pengujian Sensor potensio linear .....	52
Gambar 4.10	Gambar potensio linear .....	52
Gambar 4.11	Grafik sensor posisi.....	55
Gambar 4.12	Blok Diagram pengujian Sensor Arah Angin.....	57
Gambar 4.13	Gambar sensor arah angin.....	58
Gambar 4.14	Grafik sensor arah angin.....	60
Gambar 4.15	Blok Diagram pengujian Sensor Anemometer .....	62
Gambar 4.16	Gambar sensor Anemometer .....	63
Gambar 4.17	Grafik sensor kecepatan angin.....	65
Gambar 4.18	Blok diagram keseluruhan tugas akhir.....	67
Gambar 4.19	FlowChart – PID Kontrol.....	68
Gambar 4.20	FlowChart – Manual Kontrol.....	72

## TABLE LIST

Tabel 2.1	DATA MOTOR .....	25
Tabel 4.1	DATA SENSOR POSISI .....	54
Tabel 4.2	DATA SENSOR ARAH ANGIN .....	59
Tabel 4.3	DATA SENSOR ANEMOMETER .....	64

# **BAB I**

## **PENDAHULUAN**

### **1.1 LATAR BELAKANG**

Setelah cukup lama, terhitung sejak tahun 1963 LAPAN (Lembaga Penerbangan dan Antariksa Nasional) mengadakan penelitian dan pengembangan roket berukuran kecil (RX 70, RX 100, RX 150, dan RX 250) melalui lomba muatan/payload antar perguruan tinggi di Indonesia. Permasalahannya adalah peluncur roket LAPAN dianggap masih belum bisa meluncurkan roket secara otomatis, umumnya LAPAN menggunakan peluncur roket manual yang hanya bisa diatur derajat atau posisi peluncur roket pada saat roket mulai diluncurkan. Hal ini dianggap kurang efisien untuk itu kami berencana membantu LAPAN untuk meningkatkan teknologi dibidang pengembangan peluncur roket menggunakan sistem kontrol cerdas berdasarkan kecepatan dan posisi angin disekitar secara otomatis.

Kendali digital menjadi sangat penting dalam industri proses karena berbagai keuntungan yang ditawarkan. Hal ini didukung dengan perkembangan teknologi perangkat keras yang tetap dengan perangkat lunak yang fleksibel serta dapat melakukan integrasi dan diferensiasi sinyal yang sangat lambat. Tetapi kendali digital juga memiliki beberapa kelemahan, untuk itu didesain suatu kontrol sistem yang memiliki kemampuan mengidentifikasi sekaligus mengkoreksi kesalahan output sistem dengan menggunakan kontrol PID (Proportional, Integratif dan Derivative).

Pada Tugas Akhir ini akan dianalisa dan diimplementasikan kontrol posisi dengan menggunakan PID melalui sebuah plant peluncur roket dengan dua derajat kebebasan konfigurasi sendi-peluncur roket yang dikendalikan oleh dua buah motor DC. Masukan pergerakan sendi-peluncur roket akan dikendalikan oleh sebuah komputer dengan aplikasi Matlab dan Simulink. Masukan yang diberikan adalah berupa posisi tujuan dan konstanta PID. Kontrol peluncur roket sendiri menggunakan motor DC yang dihitung berdasarkan arah pergerakan tanpa memperhatikan aspek dinamik motor. Dalam menuju proses implementasi ini maka peluncur roket dua derajat kebebasan dirancang untuk dapat mengerjakan fungsi khusus melakukan gerakan naik-turun dan berputar hingga 360 derajat.

Dengan data tersebut diharapkan dapat diketahui nilai parameter kontrol PID yang dapat digunakan untuk merancang sistem peluncur

roket yang lebih kompleks dengan input tujuan dan berat sendi yang sama.

## **1.2 PERUMUSAN MASALAH**

Permasalahan yang dibahas dalam tugas akhir ini adalah bagaimana mendesain dan mengimplementasikan kontrol peluncur roket dengan menggunakan sensor kecepatan dan sensor posisi angin yang menghubungkannya secara interface dari mikrokontroler ke motor peluncur roket untuk menunjukkan gerak sesuai arah posisi angin dan kecepatan angin sehingga posisi arah luncur roket yang ideal dapat diketahui.

## **1.3 BATASAN MASALAH**

Pada penyelenggaraan proyek akhir ini, batasan permasalahannya adalah :

1. Input sensor yang digunakan bergantung pada keadaan angin lingkungan.
2. Data yang akan dianalisa adalah data-data posisi arah angin yang berada di wilayah pantai.
3. Perencanaan sensor arah angin sebagai alat untuk mengetahui posisi luncur roket yang ideal dan untuk data sensor kecepatan angin hanya digunakan sebagai informasi kecepatan angin saja.
4. Pemrograman menggunakan Matlab 7.1 untuk proses simulasi dan pengontrolan peluncur roket.

## **1.4 TUJUAN**

Tujuan dari proyek ini adalah diharapkan dapat membantu lembaga LAPAN untuk mempelajari, merencanakan dan membuat sebuah sistem kontrol yang fungsinya sebagai peningkatan dibidang roket.

## **1.5 METODOLOGI**

Dalam pengerjaan tugas akhir ini, dilakukan dalam beberapa tahap, sebagai berikut: tahap studi literatur, tahap permodelan dan simulasi, tahap perancangan, pembuatan 2 derajat kebebasan pada peluncur roket, tahap pengujian sistem, tahap analisa dan eksperimen. Penjelasan lebih detail adalah di bawah ini:

### **1. Studi Literatur**

Studi Permodelan dan simulasi dari sistem kontrol 2 DOF dengan menggunakan Matlab-Simulink beserta kontroler yang digunakan kemudian mensimulasikan unjuk kerjanya.

### **2. Perencanaan Sistem**

Dari data yang didapatkan, selanjutnya dilakukan analisa dan perencanaan terhadap sistem yang meliputi analisa data linearisasi kerja sensor dimana menjadikan sensor mempunyai karakteristik posisi-kecepatan yang linear.

### **3. Pembuatan Sistem**

Pembuatan peluncur roket dari kontrol 2 DOF dengan merangkai dua buah motor dan gear sedemikian hingga menjadi gerak dua derajat kebebasan arah dan untuk bagian elektronik dibuat driver untuk mengatur arus yang masuk ke motor.

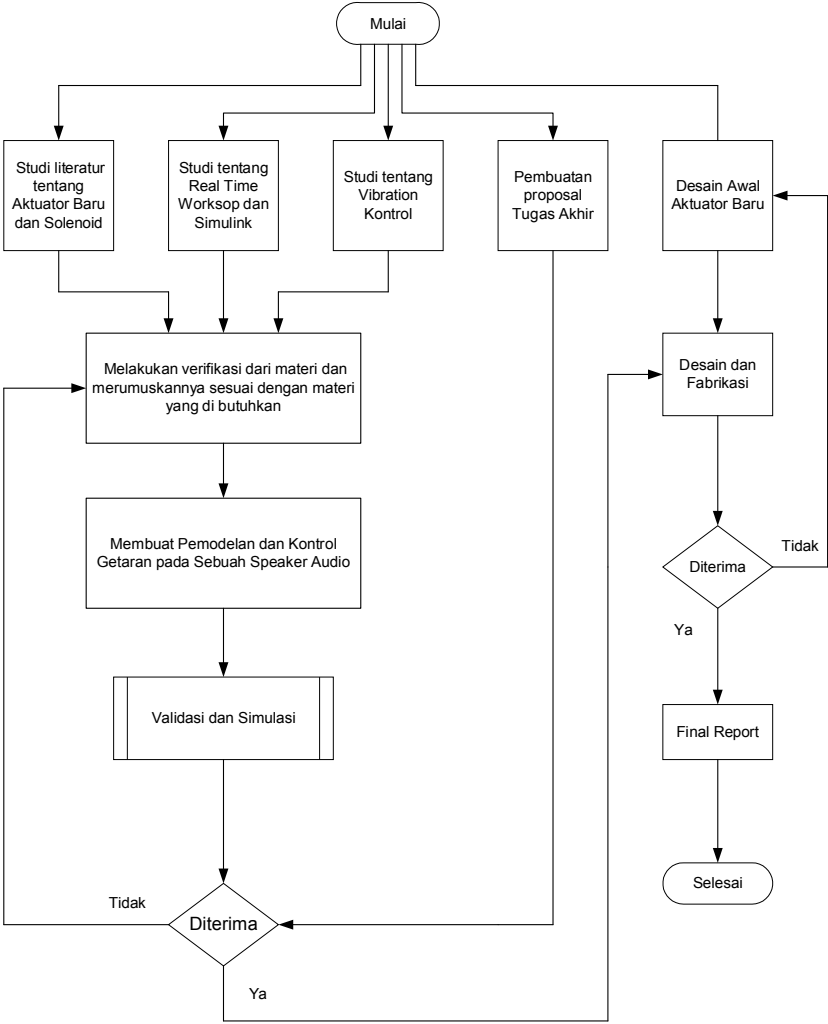
### **5. Pengujian dan Evaluasi**

Pengujian dan evaluasi dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat pada proyek akhir ini dapat berfungsi sesuai dengan proses sistem yang diharapkan.

### **6. Pembuatan Laporan**

Membuat dokumentasi dari semua tahapan proses di atas berupa laporan yang berisi tentang dasar teori dan hasil proyek akhir ini.

Strategi dari pengerjaan proyek akhir ini dibuat dalam bentuk flowchart seperti gambar 1.1:



**Gambar 1.1:** Perencanaan Penyelesaian Tugas Akhir dalam Flowchart



## **1.6 SISTEMATIKA STUDI**

Sistematika pembahasan yang akan diurai dalam buku laporan proyek akhir ini terbagi dalam bab-bab yang akan dibahas sebagai berikut :

- BAB I       PENDAHULUAN**  
Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang, tujuan proyek akhir, perumusan masalah, batasan masalah, metodologi, serta sistematika pembahasan dari proyek akhir ini.
- BAB II       STUDI PENDAHULUAN**  
Bab ini membahas mengenai teori-teori yang berkaitan dengan penyelesaian proyek akhir, yang didapatkan dari berbagai macam buku serta sumber-sumber terkait lainnya yang berhubungan dengan pembuatan proyek akhir ini.
- BAB III      DESAIN PROYEK AKHIR**  
Bab perencanaan dan pembuatan alat membahas tentang ide-ide dan gambaran tentang alat yang akan dibuat. Bagian ini akan menampilkan komponen yang dipergunakan, blok diagram, gambar tiap modul dan penjelasan singkatnya serta perancangan dan pengujian tiap-tiap bagian.
- BAB IV      UJI COBA DAN ANALISA**  
Bab ini menyajikan dan menjelaskan seluruh hasil dan analisa dalam pembuatan proyek akhir. Membahas tentang hasil dari alat yang dibuat, apakah alat tersebut telah bekerja sesuai dengan yang diinginkan.
- BAB V       PENUTUP**  
Bab ini kesimpulan dari hasil pengerjaan tugas akhir dan saran atas beberapa hal yang dianggap perlu mendapatkan perhatian khusus.

*Halaman ini sengaja dikosongkan*

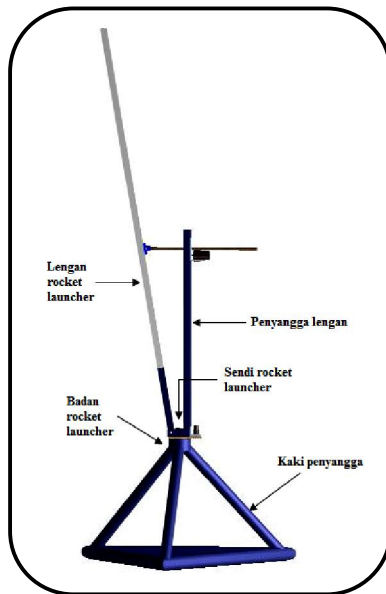
## BAB II STUDI PENDAHULUAN

Teori-teori yang akan dijadikan materi penunjang dalam pembuatan proyek akhir ini akan dibahas pada bab ini.

### 2.1. ROCKET LAUNCHER

Rocket Launcher yang dimiliki LAPAN secara umum dapat diilustrasikan seperti seperti pada gambar 2.1

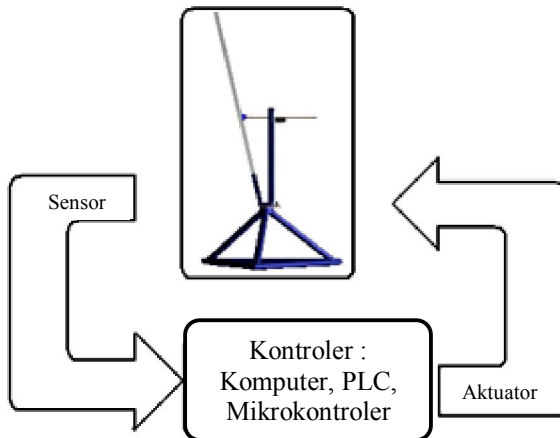
Rocket launcher yang ini adalah rocket launcher yang memiliki satu lengan dan satu sendi. Di dalam pergelangan dapat diinstal berbagai tool sesuai dengan fungsi yang diharapkan. Jika dipandang dari sudut pergerakan maka terdiri dari dua pergerakan utama, yaitu badan rocket launcher yang dapat berputar ke kiri dan kanan, lengan yang dapat bergerak rotasi ke arah atas dan bawah. [LAPAN, 2009]



**Gambar 2.1:** Anatomi rocket launcher untuk rocket tipe RUM (LAPAN)

Perangkat pendukung rocket launcher secara umum dapat diilustrasikan dalam gambar 2.2 berikut ini. Komponen utamanya terdiri dari 4 bagian, yaitu:

- Manipulator
- Sensor
- Aktuator, dan
- Kontrol



**Gambar 2.2:** *Sistem rocket launcher*

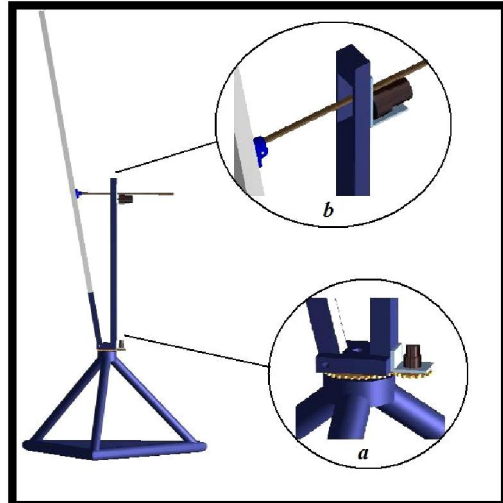
Manipulator adalah bagian mekanik yang dapat difungsikan untuk memindahkan, mengangkat dan mengarahkan rocket. Sensor adalah komponen berbasis instrumentasi (pengukuran) yang berfungsi sebagai pemberi informasi tentang berbagai keadaan atau kedudukan dari bagian-bagian manipulator. Output sensor dapat berupa nilai logika ataupun nilai analog. [Walden, 1999]

Aktuator adalah komponen penggerak yang jika dilihat dari prinsip penghasil geraknya dapat dibagi menjadi 3 bagian, yaitu penggerak berbasis motor listrik (motor DC servo, stepper motor, motor AC, dsb.) [Sclater, 2007] penggerak pneumatik (berbasis kompresi gas: udara, nitrogen, dsb.), dan penggerak hidrolik (berbasis kompresi benda cair: minyak pelumas, dsb.). [Baumann, Oliver, 2008]

Kontroler adalah rangkaian elektronik berbasis mikroprosesor yang berfungsi sebagai pengatur seluruh komponen dalam membentuk fungsi kerja. Tipe pengaturan yang bisa diprogramkan mulai dari prinsip pengurut (sequencer) yang bekerja secara open loop hingga prinsip umpan balik dengan melibatkan kecerdasan buatan. [Shaller, Robert, 2004]

## 2.2. KONFIGURASI MANIPULATOR

Secara klasik konfigurasi manipulator dapat dibagi dalam 4 kelompok yaitu polar, slinder, cartesian dan sendi-lengan (joint-arm).[Pieper, 1968] Didalam proyek akhir ini konfigurasi pada rocket launcher yang digunakan adalah konfigurasi manipulator struktur sendi-lengan. Konstruksi ini adalah yang paling populer untuk mengangkat rocket, memutar rocket dan mengarahkan rocket untuk meluncur sesuai derajat yang diinginkan. Dengan struktur lengan-sendi ini cocok digunakan untuk mengarahkan daerah kerja yang akan diluncurkan dengan sudut jangkau yang beragam.



**Gambar 2.3:** Konfigurasi sendi-lengan,  
*a. gear sendi peluncur roket, b.ulir lengan peluncur roket*

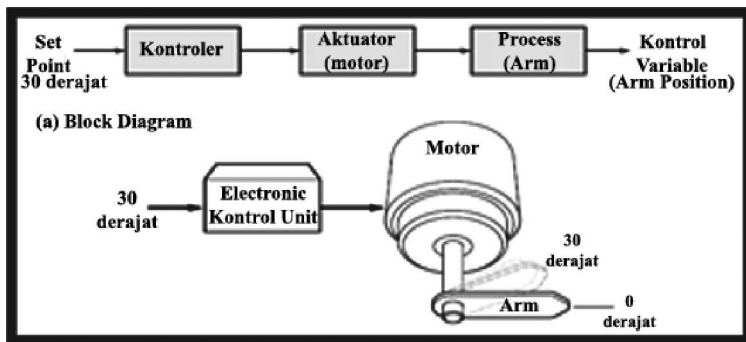
Gambar 2.3 adalah gambar rocket launcher yang memiliki konfigurasi sendi-lengan. Pada ujung lengan peluncur roket dipasang ulir yang dapat bergerak rotasi kearah atas dan bawah. Pada sendi dipasang gear agar dapat berputar ke kiri dan kanan.

Struktur rocket launcher ini banyak kajian kinematik dan dinamik gerak mengambil contoh struktur sendi-lengan, kajian dinamiknya menjadi lebih rumit karena unsur gravitasi sedemikian hingga faktor gravitasi dapat diabaikan dalam analisa dinamiknya. Dengan menghindari pengaruh gravitasi order dinamik dari sistem yang dikontrol relative dapat ditekan. Salah satu pertimbangannya adalah bahwa aplikasi robotic di angkasa luar justru tidak perlu memasukkan unsur grafitasi dalam analisa. Dinamik hanya dipengaruhi oleh faktor inerti body robot itu sendiri. [Pitowarno, 2006]

### 2.3. SISTEM KONTROL ROCKET LAUNCHER

Sistem kontrol pada dasarnya terbagi dua klompok, yaitu sistem kontrol loop terbuka (open loop) dan loop tertutup (close loop).

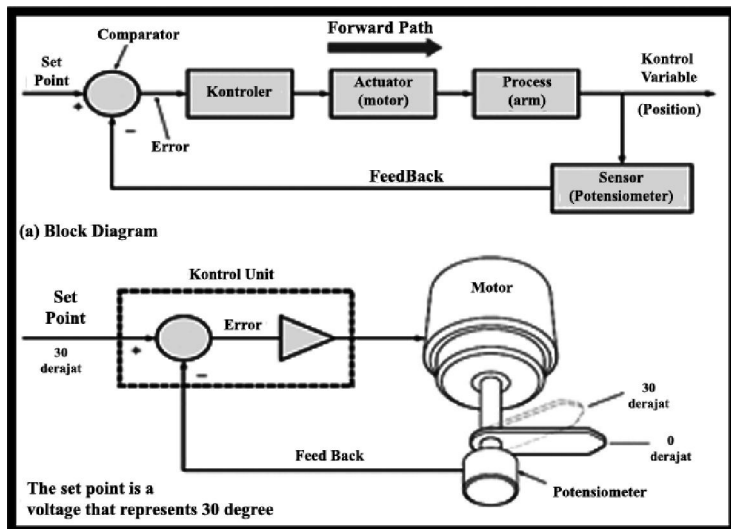
Diagram kontrol loop terbuka pada sistem lengan dapat dinyatakan dalam gambar 2.4 berikut ini.



Gambar 2.4: Kontrol lengan loop terbuka

Kontrol loop terbuka atau umpan maju (feed forward control) dengan dinyatakan sebagai sistem kontrol outputnya tidak diperhitungkan ulang oleh kontroler. [Kuo, Benjamin C, 1991] Keadaan apakah lengan rocket launcher benar-benar telah mencapai target seperti yang dikehendaki sesuai referensi, adalah tidak dapat mempengaruhi kinerja kontroler. Kontrol ini sesuai untuk sistem operasi yang memiliki actuator yang beroperasi berdasarkan umpan logika berbasis konfigurasi langkah sesuai urutan. Perlu digaris bawahi di sini bahwa kontrol sekuensi (urutan) dalam gerak dalam suatu tugas yang lengkap, misalnya memiliki urutan sebagai berikut: menuju ke posisi obyek, mengambil obyek, mengangkat obyek, adalah tidak selalu semua langkah operasi ini termasuk dalam kontrol loop terbuka. Dapat saja langkah menuju posisi obyek dan memindah obyek menuju posisi akhir adalah gerak berdasarkan loop tertutup. Sedangkan yang lainnya adalah loop terbuka berdasarkan perintah langkah berbasis delay.

Diagram kontrol loop terbuka pada sistem lengan tertutup dapat dinyatakan seperti pada gambar 2.5.



Gambar 2.5: Kontrol lengan loop tertutup

Pada gambar di atas, jika hasil gerak aktual telah sama dengan referensi maka input kontroler akan nol. Artinya kontroler tidak lagi memberikan sinyal aktuasi kepada motor karena target akhir perintah gerak telah diperoleh. Makin kecil error terhitung maka makin kecil pula sinyal pengemudian kontroler terhadap motor, sampai akhirnya mencapai kondisi tenang (steady state).

Referensi gerak dan gerak aktual dapat berupa posisi (biasanya didefinisikan melalui kedudukan ujung lengan terakhir), kecepatan, akselerasi, atau gabungan diantaranya. Kontrol bersifat konvergen jika dalam rentang waktu pengontrolan nilai error menuju nol, dan keadaan dikatakan stabil jika setelah konvergen kontroler mampu menjaga agar error selalu nol. Dua pengertian dasar, konvergen dan stabil, adalah sangat penting dalam kontrol tertutup. Stabil dan konvergen diukur dari sifat referensinya. Posisi akhir dianggap konvergen bila makin lama gerak makin perlahan dan akhirnya diam pada posisi seperti yang dikehendaki referensi, dan dikatakan stabil jika posisi akhir yang diam ini dapat dipertahankan dalam masa-masa berikutnya. Jika referensinya adalah kecepatan maka disebut stabil jika pada keadaan tenang kecepatan akhirnya adalah sama dengan referensi (atau mendekati) dan kontroler mampu menjaga 'kesamaan' ini pada masa-masa berikutnya. Dalam hal kecepatan, keadaan tenang yang dimaksud adalah bukan berarti output kontroler bernilai nol (tegangan nol Volt) seperti keadaan sesungguhnya pada kontrol posisi, namun kontroler tidak lagi memberikan penguatan (amplify) atau pelemahan (attenuate) pada aktuator. Demikian juga bila referensinya adalah percepatan (akselerasi). [Franklin, 2002]

### **2.3.1 Penggunaan Transformasi Laplace**

Sistem kontrol pada dasarnya secara teoritis memerlukan dasar-dasar pemahaman tentang sistem sinyal. Semua gerakan yang diasumsikan sebagai visualisasi operasi sistem berbasis waktu yang berjalan dapat dinyatakan sebagai fungsi sinyal. Artinya, karakteristik input (referensi), sistem kontrol, sistem yang dikontrol dan output dapat

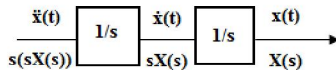


dinyatakan dalam persamaan matematik yang mempresentasikan sifat atau respon terhadap perubahan waktu.

Transformasi Laplace adalah salah satu metoda untuk menyatakan persamaan sinyal dalam fungsi waktu. Metode ini sangat berguna dalam analisa sinyal untuk kontrol robotic, bentuk dasar ekspresi matematiknya adalah sebagai berikut,

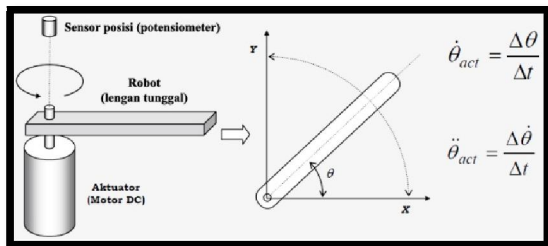
$$L\{f(t)\} \int_0^{\infty} f(t)e^{-st} dt \dots\dots\dots (2.1)$$

Dengan pendekatan ini jika  $L\{x(t)\} = X(s)$  dengan asumsi nilai pada kondisi awal adalah nol, maka  $L\{\dot{x}(t)\} = sX(s)$ . Demikian juga maka  $L\{\ddot{x}(t)\} = s^2X(s)$ . Jika  $x(t)$  adalah fungsi dari posisi,  $\dot{x}(t)$  adalah kecepatan, dan  $\ddot{x}(t)$  adalah percepatan maka penggunaan transformasi laplace [Pitowarno, 2006] dapat diilustrasikan seperti dalam gambar 2.6 berikut ini.

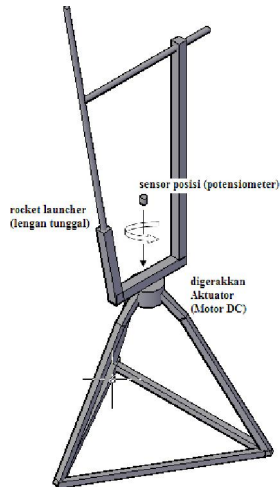


**Gambar 2.6:** *Penggunaan transformasi Laplace*  
**Sumber:** (Pitowarno, E. (2006). ROBOTIKA)

Sebagai contoh rocket launcher yang memiliki lengan tunggal atau satu sendi. Aktuatornya adalah sebuah motor DC sedang sensornya adalah potensiometer. Ilustrasinya diberikan dalam gambar 2.7 berikut ini.

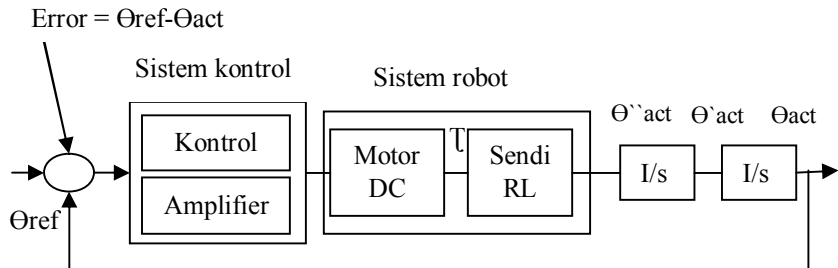


**Gambar 2.7:** *Ilustrasi rocket launcher dengan satu sendi*  
**Sumber:** (Pitowarno, E. (2006). ROBOTIKA)



**Gambar 2.8:** *rocket launcher dengan satu sendi*

Gambar di atas dapat dikontrol berdasarkan posisi, kecepatan dan percepatan. Di sebelah kanan gerakan lengan diilustrasikan dapat membentuk gerakan searah dan berlawanan dengan jarum jam. Posisi gerakan dinyatakan oleh  $\theta$ . Diagram kontrolnya dinyatakan seperti dalam gambar 2.9 di bawah ini.



**Gambar 2.9:** *Diagram kontrol rocket Launcher satu sendi*

**Sumber:** (Pitowarno, E. (2006). ROBOTIKA)

Dalam gambar,  $\theta_{ref}$  adalah posisi referensi (dalam radian),  $\theta_{ref}$  adalah posisi actual,  $I$  adalah arus motor,  $K_{tn}$  adalah konstanta motor,  $\tau$  adalah torsi yang dihasilkan poros motor,  $\ddot{\theta}_{act}$  adalah percepatan sudut actual,  $\dot{\theta}_{act}$  adalah kecepatan sudut actual, dan  $\theta_{act}$  adalah posisi actual.

Gambar 2.9 di atas sebenarnya tidak berbeda dengan ilustrasi pada gambar 2.6 dinyatakan sebagai sistem kontrol pada gambar 2.9 yang terdiri dari blok kontrol (program atau algoritma dalam computer atau rangkaian prosesor) dan blok ampliflier yang berfungsi sebagai interface ke actuator. Motor dan perangkat mekanik lengan rocket launcher berada dalam satu blok yang disebut sebagai sistem roket launcher.

Referensi gerak disisi adalah posisi, sehingga error yang dihitung adalah selisih posisi referensi dengan posisi actual. Namun ilustrasi pada gambar 2.9 dapat memberikan penjelasan yang lebih rinci tentang fenomena kontrol yang sesungguhnya terjadi. Pemberian torsi oleh motor pada lengan rocket launcher memberikan dampak dinamik pada percepatan sudut, kecepatan sudut dan posisi sudut.

Dalam aplikasi yang sebenarnya pernyataan pada gambar 2.9 memberikan pengertian bahwa output dari sistem rocket launcher (putaran sudut pada poros atau sendi lengan) dapat dibaca dalam tiga parameter ini tidak selalu diakomodasikan dengan pemasangan sensor yang bersesuaian. Jika sensor yang tersedia adalah sensor posisi saja maka kecepatan dapat diperoleh dengan mengintegrasikan bacaan posisi terhadap waktu sebagai berikut,

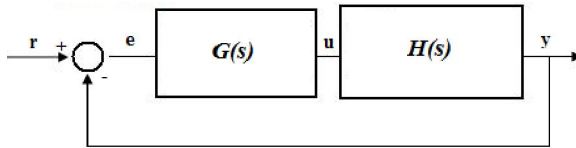
$$\dot{\theta} = \frac{\Delta\theta}{\Delta t} \dots\dots\dots (2.2)$$

Demikian juga, percepatan dapat diperoleh dengan mengintegrasikan perhitungan kecepatan terhadap waktu [Bracewell, 2000],

$$\ddot{\theta}_{act} = \frac{\Delta\dot{\theta}}{\Delta t} \dots\dots\dots (2.3)$$

### 2.3.2 Kontroler proporsional, integral dan derivative

Kembali pada gambar 2.6, dalam bentuk pernyataan standar dalam sistem kontrol seperti dalam gambar 2.10 dalam gambar,  $r$  adalah input,  $e$  adalah error,  $u$  adalah sinyal output kontroler,  $G(s)$  adalah  $H(s)$  adalah dinamik rocket launcher, dan  $y$  adalah output. Sekarang permasalahannya adalah bagaimana  $G(s)$  didisain. [Pitowarno, 2006]



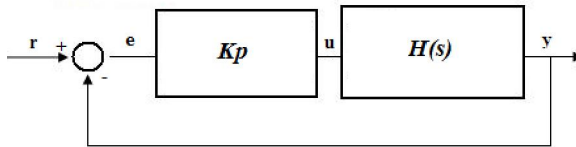
Gambar 2.10: Kontrol rocket launcher tertutup

#### 2.3.2.1 Kontrol Proporsional (Propotional Control, P)

Kontrol adalah kontrol P jika  $G(s) = k$ , dengan  $k$  adalah konstanta. Jika  $u = G(s) \cdot e$  maka,

$$u = K_p \cdot e \dots\dots\dots (2.4)$$

Dengan  $K_p$  adalah konstanta propotional.  $K_p$  berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Dengan demikian gambar 2.11 dapat dinyatakan ulang sebagai berikut,



Gambar 2.11: Kontrol Proporsional, P

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

mencapai konvergensi meskipun error keadaan tenangnya (steady-state error) relatif besar. Sebagai materi pembelajaran, kontrol P dianggap sangat baik untuk permulaan. [Pitowarno, 2006]

### 2.3.2.2 Kontrol Integral (Integral Control, I)

Jika  $G(s)$  adalah kontrol I maka  $u$  dapat dinyatakan sebagai,  $u(t) = \left[ \int_0^t e(T)dT \right] Ki$ ,  $Ki$  adalah konstanta Integral dari persamaan di atas,  $G(s)$  dapat dinyatakan sebagai,

$$G(s) = \frac{Ki}{s} \dots\dots\dots (2.5)$$

Jika  $e(T)$  mendekati konstanta (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.

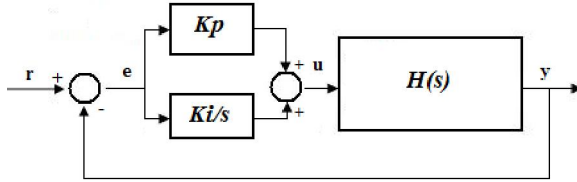
Kontrol I dapat memperbaiki respon steady-state, namun pemilihan  $Ki$  yang tidak tepat dapat menyebabkan respon transien (transient response) yang tinggi sehingga dapat menyebabkan ketidak stabilan sistem. Pemilihan  $Ki$  yang sangat tinggi justru dapat menyebabkan output berosilasi [Pitowarno, 2006]. Diagram kontrol I dapat diilustrasikan sebagai berikut,



Gambar 2.12: Kontrol Integral, I

### 2.3.2.3 Kombinasi kontrol P dan I

Dengan sifat dasar kontrol P yang cenderung konvergen dan I yang dapat memperbaiki respon steady-state maka kombinasi P-I dapat memberikan hasil yang lebih baik [Pitowarno, 2006]. Dalam diagram blok dapat dinyatakan sebagai berikut.



**Gambar 2.13:** Kontrol Proporsional Integral, P-I

Dari gambar 2.13 di atas , persamaan kontroler  $G(s)$  dapat dinyatakan sebagai berikut,

$$G(s) = Kp + \frac{Ki}{s} \dots\dots\dots (2.6)$$

atau

$$G(s) = \frac{sKp+Ki}{s} \dots\dots\dots (2.7)$$

### 2.3.2.4 Kontrol Derivatif (Derivatif Control, D)

Sinyal kontrol  $u$  yang dihasilkan oleh kontrol D dapat dinyatakan sebagai,

$$u = Kd \cdot \dot{e} \dots\dots\dots (2.8)$$

atau

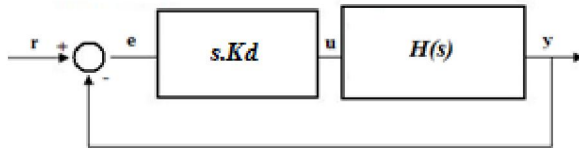
$$u = Kd \cdot \frac{\Delta e}{\Delta t} \dots\dots\dots (2.9)$$

Sehingga  $G(s)$  dapat dinyatakan,

$$G(s) = s \cdot Kd \dots\dots\dots (2.10)$$

Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini bermain dalam konteks “kecepatan” atau rate dari error. Dengan demikian sifat ini ia dapat digunakan memprediksi error yang akan terjadi. Umpan balik yang diberikan adalah kecepatan perubahan  $e(t)$

sehingga kontroler dapat mengantisipasi error yang akan terjadi. [Pitowarno, 2006] Dalam blok diagram dapat dinyatakan sebagai berikut,

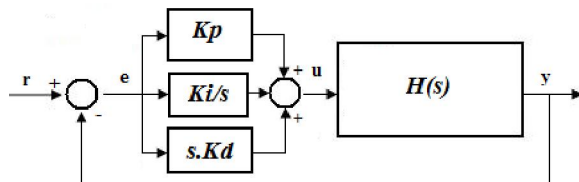


**Gambar 2.14:** Kontrol Derivatif,  $D$

### 2.3.2.5 Kombinasi kontrol P, I dan D

Diagram kombinasi ketiga kontrol klasik yang diterangkan di atas dapat dinyatakan seperti dalam gambar 2.15. Dengan menggabungkan kontrol P, I dan D maka masing-masing kelebihan dapat disatukan untuk mendapatkan kontrol yang ideal.

Namun demikian, suatu sistem kontrol klasik kombinasi, baik PI ataupun PID, hanya dapat bekerja baik untuk sistem  $H(s)$  yang cenderung linear dalam fungsi waktu. Artinya, persamaan dinamik dari model  $H(s)$  relative tidak berubah selama rentang waktu pengontrolan. Padahal kenyataannya, tidak ada fenomena sistem riil yang benar-benar linear. Bahkan hampir semua fenomena kontrol mulai dari skala kecil, hingga skala besar. Setiap sistem riil selalu berhadapan dengan gangguan (disturbance). Motor selalu bermasalah dengan friksi pada poros, gearbox, perubahan karakteristik karena temperature, dll. Pesawat di udara selalu berhadapan dengan tekanan udara yang berubah-ubah, angin, hujan, dsb.



**Gambar 2.15:** Kontrol Proporsional Integral Derivatif, PID

Untuk kontrol klasik ini, yang dapat dilakukan oleh engineer hanyalah melakukan pendekatan atau asumsi model sistem secara linear dengan mengabaikan faktor-faktor nonlinear yang dianggap terlalu sulit untuk dimodelkan secara matematik. Sehingga  $K_p$ ,  $K_i$  dan  $K_d$  yang dipilih (tuned) adalah yang dianggap paling tepat (optimum) untuk kondisi ideal model. [Pitowarno, 2006]

## **2.4. SENSOR ARAH ANGIN**

Masalah dalam penentuan arah dan kecepatan mata angin adalah perlunya sistem pemantau arah angin yang dapat mengetahui derajat arah angin menjadi sistem penentuan derajat arah angin, dan derajat pergeserannya secara tepat dan cepat untuk setiap saat.

### **2.4.1 Sistem penentu derajat arah angin**

Untuk dapat membuat alat yang dapat mengetahui arah angin dan derajat pergeseran arah angin, dapat menggunakan sistem potensiometer linear. [Johns Hopkins, 1969]

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel. potensiometer berperan sebagai resistor variabel atau Rheostat. Potensiometer dibuat dari sebuah unsur resistif semi-lingkar dengan sambungan geser (penyapu). Unsur resistif, dengan terminal pada salah satu ataupun kedua ujungnya, berbentuk datar atau menyudut. Pada potensiometer panel, terminal penyapu biasanya terletak ditengah-tengah kedua terminal unsur resistif. Untuk potensiometer putaran tunggal, penyapu biasanya bergerak kurang dari satu putaran penuh sepanjang kontak. Potensiometer "putaran ganda", elemen resistifnya mungkin berupa pilinan dan penyapu bergerak 10, 20, atau lebih banyak putaran untuk menyelesaikan siklus. Pada potensiometer geser linier, sebuah kendali geser digunakan sebagai ganti kendali putar. Potensiometer linier mempunyai unsur resistif dengan penampang konstan, menghasilkan piranti dengan resistansi antara penyapu dengan salah satu terminal proporsional dengan jarak



antara keduanya. Potensiometer linier digunakan jika relasi proporsional diinginkan antara putaran sumbu dengan rasio pembagian dari potensiometer. [Piscataway and New Jersey, 2000]



**Gambar 2.16:** *Potensiometer RS spectrol no173-596*

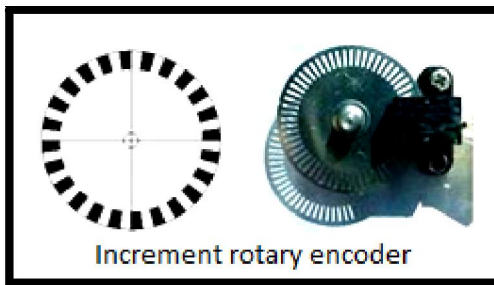
**Sumber:** (<http://docs-europe.electrocomponents.com/webdocs>)

## 2.4.2 Sistem penentu kecepatan angin

Untuk dapat membuat alat yang dapat mengetahui kecepatan pergeseran arah angin menggunakan sistem rotary encoder model seperti tampak pada gambar 2.17, [Johns Hopkins, 1969] Rotary encoder umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh rotary encoder untuk diteruskan oleh rangkaian kendali. Rotary encoder umumnya digunakan pada pengendalian robot, motor drive, dsb.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis dipasang dengan poros motor, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila

posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Gambar di bawah menunjukkan bagan skematik sederhana dari rotary encoder. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi rotary encoder tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi rotary encoder tersebut.



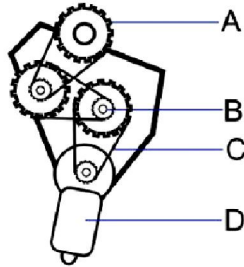
**Gambar 2.17:** *Incremental Rotary encoder*

**Sumber:** ([http://en.wikipedia.org/wiki/Rotary\\_encoder](http://en.wikipedia.org/wiki/Rotary_encoder))

Sistem ini, dalam bentuk yang paling sederhana, tidak dapat mengukur sudut absolut dari poros. Sistem ini hanya dapat mengukur perubahan sudut relatif menjadi data yang berubah-ubah, seperti posisi poros pada saat power di-ON-kan. Ketidakpastian posisi ini tidak menjadi masalah untuk menentukan kecepatan angin. [[http://en.wikipedia.org/wiki/Rotary\\_encoder](http://en.wikipedia.org/wiki/Rotary_encoder)]

## 2.5. TRANSMISI GEAR

Untuk menggerakkan peluncur roket dibutuhkan bantuan dari gear, dan jika hanya memakai gear maka peluncur roket tidak dapat bergerak naik karena beban yang akan diangkat cukup berat. maka digunakanlah transmisi gear, karena transmisi gear mempunyai torsi yang cukup besar.



**Gambar 2.18:** *Transmisi Gear*

**Sumber:** ([http://en.wikipedia.org/wiki/Transmission\\_\(mechanics\)\)](http://en.wikipedia.org/wiki/Transmission_(mechanics)))

Keterangan gambar

- A. Gear 2
- B. Gear 1
- C. Rantai kamrat
- D. Motor 2

Berikut ini adalah perhitungan dari transmisi gear untuk mengangkat beban. Misalkan :

Gear 1 = 12 gigi

Gear 2 = 40 gigi

Perbandingan gear 1 : gear 2 = 3 : 10

Transmisi gear (n) = 3

Panjang Sandaran (P) = 0.6m

Torsi motor (T) = 3.77Nm

untuk menghitung torsi gear

$$\begin{aligned}
 T_n &= T (\text{ gear 2 / gear 1 } )n \\
 &= 3.77 ( 10 / 3 ) 3 \\
 &= 139.6
 \end{aligned}$$

menghitung momen torsi

$$\begin{aligned}
 T &= \text{Momen torsi} \\
 T &= F ( \frac{1}{2} P ) \\
 139.6 &= F 0.3
 \end{aligned}$$

$$F = 465.3$$

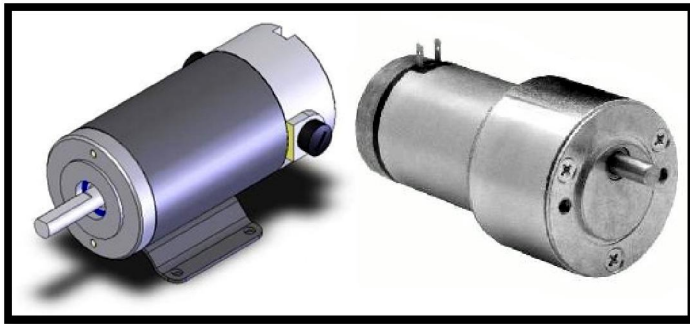
menghitung beban yang dapat diangkat

$$m = F / g \text{ (gravitasi)}$$

$$m = 456.3 / 9.81$$

$$m = 47.43\text{Kg (beban yang mampu diangkat)}$$

## 2.6. MOTOR DC



**Gambar 2.19:** *Motor DC*

**Sumber:** ([http://en.wikipedia.org/wiki/Torque\\_and\\_speed\\_of\\_a\\_DC\\_motor](http://en.wikipedia.org/wiki/Torque_and_speed_of_a_DC_motor))

**Tabel 2.1:** *Data Motor*

Motor 1	Motor 2
Voltage : 12.01(v)	Voltage : 12.01(v)
Torque : 4.31(Nm)	Torque : 3.77 (Nm)
Speed : 47.9 (rpm)	Speed : 36.4 (rpm)
Current: 5.69 (A)	Current: 3.23 (A)
Output Power : 21.59(w)	Output Power : 14.391(w)

Untuk motor ke-1 menggunakan arus 5.69 A karena motor digunakan untuk menggerakkan gear dan butuh kecepatan yang tinggi dan torsi yang besar, sedangkan untuk motor yang ada mampu mengangkat beban sampai 47Kg.

## **BAB III**

### **DESAIN DAN PROSES PEMBUATAN PROYEK AKHIR**

Pada bab ini akan menjelaskan tentang langkah-langkah rancangan desain dan proses pembuatan.

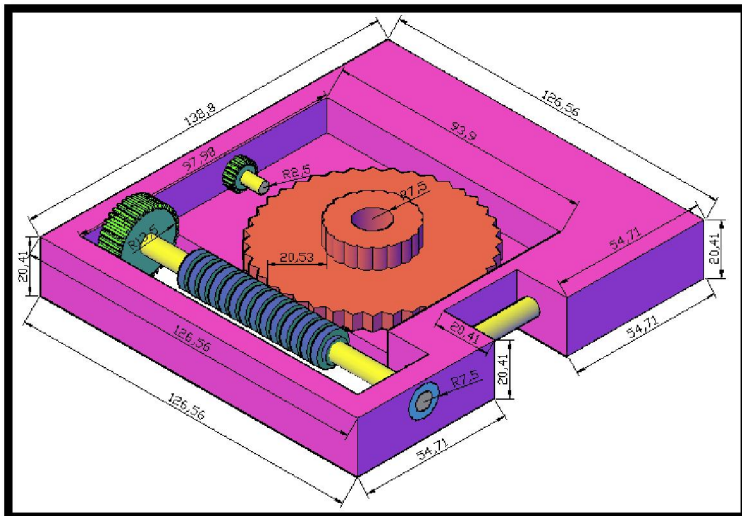
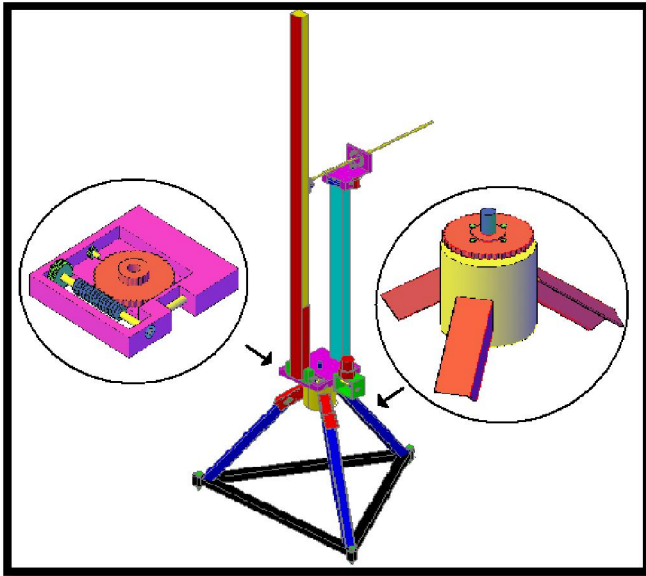
#### **3.1. DESAIN MEKANIK**

Mekanik yang dirancang disesuaikan dengan dimensi model pada peluncur roket sering digunakan oleh LAPAN pada saat lomba. Penyesuaian pada mekanik peluncur roket LAPAN adalah bagian yang terpenting karena ukuran atau bentuk mekaniknya sudah disesuaikan oleh roket itu sendiri. Di bawah ini adalah foto standar dimensi ruang untuk peluncur roket.

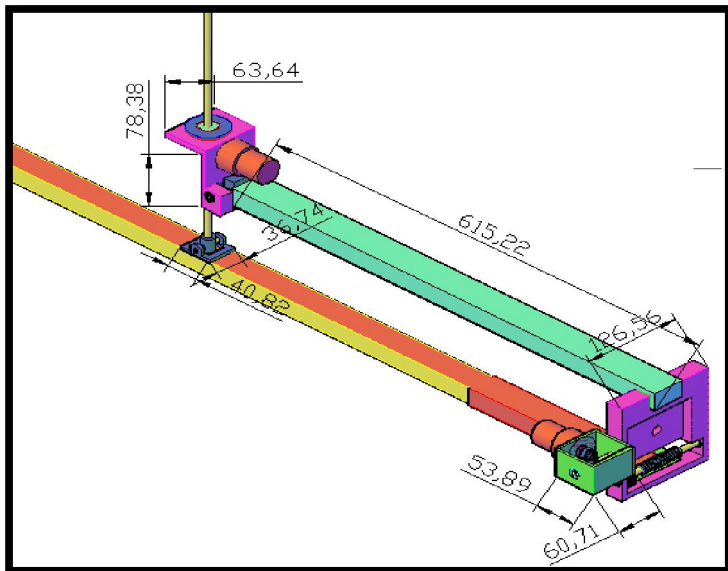
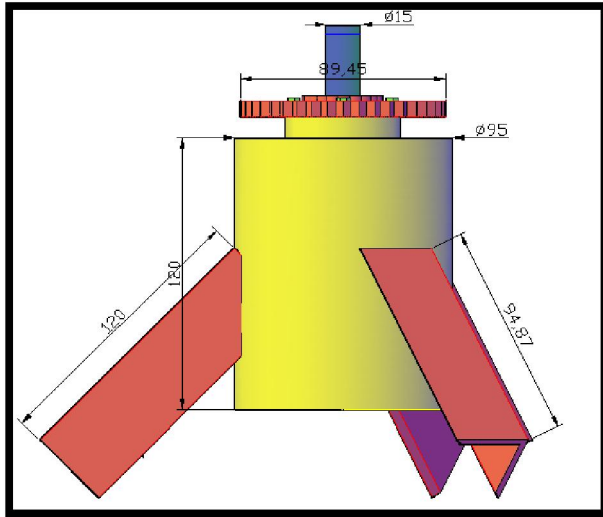


**Gambar 3.1:** *Foto Peluncur Roket.*

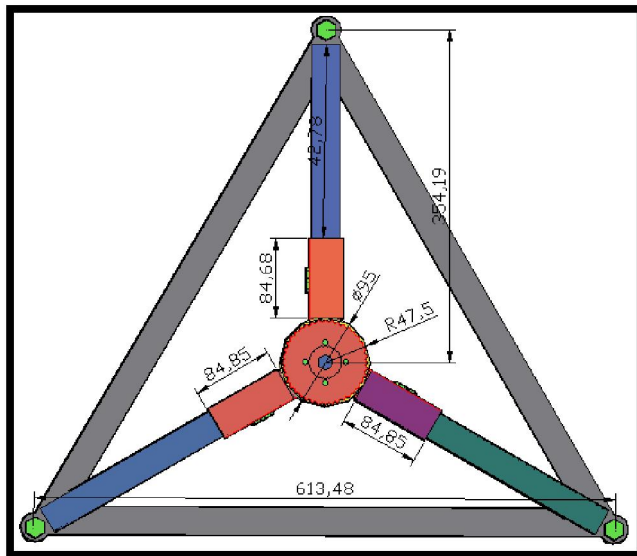
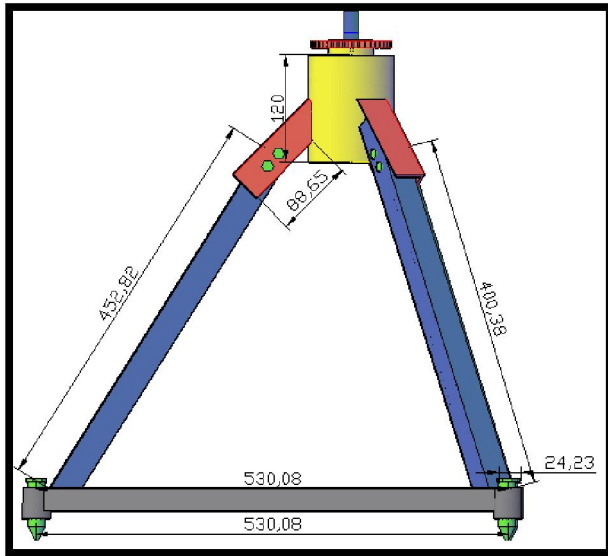
Bentuk desain peluncur roket yang dibuat harus mencukupi sama dengan ukuran roket diatas. Berikut desain muatan roket secara 3D yang sedang dalam proses pembuatan.



**Gambar 3.3:** Rancangan rocket launcher bagian atas



Gambar 3.5: Desain penyangga dan lengan peluncur roket tampak 2D



**Gambar 3.7:** *Desain kaki penyangga peluncur roket*



Dalam proses pembuatan proyek akhir ini diperlukan beberapa jenis material antara lain:

1. Pipa stainless
2. Aluminium
3. Diral
4. As Besi
5. Mur

Sedangkan peralatan yang dibutuhkan antara lain:

1. Mesin bubut
2. Gerinda duduk dan tangan
3. Mesin potong
4. Mesin bor
5. Mesin las
6. Kunci set dll

### **3.2. PEMBUATAN MEKANIK**

Pembuatan sistem dalam proyek akhir ini meliputi desain dan machining mekanik. Untuk membuat alat ini melewati beberapa proses machining. Proses machining untuk alat ini dibutuhkan alat mesin turning (bubut) dan mesin drilling (bor duduk). Implementasi rancangan mekanik pertama dalam perancangan mekanik rocket launcher, kami mengalami beberapa perubahan desain.

#### **3.2.1 MEKANIK ROCKET LAUNCHER BAGIAN BAWAH**

Dalam proses pembuatan alat ini banyak menggunakan mesin turning dikarenakan bentuk mekanik rocket launcher bagian bawah yang berupa silinder. Gambar di bawah ini merupakan rancangan bagian *rocket launcher* bagian bawah.



**Gamba**

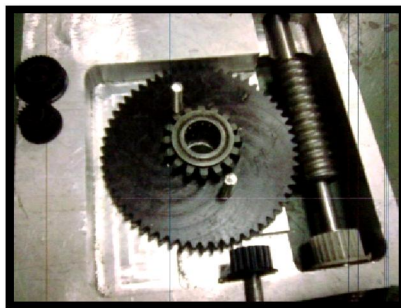
*wah*

### 3.2.2 MEK

**AS**

Dalar menggunakan launcher bagi merupakan rar

as banyak ik rocket bawah ini



**Gambar 3.9:** Foto mekanik rocket launcher bagian atas

Dokumentasi bagian-bagian terpenting dari mekanik rocket launcher adalah sebagai berikut :



**Gambar 3.11:** *Foto bagian atas rocket launcher*



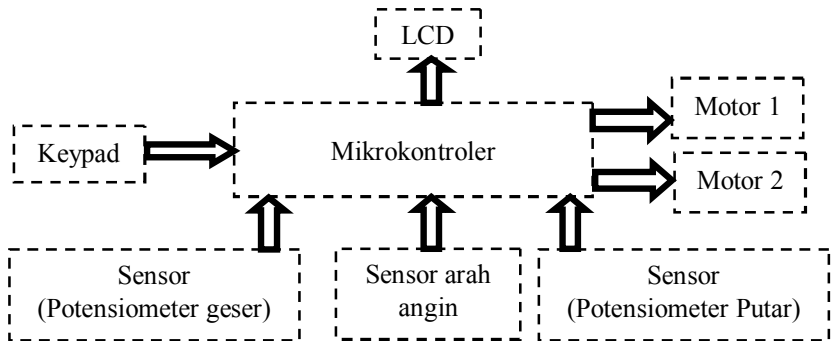
7



**Gambar 3.13:** *Foto rocket launcher bagian bawah*

### 3.3. DESAIN SISTEM

Sistem peluncur roket dengan penggerak elektrik dibuat dengan peluncur roket manual dimodifikasi dengan transmisi gear, motor DC 12 volt, rangkaian mikrokontroler, sensor yang berupa potensiometer, driver, dan supply DC 12 volt 7Ah. sistem peluncur roket dengan penggerak elektrik dibuat dua metode secara manual atau otomatis. Cara manual dibuat agar pengguna dapat mengatur posisi sudut saat roket akan diluncurkan sesuai dengan keinginan dengan menekan tombol yang telah disediakan. Sedangkan cara otomatis menggunakan sensor angin sebagai input untuk nilai parameter untuk mengatur posisi luncur roket yang ideal.



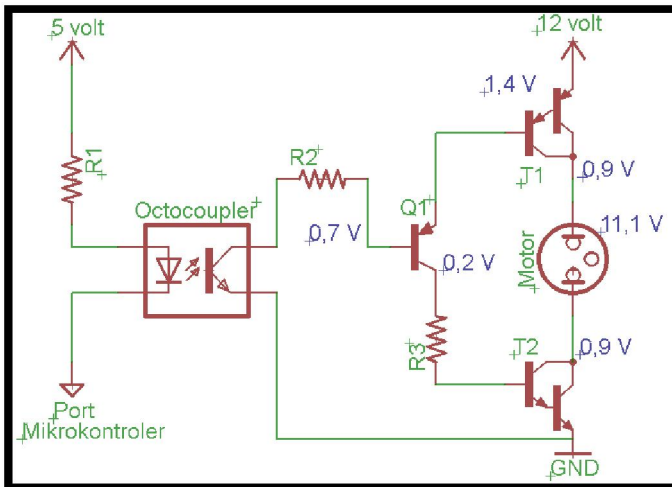
Gambar 3.14: Desain Sistem

### 3.4. DESAIN ELEKTRONIKA

Perencanaan elektronika yang melingkupi bagian dari rangkaian mikrokontroler, supply, sensor derajat dengan memakai potensiometer putar dan potensiometer geser. Bagian-bagian ini terdiri dari ATmega32, LCD, potensiometer geser dan potensiometer putar:

## A. Rangkaian driver

Rangkaian driver untuk menjalankan gear pada peluncur roket memakai rangkaian hibride karena rangkaian ini dapat menggerakkan dua arah yang dibutuhkan untuk naik turunnya peluncur roket agar didapat sudut vertikalnya dan sudut horisontalnya. Transistor yang digunakan adalah transistor darlington karena membutuhkan arus yang besar.



**Gambar 3.15:** Rangkaian hibride satu arah

**Sumber:** (Budiarto, Widodo. (2008). "Panduan Praktikum Mikrokontroler AVR ATmega16")

Untuk Menghitung  $V_{R2}$

$$V_{cc} = V_{cb}(Q1) - V_{R2} - V_{cb}(Q3)$$

$$12 = 1.4 - V_{R2} - 0.7$$

$$V_{R2} = 12 - 2.1$$

$$V_{R2} = 9.9 \text{ v}$$

Untuk Menghitung  $I_{b1}$

$$I_{b1} = I_c / HFE (Q1)$$

$$I_{b1} = 10 / 500$$

$$I_{b1} = 0.02 \text{ A}$$

Untuk menghitung R2

$$I_{b2} = I_{b1} / HFE (Q3)$$

$$I_{b2} = 0.02 / 40$$

$$I_{b2} = 0.0005 \text{ A}$$

$$V_{R2} = R2 \times I_{b2}$$

$$R2 = V_{R2} / I_{b2}$$

$$R2 = 9.9 / 0.0005$$

$$= 19.8 \text{ Kohm}$$

Untuk menghitung R1

$$V_{cc} = V_{cb} (Q1) + V_{cc} (Q2) + V_{R1} + V_{cc} (Q3)$$

$$V_{R1} = V_{cc} - V_{cb} (Q1) - V_{cc} (Q2) - V_{cc} (Q3)$$

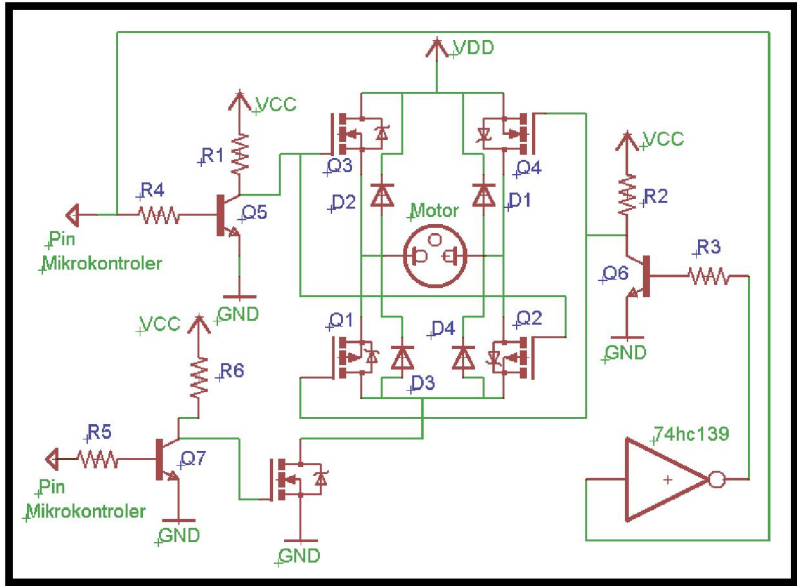
$$V_{R1} = 12 - 0,9 - 0,9 - 0,2$$

$$V_{R1} = 10 \text{ v}$$

$$V_{R1} = I_b \times R1$$

$$R1 = 10 / 0.02$$

$$R1 = 500 \text{ ohm}$$



**Gambar 3.16:** Rangkaian hibride dua arah

**Sumber:** (Budiarto, Widodo. (2008). “Panduan Praktikum Mikrokontroler AVR ATmega16”)

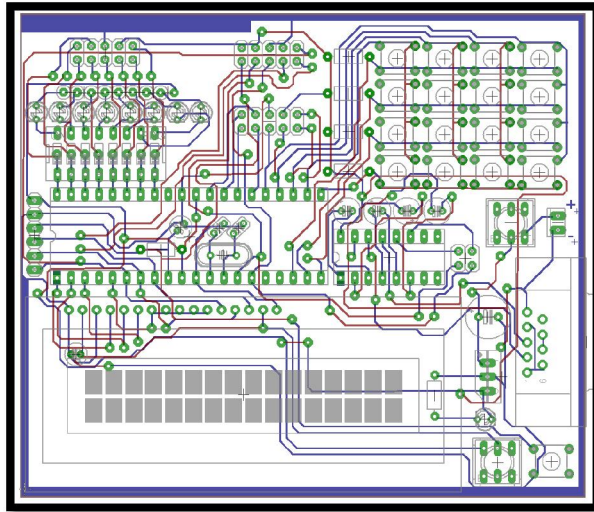
Keterangan gambar:

1. Resistor ( R1, R2, R6 ) = 1kohm.
2. Resistor ( R3, R4, R5 ) = 560ohm.
3. Transistor FET ( Q1, Q3 ) = FIR940.
4. Transistor FET ( Q2, Q4 ) = FIR540.
5. Transistor ( Q5, Q6, Q7 ) = BD139.
6. Vcc = 5volt.
7. Vdd = 12 – 24volt.



## B. ATmega32

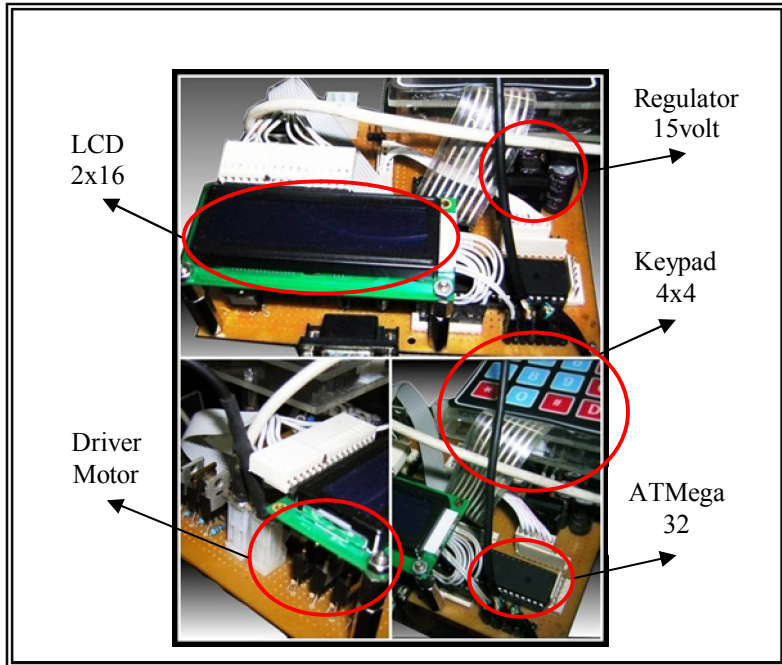
Mikrokontroler yang digunakan adalah tipe ATmega32. Mikrokontroler ini berfungsi untuk tegangan dari sensor-sensor, mengontrol motor input keypad, save dan load kemudian ditampilkan pada LCD.



**Gambar 3.17:** *Board PCB Rangkaian ATmega32*

### 3.5. PEMBUATAN SISTEM ELEKTRONIK

Rangkaian Mikrokontroler yang digunakan adalah tipe ATmega32 pada gambar di bawah ini terdiri dari 4 bagian yaitu yang pertama input sensor angin yang dapat menginformasikan posisi angin, lalu yang kedua output driver motor yang dapat dihubungkan ke motor untuk dapat diambil posisi rocket launcher mengikuti arah angin, lalu yang ketiga bagian komunikasi serial yaitu yang dapat mengirim atau menerima data, sedangkan yang terakhir yaitu LCD yang berfungsi menampilkan informasi posisi arah angin dan kecepatan angin dari sensor angin.

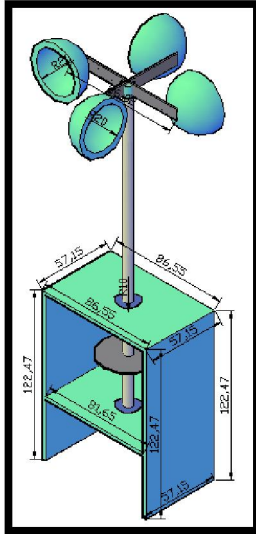


**Gambar 3.18:** *Foto rangkaian sistem mikrokontroler dan rangkaian driver motor*

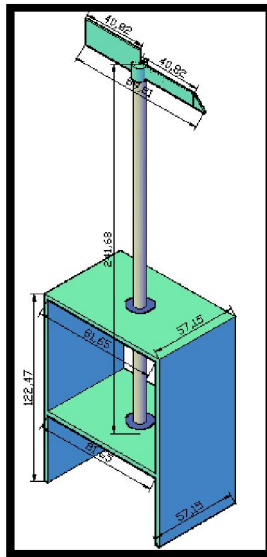
### 3.6. DESAIN SENSOR ANGIN

Spesifikasi teknis rangkaian sensor sistem penentuan derajat arah mata angin terdiri atas beberapa komponen sebagai berikut.

- A. LED infra merah (Photodioda) berfungsi sebagai pemancar sinar infra merah.
- B. Phototransistor berfungsi sebagai penerima sinar infra merah.
- C. SN74HC132 berfungsi sebagai pembangkit pulsa yang diumpankan ke LED infra merah.
- D. Komponen pendukung lainnya yaitu resistor dan dioda.



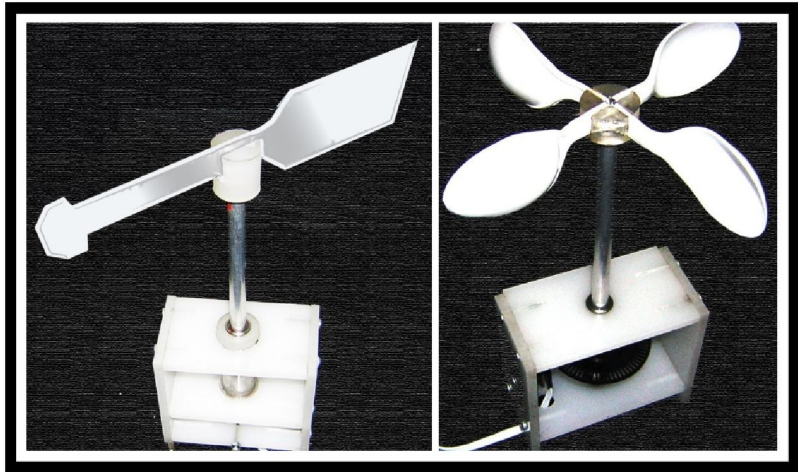
Gambar 3.19: Mekanik Sensor Kecepatan Angin



Gambar 3.20: Mekanik Sensor Kecepatan Angin

### 3.7. PEMBUATAN SENSOR ANGIN

Pada tahap pembuatan sensor angin dilakukan dengan mengambil blok yang diperlukan yaitu rotary encoder, potensiometer dan flag yang mengarahkan dan menunjukkan posisi angin.



**Gambar 3.21:** *Foto mekanik sensor angin*

# **BAB IV**

## **PENGUJIAN DAN ANALISA SISTEM ROKET LAUNCHER**

### **4.1. UJI COBA PERANGKAT PELUNCUR ROKET**

Pada bab ini akan dilakukan pengujian dan analisa terhadap alat dan algoritma yang telah dirancang dan dibuat pada bab sebelumnya. Untuk tahap-tahap pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian Kontrol PID.
2. Pengujian Driver Motor DC.
3. Pengujian Data ADC.
4. Pengujian Sensor Wind Vane Direction.
5. Pengujian Sensor Anemometer.

#### **4.1.1 Pengujian Kontrol PID.**

##### **1. Tujuan.**

Melakukan pengujian pada PID kontrol posisi. Kontrol ini pada dasarnya adalah kontrol kecepatan putar pada motor DC. Hal ini berkaitan dengan prinsip dasar pembangkit gerakan yang dilakukan oleh motor, yaitu bila diberikan tegangan pada terminalnya maka poros motor atau rotor akan berputar. Jadi mengontrol berapa besar tegangan yang dikenakan ke motor pada dasarnya adalah berapa kecepatan putar poros motor yang dikehendaki dalam rpm (revolution per minute).

##### **2. Peralatan.**

- Software Matlab
- PC (personal Computer)

### 3. Persiapan.

Komponen yang akan dikontrol terdiri dari motor dan sebagai deteksi posisi menggunakan sensor potensio linear yang dapat memberikan informasi posisi motor saat itu.

Adapun nilai-nilai yang terdapat pada plant tersebut meliputi:

- Moment of inertia of the rotor ( $J_{eff}$ )=0.0001 kg/m<sup>2</sup>
- Damping ratio of the mechanical system ( $b$ )=0.005 Nms
- Electromotive force resistance(  $K=K_e=K_t$ )=0.04 Nm/Amp
- Electric resistance( $R$ )=4 ohm
- Electric inductance( $L$ )=0.1 H
- Input (V): source voltage
- Output ( $\theta$ ):position of shaft

Kita menginginkan perubahan posisi motor dapat tepat. Error stady state motor menjadi nol. kita juga ingin dapat mengendalikan motor yang mendapat pengaruh dari luar dengan error stady state mendekati 0. Jika kita simulasi dengan inputan ( $R$ ) sebagai input fungsi step diharapkan keluaran motor akan sesuai dengan:

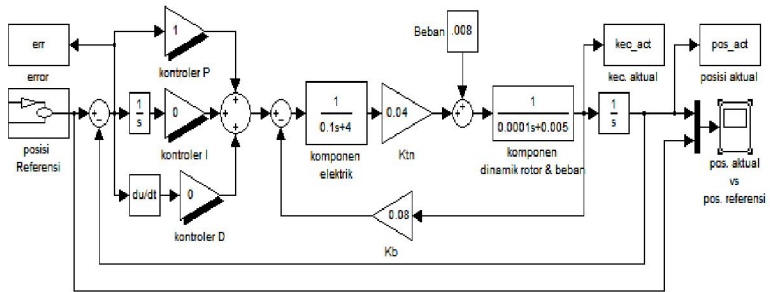
Settling time less than 40 milliseconds

Overshot less than 16%

No steady-state error

No steady-state error walau mendapat gangguan dari luar sistem.

#### 4. Blok Diagram.

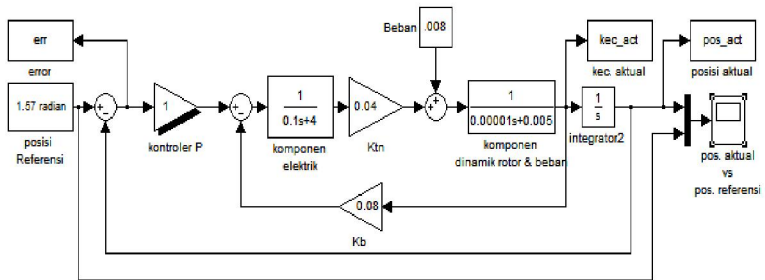


Gambar 4.1: Blok Diagram kontrol PID

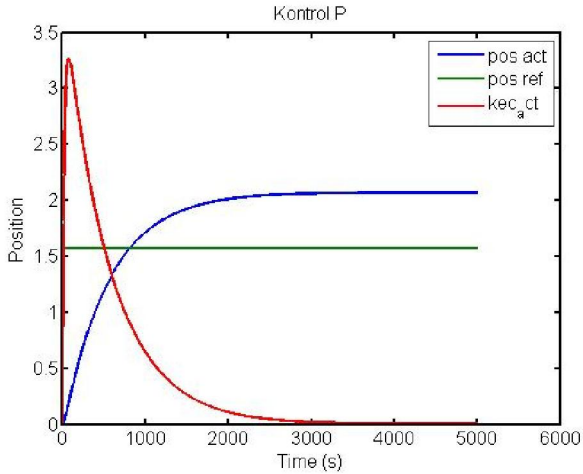
#### 5. Pengujian.

### KONTROL PROPORTIONAL

Yang pertama menggunakan konstanta kontrol proporsional ( $k_p$ )



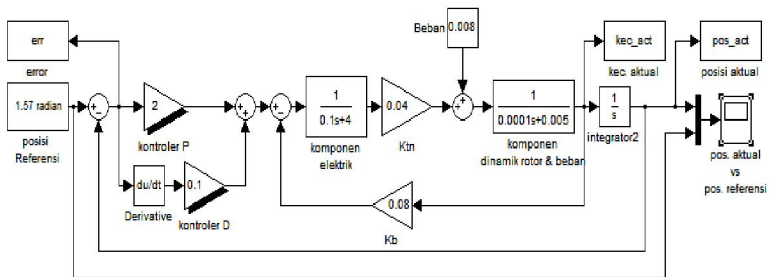
Gambar 4.2: Blok diagram kontrol P



**Gambar 4.3:** Kontrol P dengan respon sistem terhadap gangguan

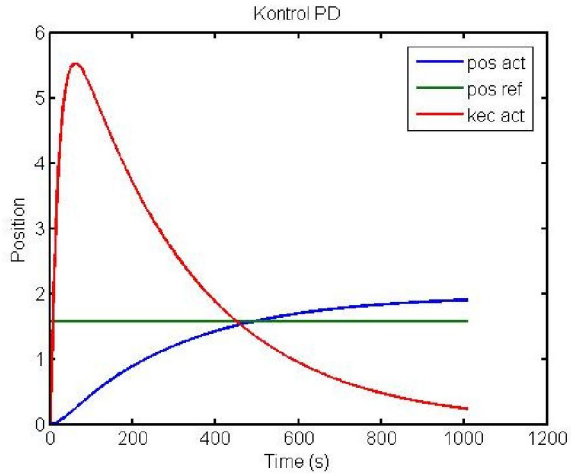
## KONTROL PROPOSIONAL DERIVATIF

Dari Gambar dapat terlihat stady state masih error tapi setting time bagus dengan overshoot masih lebar. Sekarang menggunakan kontrol PID. untuk langkah ini dapat dilihat pada perintah di bawah:



**Gambar 4.4:** Blok diagram kontrol PD

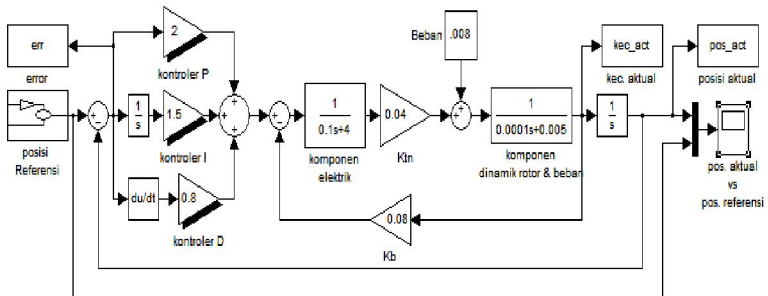




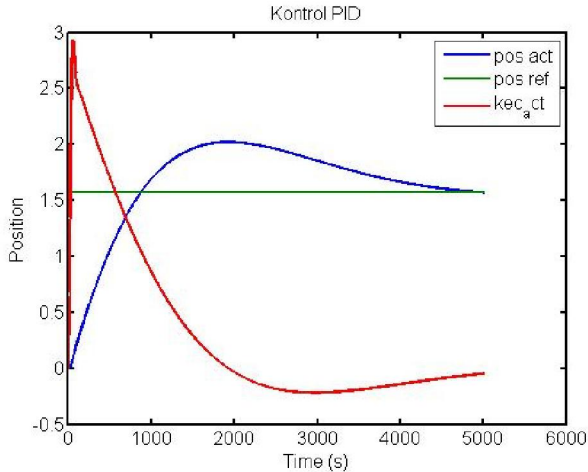
**Gambar 4.5:** Kontrol PD dengan respon sistem terhadap gangguan

### KONTROL PID PROPOSIONAL INTEGRAL DERIVATIF

Kontrol PD pada sistem dari gambar di atas belum diperoleh hasil sesuai dengan ketentuan awal. Dengan membuka m file dengan nilai  $k_p=2$ ,  $k_i=1.5$ , dan  $k_d=0.8$ .



**Gambar 4.6:** Blok diagram kontrol PID



**Gambar 4.7:** Kontrol PID dengan respon sistem terhadap gangguan

## 6. Hasil dan Analisa.

Dari hasil simulasi terlihat bahwa pada saat hanya pengendali P dan PI saja yang digunakan, respon sistem beresonansi dengan lonjakan cukup besar mencapai nilai 2.7. Untuk menghilangkan lonjakan yang terjadi maka ditambahkan pengendali D dengan beberapa nilai parameter pengendali yang bervariasi. Dari beberapa variasi nilai parameter pengendali PID, respon sistem mampu memberikan respon yang baik dan mampu mengikuti perubahan input yang diberikan. Respon sistem terbaik diperoleh untuk nilai  $K_p=2$ ,  $K_i=1.5$ , dan  $K_d=0.8$ , dimana respon sistem tidak terjadi lonjakan dan mampu mencapai keadaan tunak pada  $t=5$  detik (Gambar 51).

### 4.1.2 Pengujian Driver Motor DC

#### 1. Tujuan.

Mengetahui kerja dari driver motor yang digunakan pada peluncur roket.

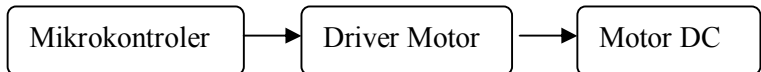
## 2. Peralatan.

- Motor DC.
- Minimum Sistem AVR Mikrokontroler.
- Power Supply.
- Stopwatch.

## 3. Persiapan.

- Memasang rangkaian Minimum Sistem AVR dengan Motor DC sesuai dengan diagram di bawah.
- Melakukan inisialisasi pada PORT mikrokontroler.
- Mendownload program dari CodeVision AVR ke mikrokontroler.
- Mengukur sinkronisasi putaran motor DC.

## 4. Blok Diagram.



**Gambar 4.8:** *Diagram pengujian motor DC*

## 5. Pengujian.

Pengujian driver motor ini dilakukan untuk memastikan bahwa motor DC sebagai penggerak utama pada peluncur roket telah berkerja dengan baik. Pengujian ini dilakukan dengan mengerakkan gear pada *sendi utama* dan *lengan seluncur* pada peluncur roket. Berikut potongan program dalam melakukan uji coba.

```
#include <mega32.h>
#include <delay.h>
#define pwmka OCR1A
#define pwmki OCR1B
#define on 1
#define off 0
```

```

// Declare your global variables here
void maju(unsigned char kanan,unsigned char
kiri)
{
    PORTA.0=on;
    PORTA.1=on;
    pwmka=kanan;
    pwmki=kiri;
}
void mundur(unsigned char kanan,unsigned
char kiri)
{
    PORTA.0=off;
    PORTA.1=off;
    pwmka=kanan;
    pwmki=kiri;
}
while (1)
    {
        maju(255,255); //konstanta PWM pada
rocket launcher
        delay_ms(1000);
        mundur() ;//Perintah maju atau
mundur
        delay_ms(1000);};}

```

## 6. Hasil dan Analisa.

Pada pengujian dengan menggunakan program di atas saat setiap motor DC diberikan nilai konstanta PWM yang sama ternyata hasil putaran yang dihasilkan berbeda-beda. Hal ini disebabkan oleh beberapa faktor, diantaranya:

1. Gear peluncur rocket tidak presisi dikarenakan salah dalam perhitungan perbandingan gear dan ulir.
2. Stuktur peluncur roket yang tidak robus.
3. Elastisitas belt pada *sendi utama* pada peluncur roket terlalu kencang.
4. Motor DC yang digunakan adalah motor bekas pakai.

Untuk mengatasi hal tersebut, tiap motor diberikan konstanta PWM yang berbeda-beda agar menghasilkan putaran yang diharapkan. Dengan mengubah nilai OCR mulai dari 0 hingga 255, putaran motor dapat dikendalikan agar menuju target yang diinginkan.

Dalam uji coba yang dilakukan motor DC di program untuk berputar dengan kecepatan penuh, yaitu dengan memberikan konstanta PWM 255. Motor berputar CW selama 1 detik kemudian berganti dengan putaran CCW selama 1 detik, dan terus berulang. Dengan program ini driver motor dapat berjalan dengan baik pada arah CW dan CCW.

### **4.1.3 Pengujian Data ADC**

#### **1. Tujuan.**

Melakukan pengujian pada sensor posisi dengan menggunakan sensor potensio linear yang dihubungkan menggunakan pin ADC mikrokontroler.

#### **2. Peralatan.**

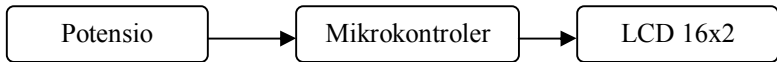
- Potensio Linear Multiturn Wirewound 534: 10.
- Minimum Sistem AVR Mikrokontroler.
- LCD 16x2.
- Power Supply.

#### **3. Persiapan.**

- Membuat rangkaian pembagi tegangan karena pada prinsipnya output dari sensor adalah variabel resistor.
- Memasang rangkaian Minimum Sistem AVR dengan Sensor potensio linear sesuai dengan diagram di bawah.
- Melakukan perhitungan pada adc sensor agar mendapatkan hasil yang mudah ditentukan posisinya.

- Melakukan tes dan mengambil data ADC sensor menggunakan LCD 16x2.

#### 4. Blok Diagram.



**Gambar 4.9:** Blok diagram pengujian Sensor potensio linear

#### 5. Pengujian.

Pada pengujian sensor potensio linear menggunakan pengolahan data ADC yang dari pengukuran dengan pembagi tegangan. Sensor potensio linear ini didalamnya memiliki putaran hingga 10 kali, output sensor ini adalah resistor variabel yang bernilai resistor 20k untuk putaran maksimal. Permasalahannya adalah di dalam instaliasi *rocket launcher* dipasang gear dengan perbandingan 1:4.25 yaitu Gear1=20 dan Gear2=85. Dengan cara ini dapat menghasilkan putaran 1 kali *rocket launcher* berputar sama dengan 4.25 kali putaran potensio untuk menghasilkan gerak putar 360 derajat, maka dibuatlah perhitungan untuk mencapai hasil tersebut. Nilai potensio 20k dengan putaran maksimal 10 kali hal ini bisa didapat perhitungan 1 kali putar bernilai 2k, maka  $2000 \text{ ohm} \times 4.25 = 8500 \text{ ohm}$ . Sehingga dapat disimpulkan bahwa untuk setiap perubahan 1 derajat menghasilkan 23.61 ohm.



**Gambar 4.10:** Gambar Potensio Linear

Input ADC pada mikrokontroler dihubungkan ke sebuah 8 channel Analog multiplexer yang digunakan untuk single ended input channel. Jika sinyal input dihubungkan ke masukan ADC dan 1 jalur lagi terhubung ke ground, disebut single ended input. Jika input ADC terhubung ke 2 buah input ADC disebut sebagai differential input, yang dapat dikombinasikan sebanyak 16 kombinasi. Tegangan referensi ADC dapat dipilih antara lain pada pin AREF, pin AVCC atau menggunakan tegangan referensi internal sebesar 2.56 Volt. Agar fitur ADC mikrokontroler dapat digunakan maka ADEN (ADC Enable, dalam I/O register ADCSRA) harus diberi nilai 1.

$$ADC = \frac{V_{in} \times 1024}{V_{ref}} \dots\dots\dots (4.1)$$

Langkah selanjutnya adalah menghitung data ADC agar didapat nilai yang berbeda-beda tiap posisi dengan hasil data tersebut maka kita bisa menentukan derajat posisi *rocket launcher*.

$$\begin{aligned} V_{in} &= V / 10 \text{ putaran potensio} \\ &= 5 / 10 \\ &= 0.5 \text{ v} \end{aligned}$$

Maka didapat hasil 0.5 Volt per 1 kali putaran, yang dibutuhkan *rocket launcher* dalam 1 kali putaran adalah 4.25 putaran nilai potensio.

$$\begin{aligned} V_{in} &= 4.25 \times 0.5\text{v} \\ &= 2.125\text{v} \end{aligned}$$

Hasil  $V_{in} = 2.125\text{v} / 360 \text{ derajat} = 0.0059 \text{ per } 1 \text{ derajat}$ .

Cara mendapatkan nilai ADC yaitu:

$$\begin{aligned} ADC &= 1024 / 10 \text{ putaran potensio} \\ &= 102.4 \end{aligned}$$

Data 102.4 per 1 kali putaran, sehingga dapat dihitung  $102.4 \times 4.25 = 435.2$ . dari hasil tersebut dibagi 360 derajat mendapatkan 1.208 per 1

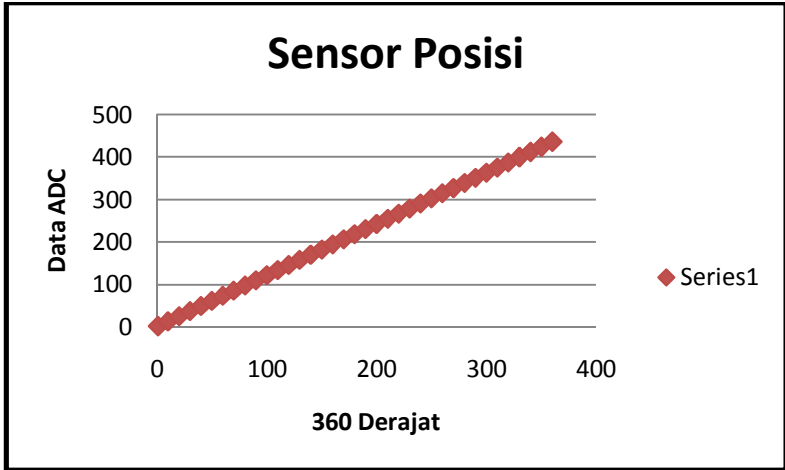
derajat. Data hasil sensor potensio linear dimasukkan tabel yaitu sebagai berikut:

**Tabel 4.1:** *Data Sensor Posisi*

Posisi	Resistor	Vin	Data ADC
1	23.61 $\Omega$	0.0059v	1.208
10	236.1 $\Omega$	0.059v	12.08
20	472.2 $\Omega$	0.118v	24.16
30	708.3 $\Omega$	0.177v	36.24
40	944.4 $\Omega$	0.236v	48.32
50	1.180k $\Omega$	0.295v	60.4
60	1.416k $\Omega$	0.354v	72.48
70	1.652k $\Omega$	0.413v	84.56
80	1.888k $\Omega$	0.472v	96.64
90	2.124k $\Omega$	0.531v	108.72
100	2.361k $\Omega$	0.59v	120.8
110	2.597k $\Omega$	0.649v	132.88

Tabel hasil data percobaan sensor (posisi) potensiometer dapat dilihat lebih lengkapnya di lampiran.





**Gambar 4.11:** *Grafik Sensor Posisi*

Langkah selanjutnya adalah mengirim data untuk ditampilkan ke LCD agar dapat didefinisikan ke program selanjutnya. Berikut ini adalah potongan program dalam melakukan uji coba.

```
#include <mega32.h>
#include <delay.h>
#include <math.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC
```

```

input voltage
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
while (1)
{
temp=read_adc(0);
vref=((float)temp*0.00488); //baca dari
ADC dikalikan (5v/1024)
set=(((int)temp-540)*2.44); //baca ADC di
set nilai yg diinginkan
lcd_gotoxy(1,0);
lcd_putsf("==Voltmeter==");
lcd_gotoxy(0,1);
sprintf(lcd_buffer,"Volt=%3f
set:%d",vref,set);
lcd_puts(lcd_buffer);
delay_ms(100);
};
}

```

## 6. Hasil dan Analisa.

Nilai Vref yang dihasilkan dari sensor posisi diterima oleh mikrokontroler dengan cara input potensio dimasukkan pin ADC mikrokontroler dan menghasilkan keluaran tegangan rata-rata. Data dari pengujian sensor Potensio Linear menghasilkan nilai berupa bilangan pecahan hal ini akan menghasilkan nilai error sekitar:

$$\begin{aligned}
 \text{Error} &= (\text{Vin perhitungan} - \text{Vin percobaan}) \times 100\% \\
 &= (2.125 - 2.124) \times 100\% \\
 &= 0.1\%
 \end{aligned}$$

Dari hasil di atas diketahui nilai error sebesar 0.1%. hal ini membuat nilai aktual sulit mencapai target sebenarnya.

#### 4.1.4 Pengujian Sensor Wind Vane Direction.

##### 1. Tujuan.

Melakukan pengujian pada sensor arah angin dengan menggunakan potensio linear yang dihubungkan menggunakan pin ADC mikrokontroler.

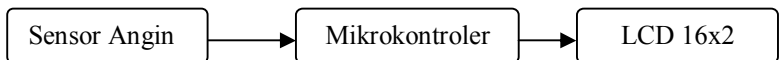
##### 2. Peralatan.

- Sensor Angin
- Minimum Sistem AVR Mikrokontroler.
- LCD 16x2.
- Power Supply.

##### 3. Persiapan.

- Membuat rangkaian pembagi tegangan karena pada prinsipnya output dari sensor adalah variabel resistor.
- Memasang rangkaian Minimum Sistem AVR dengan sensor angin sesuai dengan diagram di bawah.
- Melakukan perhitungan pada ADC sensor agar mendapatkan hasil yang mudah ditentukan arahnya.
- Melakukan tes dan mengambil data ADC sensor menggunakan LCD 16x2.

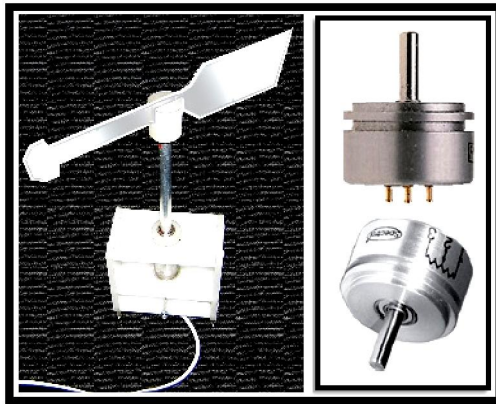
##### 4. Blok Diagram.



**Gambar 4.12:** Blok diagram pengujian Sensor Arah Angin

## 5. Pengujian.

Pada pengujian sensor angin menggunakan pengolahan data ADC yang dari pengukuran dengan pembagi tegangan. Sensor angin ini didalamnya terdapat serangkaian potensio linear *continuous* yang jika diputar 1 kali putaran penuh nilai outputnya akan kembali ke nilai awal, output sensor ini adalah resistor variabel yang difungsikan sebagai pembagi tegangan agar dapat diambil data ke ADC mikrokontroler.



**Gambar 4.13:** *Gambar sensor Arah Angin*

Langkah selanjutnya adalah menghitung tegangan ADC  $V_{ref}$  agar didapat nilai tegangan yang berbeda-beda tiap arah dengan hasil data tersebut maka kita bisa menentukan masing-masing arah.

$$\begin{aligned}V_{in} &= V_{ref} / 360 \\ &= 5 / 360 \\ &= 0.013V\end{aligned}$$

Maka didapat hasil 5 Volt per 1 kali putaran penuh, yang dibutuhkan sensor posisi angin didalam menentukan posisi derajat arah angin. Dari hasil di atas dapat disimpulkan bahwa nilai potensio 0.01 Volt per 1 derajat.

$$ADC = \frac{V_{in} \times 1024}{V_{ref}} \dots\dots\dots (4.2)$$

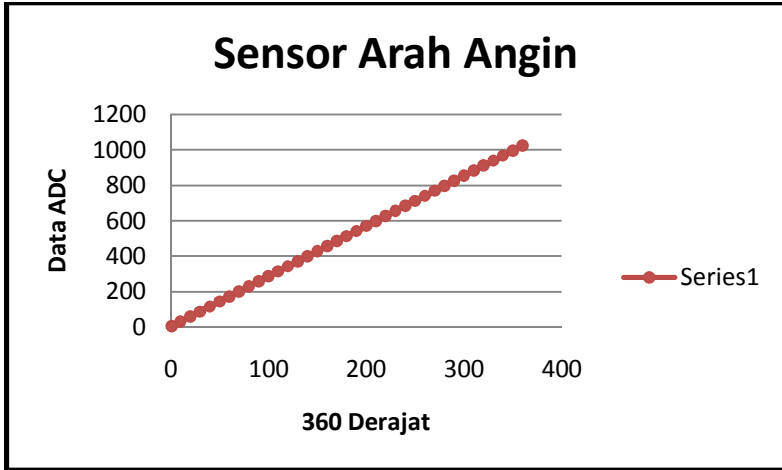
Misal nilai vin 0.138V maka nilai ADC = (0.138V / 5V) x 1024  
= 28.44

Data hasil sensor angin adalah sebagai berikut.

**Tabel 4.2:** *Data Sensor Arah Angin*

Posisi	Resistor	Vin	Data ADC
1	55.556Ω	0.013v	2.844
10	555.55Ω	0.138v	28.44
20	1.111kΩ	0.276v	56.88
30	1.667kΩ	0.414v	85.32
40	2.223kΩ	0.552v	113.76
50	2.779kΩ	0.690v	142.2
60	3.335kΩ	0.828v	170.64
70	3.891kΩ	0.966v	199.08
80	4.447kΩ	1.104v	227.52
90	5.003kΩ	1.242v	255.96
100	5.559kΩ	1.380v	284.4
110	6.115kΩ	1.518v	312.84
120	6.671kΩ	1.656v	341.28

Tabel hasil data percobaan sensor arah angin dapat dilihat lebih lengkapnya di lampiran.



*Gambar 4.14: Grafik sesnsor arah angin*

Langkah selanjutnya adalah mengirim data untuk ditampilkan ke LCD agar dapat didefinisikan ke program selanjutnya. Berikut ini adalah potongan program dalam melakukan uji coba.

```
#include <mega32.h>
#include <stdio.h>
#include <delay.h>
#include <math.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
```

```

// Delay needed for the stabilization of the ADC
input voltage
delay_us(10);
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
while (1)
{
// Place your code here
temp=read_adc(0);
vin=((float)temp*0.00488); //baca dari ADC
dikalikan (5/1024)
lcd_gotoxy(1,0);
lcd_putsf("==Voltmeter==");
lcd_gotoxy(0,1);
sprintf(lcd_buffer,"Volt=%0.3f",vin);
lcd_puts(lcd_buffer);
delay_ms(100);
};
}

```

## 6. Hasil dan Analisa.

Nilai  $V_{ref}$  yang dihasilkan dari sensor arah angin diterima oleh mikrokontroler dengan cara input potensio dimasukkan pin ADC mikrokontroler dan menghasilkan keluaran tegangan rata-rata. Data dari pengujian sensor Potensio Linear *continuous* menghasilkan nilai berupa bilangan pecahan hal ini akan menghasilkan nilai error sekitar:

$$\begin{aligned}
 \text{Error} &= \frac{(V_{in \text{ perhitungan}} - V_{in \text{ percobaan}})}{V_{ref}} \times 100\% \\
 &= ((5v - 4.968v) / 5v) \times 100\% \\
 &= 0.64\%
 \end{aligned}$$

Dari hasil di atas diketahui nilai error sebesar 0.64%. hal ini membuat nilai aktual sulit mencapai target sebenarnya.

#### 4.1.5 Pengujian Sensor Anemometer.

##### 1. Tujuan.

Melakukan pengujian pada sensor kecepatan angin dengan menggunakan sensor anemometer yang dihubungkan menggunakan pin B (T0) mikrokontroler yang dapat menerima eksternal clock dari luar.

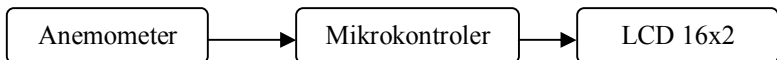
##### 2. Peralatan.

- Sensor Anemometer.
- Minimum Sistem AVR Mikrokontroler.
- LCD 16x2.
- Power Supply.

##### 3. Persiapan.

- Sensor di pasang rotary encoder yang berjumlah 50 lubang dan dihubungkan dengan LED photodiode agar dapat dibaca selama 1 detik kemudian dimasukan buffer SN74HC132 ke mikrokontroler agar output sensor mendapatkan nilai pulsa.
- Melakukan perhitungan counter PWM pada sensor yang kemudian dikalikan nilai konstanta pada sensor agar mendapatkan hasil yang bisa ditentukan tempuh kecepatan angin.
- Melakukan tes dan mengambil data counter PWM sensor menggunakan LCD 16x2.

##### 4. Blok Diagram.

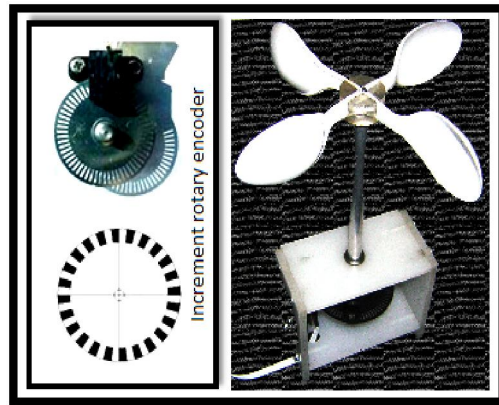


**Gambar 4.15:** Blok Diagram pengujian Sensor Anemometer



## 5. Pengujian.

Pada pengujian Sensor Anemometer menggunakan pengolahan data pulsa yang dari pengukuran dengan counter/timer. Sensor Anemometer ini didalamnya terdapat LED dan photodiode yang dipasang ke rotary encoder, output sensor ini adalah pulsa yang dicounter dengan sampling 1 detik tujuannya adalah pulsa eksternal dari mikrokontroler dapat dibaca dan dapat diambil data kemudian dikalikan dengan konstanta anemometer. Dari hasil tersebut data sensor bisa didefinisikan.



Gambar 4.16: Gambar sensor Anemometer

Langkah selanjutnya adalah menghitung pulsa yang dihasilkan Sensor Anemometer di pin B (T0) pada mikrokontroler agar didapat nilai pulsa yang berbeda-beda tiap kecepatan dengan hasil data tersebut maka kita bisa menentukan kecepatan angin sebenarnya.

N merupakan banyaknya pulsa dan V merupakan kecepatan Anemometer

$$\begin{aligned} \text{Jumlah lubang} &= 70 \text{ perputaran} \\ \text{Jarak(cm)} &= \text{read hole}/70 \\ \text{Keliling lingkaran} &= 2\pi r \end{aligned}$$

Jari-jari lingkaran = 3 cm  
 Diameter lingkaran = 6 cm  
 Kecepatan (km/H) =  $V \times (3600/1000)$

$$V = \frac{N}{70} \times (2\pi r) \dots\dots\dots(4.3)$$

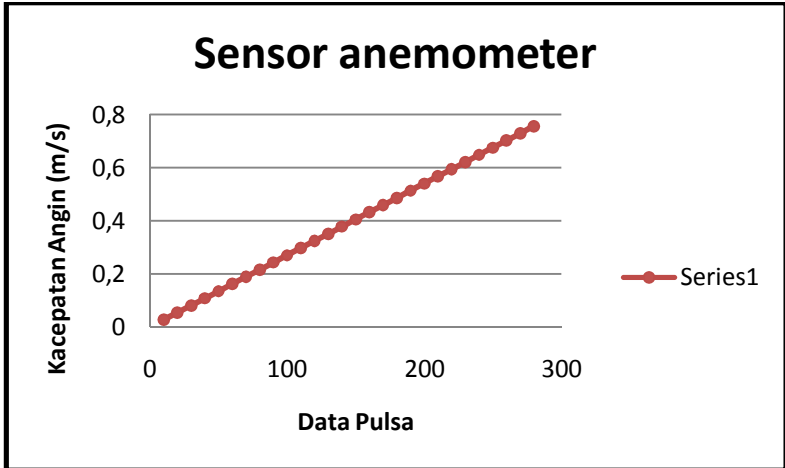
Misal nilai pulsa yang didapat  $N = 140$ , maka  $V = 140/70 \times (0.1884)$   
 Sehingga Nilai  $V = 0.337$  m/s

Data hasil Sensor Anemometer adalah sebagai berikut.

**Tabel 4.3:** *Data Sensor Anemometer*

N (pulsa)	N / 70	V (m/s)	V (km/H)
10	0.142	0.026	0.093
20	0.285	0.053	0.19
30	0.428	0.08	0.287
40	0.571	0.107	0.384
50	0.714	0.134	0.481
60	0.857	0.161	0.578
70	1	0.188	0.675
80	1.143	0.215	0.772
90	1.286	0.242	0.869
100	1.429	0.269	0.966
110	1.572	0.296	1.063
120	1.715	0.323	1.16

Tabel hasil data percobaan sensor Anemometer dapat dilihat lebih lengkapnya di lampiran.



**Gambar 4.17:** *Grafik sensor kecepatan angin*

Langkah selanjutnya adalah mengirim data untuk ditampilkan ke LCD agar dapat didefinisikan ke program selanjutnya. Berikut ini adalah potongan program dalam melakukan uji coba.

```
#include <mega32.h>
#include <stdio.h>
#include <delay.h>

unsigned char lcd_buffer[30];
int count, rotary;

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
```

```

TCNT1H=0xD5; // reinitialize Timer 1 value
TCNT1L=0xD0; // menghasilkan sampling waktu
ldetik

count=TCNT0; // hitung pulsa di pinB (T0)
TCNT0=0x00; // nilai TCNT direset 0
}
void kecp_angin()
{
    rotary=((float)Count/70)*1.884; //
    dikalikan keliling lingkaran
    lcd_gotoxy(2,0);
    lcd_putsf("== DATA ==");
    lcd_gotoxy(2,1);
    sprintf(lcd_buffer," %0.3f m/s", rotary);
    lcd_puts(lcd_buffer);
}

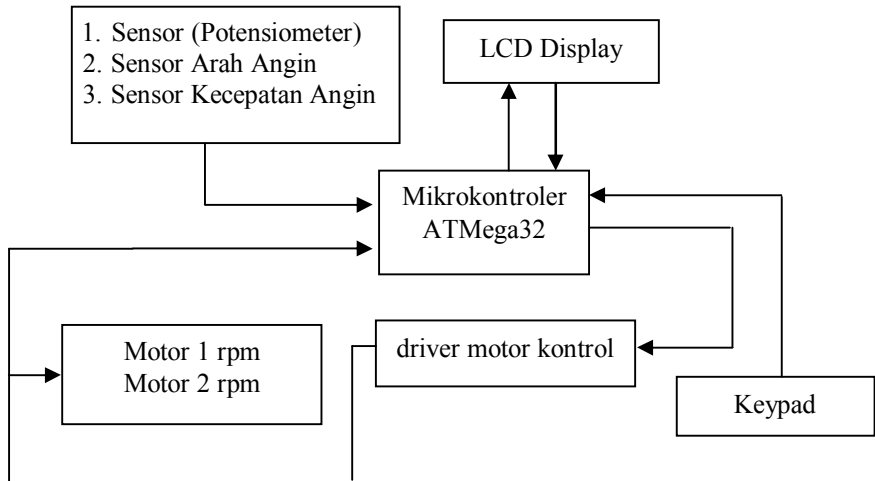
```

## 6. Hasil dan Analisa.

Data sensor sebagai input yang melalui pin B (T0) mikrokontroler hasil yang didapat adalah pulsa yang dicounter selama 1 detik. Data dari pengujian Sensor Anemometer hasil nilai harus dikalikan dengan konstanta anemometer : lubang rotary encoder 70, keliling lingkaran 0.1884m agar mendapatkan nilai kecepatan angin sebenarnya maka hasil pulsa yang didapat selama sampling 1 detik dibagi 70 dan dikalikan 0.1884m .

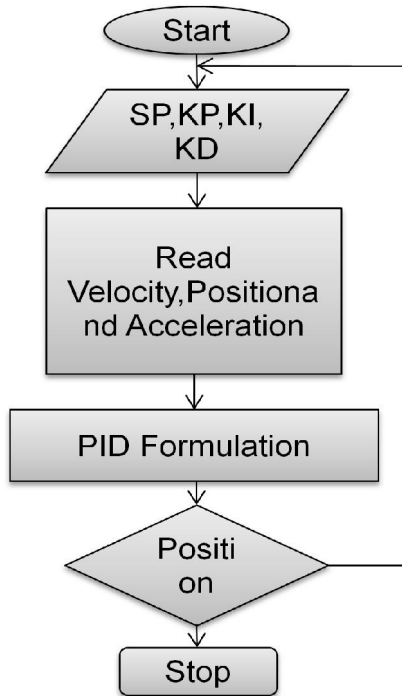
### 4.2. UJI COBA KESELURUHAN SISTEM PELUNCUR ROKET

Pada desain blok diagram keseluruhan dari tugas akhir terdiri dua blok utama, yaitu blok kontroler dan blok sistem driver motor kontrol. Dimana nantinya mulai dari mekanik atas peluncur roket sampai bawah peluncur roket akan dikontrol oleh mikrokontroler ATmega32 dan ditampilkan melalui LCD. Berikut dibawah ini merupakan gambar blok diagram keseluruhan dari tugas akhir.



**Gambar 4.18:** Blok diagram keseluruhan tugas akhir

Spesifik blok diagram dari tugas akhir, menunjukkan bahwa input dari pergerakan peluncur roket adalah sensor arah angin dan manual keypad. Untuk gerak peluncur roket yang dideteksi dari sensor angin hanya digunakan sensor posisi arah angin saja untuk mengetahui arah angin berhembus namun untuk sensor kecepatan angin hanya sebagai informasi mengetahui kecepatan angin berhembus. Begitu angin dideteksi oleh sensor angin maka peluncur roket akan mengarah ke posisi yang tepat dimana angin berhembus. Pada halaman selanjutnya adalah gambar dari flowchart program tugas akhir.



**Gambar 4.19:** *FlowChart – PID Kontrol*

Pada blok flowchart PID kontrol input set point berupa sensor arah angin. Dimana posisi angin yang berasal dari angin disekitar peluncur roket akan dideteksi oleh sensor posisi angin yang nantinya digunakan sebagai nilai posisi target peluncur roket. Sedangkan untuk kecepatan angin dideteksi dengan sensor anemometer. Anemometer ini berfungsi sebagai media informasi kecepatan angin saja. Data dari sensor anemometer tidak berpengaruh terhadap kontrol peluncur roket, sedangkan data dari sensor posisi angin sangat berpengaruh terhadap mekanik pergerakan peluncur roket. Data berupa data analog, Output yang dihasilkan dari kontroler PID yaitu banyaknya tegangan yang melewati mosfet yang digunakan sebagai switching untuk mengkondisikan tegangan yg masuk ke motor agar bekerja cepat

sehingga mengeluarkan tegangan output yang optimal ke driver motor. Data-data dari sensor potensiometer yang dipasang di mekanik peluncur roket berupa data analog. Berikut ini adalah potongan program dalam melakukan uji coba.

```
#include <mega32.h>
#include <delay.h>
#include <math.h>

unsigned char xcount,lpwm,rpwm;
int Error, dError, iError, ErrorSebelumnya,KP,
KI, KD;
float pwm;
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
xcount++; // xcount=xcount+1
    if(xcount<=lpwm)ENKI=0; // EnKi=1 jika
xcount <= lpwm
    else ENKI=1; // EnKi=0 jika xcount >
lpwm
    if(xcount<=rpwm)ENKA=0; // EnKa=1 jika
xcount <= rpwm
    else ENKA=1; // EnKa=0 jika xcount >
rpwm
}
void maju() { DIRKA=0; PORTKA=1; }
void mundur() { DIRKA=1;PORTKA=1; }
void rem() { rpwm=0;ENKA=0;PORTKA=0; }
void main(void)
{
KP = 20; // Nilai KP diset 20
KI = 0; // Nilai KI diset 0
KD = 1; // Nilai KD diset 1

while (1)
```

```

    {
        target=read_adc(0); actual=read_adc(3);
        if (set==target) { rem(); }
        if (set<target)
            {
                Error = target - set;
                dError = Error -
ErrorSebelumnya;
                iError = iError + (float)(Error
+ ErrorSebelumnya)* 0.005;
                ErrorSebelumnya = Error;
                pwm = pwm + (KP * Error) + (KI *
iError) + (KD * dError);
                rpwm = (int)pwm;
            if (pwm>1023)
                { maju(); pwm =1023; }
            if (pwm<0)
                { maju(); pwm =0; }
            }
        if (set>target)
            {
                Error = target - set;
                dError = Error - ErrorSebelumnya;
                iError = iError + (float)(Error +
ErrorSebelumnya)* 0.005; ErrorSebelumnya =
Error;

                pwm = pwm + (KP * Error) + (KI * iError) + (KD *
dError);
                rpwm = (int)pwm;
            if (pwm>1023)
                { mundur(); pwm =1023;}
            if (pwm<0)
                { mundur(); pwm =0; }
            }
    }

```



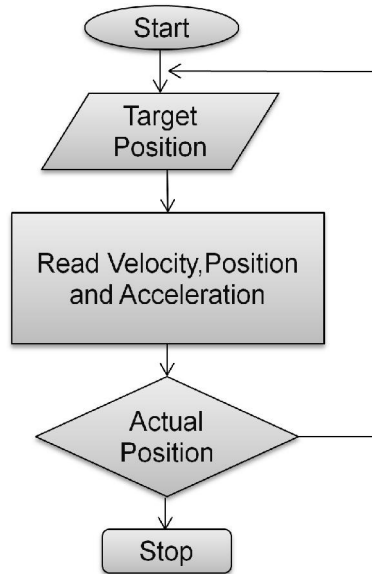
Nilai Vref yang dihasilkan dari sensor arah angin diterima oleh pin ADC mikrokontroler PortA(0) sebagai target dan menghasilkan keluaran tegangan rata-rata. Data dari pengujian sensor arah angin menghasilkan nilai berupa bilangan pecahan hal ini akan menghasilkan nilai error sekitar:

$$\begin{aligned}\text{Error} &= \frac{(\text{Vin perhitungan} - \text{Vin percobaan})}{V_{ref}} \times 100\% \\ &= ((5v - 4.968v) / 5v) \times 100\% \\ &= 0.64\%\end{aligned}$$

Sedngkan nilai Vref yang dihasilkan dari sensor posisi diterima oleh pin ADC mikrokontroler portA(3) dan menghasilkan keluaran tegangan rata-rata. Data dari pengujian sensor Posisi menghasilkan nilai berupa bilangan pecahan hal ini akan menghasilkan nilai error sekitar:

$$\begin{aligned}\text{Error} &= (\text{Vin perhitungan} - \text{Vin percobaan}) \times 100\% \\ &= (2.125 - 2.124) \times 100\% \\ &= 0.1\%\end{aligned}$$

Dari hasil di atas diketahui nilai error input sensor sebesar 0.64% dibandingkan nilai sensor posisi sebagai actual sebesar 0.1%. hal ini membuat nilai aktual sulit mencapai target sebenarnya. Akibatnya untuk mencapai keadaan posisi arah angin sebenarnya mendapatkan banyak kesulitan dalam hal kepresisian, namun perbedaan dalam pengujian tidak terlalu jauh dari perubahan arah angin yang sebenarnya. Jika arah angin menunjukkan utara maka peluncur roket berubah dengan arah yang sama yaitu utara, perbedaan sensor arah angin dengan peluncur roket hanya berbeda sekitar 0.7 derajat.



**Gambar 4.20:** *FlowChart – Manual Kontrol*

Cara kerja algoritma program menu manual peluncur roket yaitu pertamakali set point posisi manual dimasukan melalui keypad, kemudian akan dibaca oleh system sebagai arah posisi peluncur roket. nilai target akan dibandingkan dengan nilai posisi aktual, jika nilai error sama dengan "0" maka motor aktuator akan berhenti. Berikut ini adalah potongan program dalam melakukan uji coba.

```

#include <mega32.h>
#include <delay.h>
#include <math.h>

void main(void)
{ w=''; c = 1;
while (1)
    { keypad(); RocketLauncher(); }; }
  
```

```

void RocketLauncher()
{
    if (i == 1 && w != 42) {x1 = w; x = x1; }
    if (i == 2 && w != 42) {x2 = x1*10 + w; x
= x2; }
    if (i == 3 && w != 42) {x3 = x2*10 + w; x
= x3; } if (i == 4)x=x;
    if (i == 4 || w == 42) {
        DATA=x; while (c)
        {
temp=read_adc(0); temp1=90-((820-temp)*0.078);
if (temp1==DATA) { stop(); c=0; }
else if (temp1<DATA) { maju(); }
else if (temp1>DATA) { mundur(); }
        }
        w=0;i=0;x=0;c = 1; }}

void keypad()
{
    PORTD = 0b11101111;
    if(PIND.0 == 0){w='a';i++;} if(PIND.1 ==
0){w='#';i++;}
    if(PIND.2 == 0){w=0;i++;} if(PIND.3 ==
0){w='*';i++;}
    PORTD = 0b11011111;
    if(PIND.0 == 0){w='d';i++;} if(PIND.1 ==
0){w=9;i++;}
    if(PIND.2 == 0){w=8;i++;} if(PIND.3 ==
0){w=7;i++;}
    PORTD = 0b10111111;
    if(PIND.0 == 0){w='c';i++;} if(PIND.1 ==
0){w=6;i++;}
    if(PIND.2 == 0){w=5;i++;} if(PIND.3 ==
0){w=4;i++;}

```

```
PORTD = 0b01111111;  
    if (PIND.0 == 0) {w='b';i++;}    if (PIND.1 ==  
0) {w=3;i++;}  
    if (PIND.2 == 0) {w=2;i++;}    if (PIND.3 ==  
0) {w=1;i++;}  
}
```

Data dari pengujian pada mode manual menghasilkan perubahan yang sangat bagus, perbedaannya tidak terlalu jauh dimana nilai input keypad dibandingkan dengan nilai aktual yang dibaca dari sensor posisi. Jika menghasilkan nilai Error = '0', maka akan dianggap tercapai target.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Pada bagian ini akan diulas tentang kesimpulan dari seluruh bab yang telah dikerjakan. Berikut adalah beberapa kesimpulan yang dapat diambil diantaranya adalah sbb:

1. Pada bab pertama penelitian tugas akhir kali ini begitu luas sekali sehingga perlu dilakukan batasan agar lebih fokus ke dalam suatu permasalahan.
2. Bagian dari bab kedua di atas merupakan materi yang dipelajari pada saat awal pembuatan alat hingga pada proses pembuatan alat. Materi di atas adalah pengetahuan dasar untuk mengenal materi selanjutnya.
3. Dari pada bab ketiga penelitian tugas akhir ini rancangan disain bentuk peluncur roket. Banyak sekali bagian yang harus dibuat untuk menghasilkan peluncur roket dan mempunyai tingkat kepresisian yang tinggi.
4. Pada bab keempat penelitian tugas akhir desain yang selalu berubah-ubah karena menyesuaikan ketersediaan barang di pasaran.
5. Pada bab kelima hasil uji coba PID terhadap pergerakan mekanik roket launcher dapat ditarik beberapa kesimpulan yaitu :
  - a) kinerja kontrol harus ditentukan dengan mematuhi seluruh tujuan penting operasi mekanik secara teliti, perilaku yang diinginkan dari variabel kontrol dan manipulasi harus ditentukan untuk gangguan yang diharapkan, model eror dan pengukuran noise.
  - b) Perilaku kinematik dari kedua variabel kontrol dan manipulasi diperlukan untuk mengevaluasi kinerja dari sistem kontrol umpan balik.
  - c) Secara praktik, nilai dari optimasi atau hubungan digunakan sebagai nilai awal untuk diaplikasikan pada sistem fisik dan ditingkatkan berdasarkan kinerja empiris selama penyetelan halus. Pada saat melakukan penyetelan sebuah kontroler umpan balik, titik awal tidaklah lebih penting dari pada titik akhir.

Kendala yang di hadapi dalam proses pembuatan mekanik

1. Bahan-bahan yang didapat dipasaran tidak memadai sehingga merubah desain.
2. Keterbatasan tools pada alat-alat tertentu yang tersedia di bengkel manufaktur , menyebabkan kita harus selalu menyesuaikan desain yang kita buat.
3. Mahalnya beberapa komponen yang harus di beli menyebabkan kita mengganti dengan komponen yang mempunyai nilai manfaat yang sama dengan harga yang lebih murah, konsekuensinya tentunya pada kualitas mekanik yang kita buat.

Kendala yang di hadapi dalam proses pembuatan rangkaian elektronika

1. Karena menggunakan ATmega32 konsekuensinya adalah pada pembuatan program yang dibatasi memory sedangkan peripheral yang terhubung seperti sensor atau motor memerlukan memory yang cukup banyak.
2. Banyaknya jumlah motor yang harus di kontrol dan sensor posisi rocket launcher berdampak pada rumitnya membuat sistem yang sederhana.
3. Driver motor dengan motor 12-24 volt yang harus dibuat yang memerlukan waktu dan biaya yang banyak.

## **5.2 Saran**

Adapun saran yang bisa diberikan dalam pengembangan perangkat keras ini lebih lanjut adalah sebagai berikut :

1. Menghitung secara tepat, konsep yang akan dibuat untuk mencegah atau meminimalisir error di kemudian hari.
2. Menambah jam kerja dalam proses pembuatan proyek akhir agar bisa selesai tepat waktu dan sesuai target.
3. Meminimalisir kesalahan untuk mengurangi beban biaya yang ditimbulkan dari kesalahan tersebut serta meng- efisienkan waktu.

## DAFTAR PUSTAKA

- Baumann, Oliver "Type of documents/Cut-away drawings/Actuators". auma Riester GmbH & Co. KG. September 28, 2008.
- Bracewell, R. N. (2000), *The Fourier Transform and Its Applications* (3rd ed.), Boston: McGraw-Hill, ISBN 0071160434.
- Budiarto, Widodo. (2008). "Panduan Praktikum Mikrokontroler AVR ATMega16". Jakarta: PT Elex Media Komputindo.
- D.L. Pieper. *The kinematics of manipulators under computer control*. PhD Thesis, Stanford University, Department of Mechanical Engineering, 1968.
- Franklin et al. (2002). *Feedback Control of Dynamic Systems* (4 ed.). New Jersey: Prentice Hall. ISBN 0-13-032393-4.
- Heriyanto, Abdul Kadir. (2005). "Algoritma Pemrograman menggunakan C++". Yogyakarta: Penerbit ANDI.
- Kuo, Benjamin C. (1991). *Automatic Control Systems* (6th ed.). New Jersey: Prentice Hall. ISBN 0-13-051046-7.
- LAPAN. (2009). "Lomba Muatan Roket Payload Meteorologi". Panduan Peserta. Pandansimo, Yogyakarta, Indonesia.
- Middleton, The Johns Hopkins Press, Baltimore, 1969.
- Pitowarno, E. (2006). *ROBOTIKA: Desain, Kontrol dan Kecerdasan Buatan*. Buku Teks. Yogyakarta: Penerbit Andi.
- Slater, N., *Mechanisms and Mechanical Devices Source book*, 4th Edition (2007), 25, McGraw-Hill.
- Shaller, Robert R. (15 April 2004). "Dissertation: Technological

Innovation in the Semiconductor Industry: A Case Study of the International Technology Roadmap for Semiconductors".

The Authoritative Dictionary of IEEE Standards Terms (IEEE 100) (edisi ke-edisi ketujuh). Piscataway, New Jersey: IEEE Press. 2000. ISBN 0-7381-2601-2.

Walden, R. H. (1999), "Analog-to-digital converter survey and analysis", IEEE Journal on Selected Areas in Communications 17 (4): 539–550, doi:10.1109/49.761034, ISSN 0733-8716.

( [http://en.wikipedia.org/wiki/Rotary\\_encoder](http://en.wikipedia.org/wiki/Rotary_encoder) )

( [http://en.wikipedia.org/wiki/Transmission\\_\(mechanics\)](http://en.wikipedia.org/wiki/Transmission_(mechanics)) )

( [http://en.wikipedia.org/wiki/Torque\\_and\\_speed\\_of\\_a\\_DC\\_motor](http://en.wikipedia.org/wiki/Torque_and_speed_of_a_DC_motor) )

( <http://docs-europe.electrocomponents.com/webdocs> )