

repository.ub.ac.id

**PERANCANGAN KESEIMBANGAN GERAK *PITCH* PADA *BICOPTER*
SECARA STATIS DENGAN METODE MENGUBAH-UBAH *PITCH PROPELLER*
MENGUNAKAN KONTROLER PID**

SKRIPSI
TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh :

ARISKI FADILLAH

NIM. 105060307111007

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2016





LEMBAR PENGESAHAN

**PERANCANGAN KESEIMBANGAN GERAK PITCH PADA
BICOPTER SECARA STATIS DENGAN METODE
MENGUBAH-UBAH PITCH PROPELLER MENGGUNAKAN
KONTROLER PID**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL

Diajukan untuk Memenuhi Persyaratan
Memperoleh Gelar Sarjana Teknik



ARISKI FADILLAH
NIM. 105060307111007

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
pada tanggal 29 Januari 2016

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Ir. Bambang Siswojo, MT
NIP. 19621211 198802 1 001

Ir. Purwanto, MT.
NIP. 19540424 198601 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

M. Azis Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

JUDUL SKRIPSI :

Perancangan keseimbangan Gerak *Pitch* Pada *Bicopter* Secara Statis Dengan Metode Mengubah – Ubah *Pitch Propeller* Menggunakan Kontroler PID

Nama Mahasiswa : Ariski Fadillah

Nim : 105060307111007

Program Studi : Teknik Elektro

Konsentrasi : Teknik Kontrol

Komisi Pembimbing :

Ketua : Dr. Ir. Bambang Siswojo, M.T

Anggota : Ir. Purwanto, M.T

Tim Dosen Penguji :

Dosen Penguji 1 : Ir. Retnowati, M.T

Dosen Penguji 2 : M. Aziz Muslim, S.T, M.T., Ph.D

Dosen Penguji 3 : Ir. Dipl.Ing. Moch. Rusli

Tanggal Ujian : 22 Januari 2016

SK Penguji : No. 129/UN10.6/SK/2016

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 14 Januari 2016

Mahasiswa,
(Materai Rp 6.000,-)

(Tanda Tangan)

Ariski Fadillah

NIM. 105060307111007



RINGKASAN

Ariski Fadillah, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2016, *Perancangan keseimbangan Gerak Pitch Pada Bicopter Secara Statis Dengan Metode Mengubah – Ubah Pitch Propeller Menggunakan Kontroler PID*, Dosen Pembimbing : Bambang Siswojo dan Purwanto.

Seiring dengan perkembangan teknologi modern, saat ini robot udara atau yang sering disebut *Unmanned Aerial Vehicle (UAV)* sering digunakan sebagai alat bantu pengambilan gambar dari udara dari sudut-sudut yang sulit dan berbahaya untuk dijangkau manusia. Salah satu jenis UAV yang dapat membantu manusia adalah *multicopter*. *Multicopter* yang memiliki dua baling-baling disebut *bicopter*. Dalam perancangan *bicopter* ada beberapa hal yang mempengaruhi keseimbangan antara lain panjang *frame*, berat total, dan gaya dorong motor (*thrust*). Untuk mendesain *frame bicopter* secara lengkap diperlukan alat uji satu *frame*. Alat bantu desain *frame bicopter* merupakan model satu *frame* dengan satu aktuator motor dan *Propeller*. Sisi lainnya merupakan beban uji yang dapat diubah-ubah beratnya.

Proses perancangan PID pada penelitian ini menggunakan metode 1 Ziegler-Nichols pada setpoint 2,5 V menghasilkan $K_p = 3,6$, $K_i = 18$, dan $K_d = 0,24$ yang menunjukkan bahwa respon sistem secara keseluruhan tidak melebihi 5% dari setpoint dan mampu kembali stabil ketika mendapatkan gangguan perubahan beban uji. Hal ini menunjukkan bahwa kontroler PID dapat mengendalikan *pitch* (sudut) dengan baik.

Kata kunci : *Bicopter, Pitch Propeller, PID, UAV*

SUMMARY

Ariski Fadillah, Department of Electrical Engineering , Faculty of Engineering, University of Brawijaya , January 2016, *Design Motion Of Pitch Balance In Immovable Bicopter With Changing Pitch Of Propeller Method Using PID Controller*, Academic Supervisor : Bambang Siswojo and Purwanto .

Along with the development of modern technology, today's robots air or often called *unmanned Aerial Vehicle* (UAV) is often used as a tool for taking pictures of air from the corners that are difficult and dangerous to reach humans . One type of UAV that can help people is Multicopter. Multicopter which has two propeller called bicopter. In designing bicopter there are several things that affect the balance between the other frame length, total weight, and the motor thrust force (*thrust*). To design a complete frame bicopter required test equipment one frame. Bicopter frame design tool is a model of the frame with an actuator motor and the propeller. The other side is the test load can be varied severity

The process of designing PID in this study using the Ziegler - Nichols method 1 on setpoint of 2.5 V produces $K_p = 3.6$, $K_i = 18$, and $K_d = 0.24$ which indicates that the overall system response does not exceed 5 % of setpoint and capable stabilized when getting interference test load changes . This indicates that the PID controller can control the pitch (*Angle*) well .

Keywords : Bicopter , Pitch Propeller , PID , UAV

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa tanpa bantuan, bimbingan serta dorongan dari semua pihak penyelesaian skripsi ini tidak mungkin bisa terwujud. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

- Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
- Kedua orang tua Arfan Fadillah dan Sunarwati serta kakak tercinta Rohmi Indarti yang telah banyak memberikan do'a, kasih sayang, dukungan, serta semangat.
- Bapak Ir. Purwanto, M.T. selaku KKDK Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang sekaligus dosen pembimbing skripsi.
- Bapak Hadi Suyono ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
- Bapak Dr. Ir. Bambang Siswojo, MT. dan Ir. Purwanto, M.T. selaku dosen pembimbing skripsi.
- Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
- Dandy, Randy, Bagus terutama yang sangat membantu skripsi saya.
- Teman – teman “MaGiC” Rainer, Dany, Mukson, Hanip, Rizki, yang telah meluangkan waktu disaat penat mengerjakan skripsi, pelajaran hidup, serta canda dan tawa
- Abas, Irfan, Marsudi, Dita nirmala yang telah meluangkan waktu untuk bercanda dan bercengkerama bersama selama mengerjakan skripsi.

- Tak lupa keluarga besar MAGNET'10, yang memberikan semangat, do'a, dan dukungan.
- Faradita Nova Wijayanti. Terima kasih atas semangat yang selalu di berikan.
- Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala – kendala lain yang terjadi selama pengerjaan skripsi ini. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Januari 2016

Penulis



DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
DAFTAR LAMPIRAN	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 Baling-baling (<i>Propeller</i>)	5
2.2 Motor BLDC (Brushless Direct Current)	6
2.3 Motor Servo	7
2.4 Catu Daya	9
2.5 ESC (Electronic Speed Control)	10
2.6 PWM Digital Controller	10
2.7 Kontroler	11
2.7.1 Kontroler Proporsional	12
2.7.2 Kontroler Integral	13
2.7.3 Kontroler Differensial	14
2.7.4 Kontroler Proporsional Integral (PI)	14
2.7.5 Kontroler Proporsional Derivative (PD)	15
2.7.6 Kontroler Proportional Integral Derivative (PID)	15
2.7.7 Metode Perancangan Kontroler PID	16
2.7 Mikrokontroler	20
2.7.1 Mikrokontroler ATmega328	21
2.8 Arduino Mega	22
BAB III METODOLOGI PENELITIAN	23



3.1	Penentuan Spesifikasi Alat	23
3.2	Studi Literatur	23
3.3	Perancangan dan Realisasi Pembuatan Alat	24
3.3.1	Perancangan Perangkat Keras dan Realisasi Pembuatan Alat	24
3.3.2	Perancangan dan Perhitungan Komponen yang akan Digunakan	24
3.3.3	Perancangan Perangkat Lunak	24
3.4	Pengujian Alat	24
3.5	Pengambilan Kesimpulan dan Saran	25
BAB IV PERANCANGAN DAN PEMBUATAN ALAT		27
4.1	Perancangan Sistem	27
4.1.1	Perancangan Sistem Mekanik	28
4.1.2	Perancangan Sistem Kontroler	28
4.2	Diagram Blok Sistem	29
4.2.1	Alasan menggunakan <i>input</i> 75%	30
4.3	Perancangan Perangkat Keras	31
4.3.1	Propeller dan Motor BLDC	31
4.3.2	Sensor Potensiometer	31
4.3.3	Modul arduino Mega	32
4.4	Perancangan Kontrol PID	32
4.4.1	Penentuan Nilai Penguatan Kontroler	32
4.5	Perancangan Perangkat Lunak	34
BAB V PENGUJIAN DAN ANALISIS		37
5.1	Pengujian Sensor	37
5.2	Pengujian respon aktuator Motor DC Brushless	40
5.3	Analisis	42
5.4	Pengujian sistem keseluruhan	44
BAB VI KESIMPULAN DAN SARAN		47
6.1	Kesimpulan	47
6.2	Saran	47

DAFTAR GAMBAR

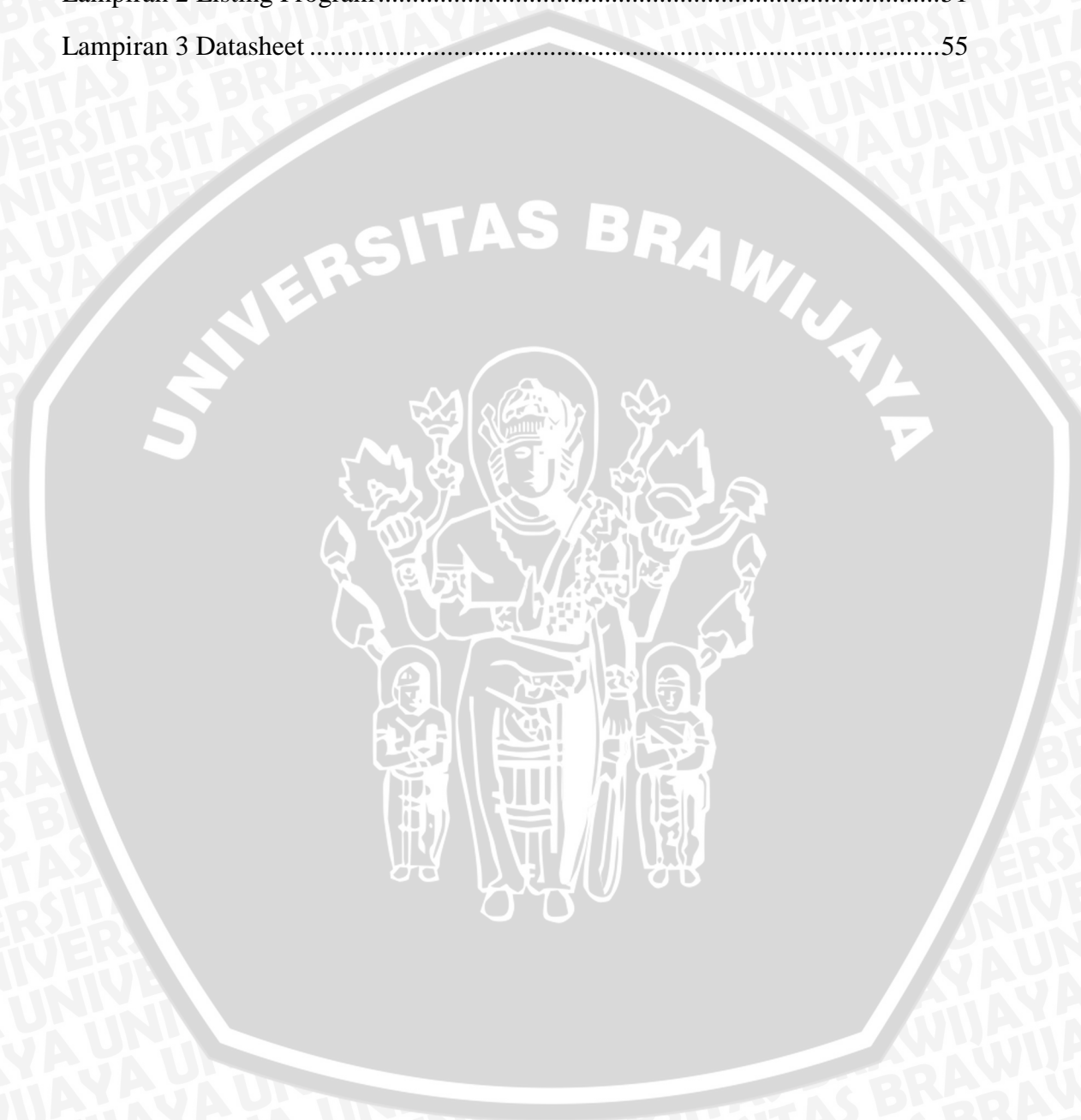
NO.	Judul	Halaman
Gambar 2.1	Baling-baling.....	5
Gambar 2.2	Motor BLDC	7
Gambar 2.3	Motor Servo.....	7
Gambar 2.4	Cara Kerja Motor Servo	9
Gambar 2.5	ESC	7
Gambar 2.6	PWM <i>Digital Controller</i>	10
Gambar 2.7	Sinyal PWM Secara Umum	10
Gambar 2.8	Diagram blok kontroler proporsional	12
Gambar 2.9	Blok diagram kontroler integral	13
Gambar 2.10	Blok diagram kontroler differensial	14
Gambar 2.11	Diagram blok kontroler PID	15
Gambar 2.12	Hubungan fungsi waktu antara sinyal keluaran	16
Gambar 2.13	Kurva Respon Unit step	17
Gambar 2.14	Respon Plant Terhadap Masukan Berupa Unit Step.....	17
Gambar 2.15	Kurva Respon yang Berbentuk S.....	18
Gambar 2.16	Sistem Loop Tertutup dengan Kontroler Proporsional.....	19
Gambar 2.17	Osilasi Berkesinambungan dengan Periode P_{cr1}	19
Gambar 2.18	Board Arduino Mega	22
Gambar 4.1	Gambar skema alat uji satu frame.....	28
Gambar 4.2	Gambar Foto alat uji stau frame	28
Gambar 4.3	Diagram Blok Sistem frame.....	29
Gambar 4.4	<i>Propeller dan Motor BLDC</i>	31
Gambar 4.5	Sensor Potensiometer.....	31
Gambar 4.6	Arduino mega.....	32
Gambar 4.7	Kurva Respon yang Berbentuk S	32
Gambar 4.8	Karakteristik Respon <i>Dimming</i>	33
Gambar 4.9	<i>Flowchart</i> Perangkat Lunak.....	35
Gambar 5.1	Diagram Blok Pengujian Sensor dengan Beban	38
Gambar 5.2	Grafik perbandingan antara PWM terhadap sudut	39
Gambar 5.2.2	Grafik perbandingan antara PWM	39

Gambar 5.3	Respon aktuator dalam grafik	42
Gambar 5.3.1	Respon aktuator dalam grafik	43
Gambar 5.4	Diagram Perancangan Pengujian keseluruhan sistem.....	44
Gambar 5.5	Grafik hasil pengujian keseluruhan.....	46



DAFTAR LAMPIRAN

No	Judul	Halaman
	Lampiran 1 Foto Alat	49
	Lampiran 2 Listing Program.....	51
	Lampiran 3 Datasheet	55



UNIVERSITAS BRAWIJAYA



BAB I PENDAHULUAN

1.1 Latar Belakang

Robot merupakan hasil karya teknologi yang dalam hal tertentu dapat digunakan untuk menyelesaikan permasalahan-permasalahan yang dihadapi dalam kehidupan manusia. Terutama agar proses-proses yang dilakukan secara manual dapat bekerja secara otomatis, terutama pekerjaan yang memerlukan ketelitian proses yang tinggi, daerah kerja yang berbahaya, memerlukan proses yang berulang-ulang untuk mendapatkan produk dengan kualitas produksi yang seragam serta alasan-alasan lainnya. Robot udara yang berkembang saat ini adalah pesawat udara yang bekerja secara *auto-pilot* atau tanpa awak. Robot pesawat udara *auto-pilot* ini biasa disebut dengan *Unmanned Aerial Vehicle* atau disingkat UAV. Seiring dengan perkembangan teknologi modern, saat ini UAV sering digunakan sebagai alat bantu pengambilan gambar dari udara dari sudut-sudut yang sulit dan berbahaya untuk dijangkau manusia. UAV juga diaplikasikan untuk untuk kebutuhan militer, seperti pengintaian atau bahkan invasi militer. UAV sangat aplikatif untuk kebutuhan ini dimana dapat mengintai wilayah musuh tanpa membahayakan pilot. Atau dapat mengambil gambar dari wilayah yang belum diketahui. Salah satu jenis UAV yang dapat membantu manusia adalah *bicopter*. *bicopter* merupakan produk robot udara yang digerakkan oleh sejumlah baling-baling yang dimiliki. Semakin banyak baling-baling yang dimiliki, kecepatan, daya angkat dan keseimbangan *bicopter* akan meningkat, sehingga kapasitas beban yang diangkat akan semakin meningkat. (Kristianto, D, 2012)

Dalam pemanfaatan *bicopter* untuk berbagai tujuan unjuk kerja, kestabilan pada saat keadaan melayang (*hover*) pada *bicopter* sangatlah penting dan harus dimiliki *bicopter* agar pemanfaatannya dapat secara maksimal. Dalam perancangan *bicopter* ada beberapa hal yang mempengaruhi keseimbangan antara lain adalah panjang *frame*, berat total, dan gaya dorong motor (*thrust*). Semakin pendek panjang *frame* artinya semakin ringan beban pada sisi tersebut. Begitu pula sebaliknya, semakin panjang *frame* maka semakin berat pula beban pada sisi

tersebut. Motor menghasilkan gaya dorong sesuai spesifikasi. Motor digabungkan bersama dengan *propeller* dengan spesifikasi tertentu.

Untuk mendesain *frame bicopter* secara lengkap diperlukan alat uji satu *frame*. Alat bantu desain *frame bicopter* merupakan model satu *frame* dengan satu aktuatur motor dan *propeller*. Sisi lainnya merupakan beban uji yang dapat diubah-ubah beratnya. Dengan menggabungkan kontroler PID, dapat diuji kestabilan yang dihasilkan dengan mengubah-ubah parameter, panjang *frame*, spesifikasi motor dan *propeller* dengan konstanta PID tertentu. Parameter yang diperoleh dari alat uji untuk membantu desain *bicopter* akan dapat digunakan sebagai dasar desain *bicopter* secara keseluruhan.

Dalam skripsi ini penulis merancang sebuah alat uji coba statis satu lengan. Alat uji coba ini adalah sebuah alat yang mampu mengendalikan kestabilan horizontal dengan menggunakan metode Kontrol Proporsional Integral Derifensial.

1.2 Rumusan Masalah

1. Bagaimana mendesain model mekanik satu *frame*?
2. Bagaimana merancang *hardware* dan *software* sistem pengendalian pitch pada *propeller* menggunakan metode Proporsional Integral Derifensial?

1.3 Batasan Masalah

Untuk menekankan pada objek pembahasan yang ada, maka penelitian ini diberikan batasan masalah sebagai berikut :

1. Pembahasan ditekankan pada aplikasi Kontroler Proporsional Derivative.
2. Sensor yang digunakan adalah sensor Potensiometer
3. Arduino mega sebagai pusat pengendali sistem
4. Gangguan diberikan dengan beban yang diubah-ubah (berat beban berubah- ubah) pada lengan sebelahnya.
5. Kinerja Rangkaian elektrik tidak dibahas mendalam.

1.4 Tujuan

Merancang dan membuat alat bantu desain *frame bicopter* secara statis satu *frame* dengan Kontroler Proporsional Integral Diferensial.

1.5 Sistematika Penulisan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini secara garis besar adalah sebagai berikut :

BAB I

Pendahuluan

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

BAB II

Tinjauan Pustaka

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

BAB III

Metodologi Penelitian

Menjelaskan tentang metodologi penelitian yang terdiri dari studi literatur, perancangan alat, pembuatan alat, pengujian alat, serta pengambilan kesimpulan dan saran.

BAB IV

Perancangan dan Pembuatan Alat

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras, dan perangkat kontroler PID dan menerapkannya ke dalam *software*, sehingga sistem dapat bekerja dengan baik.

BAB V

Pengujian dan Analisis

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem dan pengujian sistem secara keseluruhan.

BAB VI

Kesimpulan dan Saran

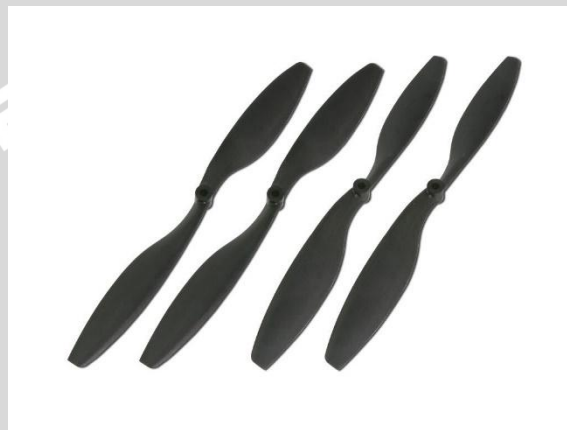
Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil masalah, serta pemberian saran untuk pengembangan



BAB II TINJAUAN PUSTAKA

2.1 Baling-baling (*Propeller*)

Baling-baling adalah alat yang mengubah gerak putar menjadi daya dorong. Daya dorong inilah yang dimanfaatkan pesawat terbang dan kapal laut sebagai penghasil daya dorong utama. Pembahasan baling-baling pada skripsi ini dibatasi hanya pada parameter baling-baling yang digunakan dalam RC (*Radio Control*) *aeromodelling*. Bentuk *propeller* ditunjukkan pada Gambar 2.1 berikut.



Gambar 2.1 Propeller

Sumber: <http://helitech-jp.com/>

Ada beberapa parameter penting yang dimiliki baling-baling pada RC *aeromodelling*. Parameter-parameter ini bisa dijadikan pedoman untuk memilih baling-baling sesuai dengan kebutuhan :

1. Diameter dan *pitch*

Semua baling-baling RC yang tersedia memiliki 2 buah ukuran, yaitu diameter dan *pitch*. Diameter dihitung berdasarkan diameter lingkaran yang dibentuk saat baling-baling berputar. Jika baling dianalogikan sebagai sebuah sekrup, *pitch* merupakan jarak yang ditempuh oleh baling-baling jika diputar 1 putaran penuh. Semakin panjang diameter dan *pitch* maka baling-baling semakin banyak pula udara yang disapu dan semakin besar pula daya dorong yang dihasilkan. Tapi diameter dan *pitch* dari baling-baling ini harus disesuaikan dengan motor dan sumber daya yang digunakan. Biasanya produsen motor sudah memberikan spesifikasi baling-baling untuk motornya.

2. Jumlah bilah

Umumnya, jumlah bilah pada baling-baling RC *aeromodelling* adalah 2 bilah, Tetapi ada beberapa yang menggunakan 3 bilah dan 4 bilah. Semakin banyak bilah pada baling-baling menyebabkan banyak udara yang disapu sehingga menghasilkan daya dorong yang lebih besar. Biasanya penambahan jumlah bilah bertujuan untuk memperkecil diameter baling-baling, tentunya untuk menghasilkan performa yang sama (dengan motor yang sama) *pitch*-nya harus dikurangi.

3. Arah putar

Dengan arah gaya dorong yang sama, baling-baling RC *aeormodelling* memiliki dua jenis arah putaran, searah jarum jam (CW/*clockwise*) dan berkebalikan arah jarum jam (CCW/*counter clockwise*). Arah putar ini menentukan *yawing moment* yang dihasilkan dari baling-baling. Pada *Bicopter*, dibutuhkan sepasang baling-baling CW dan CCW agar *yawing moment* saling menghilangkan.

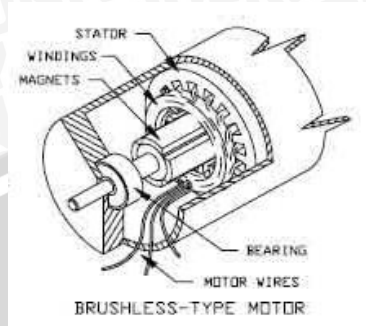
2.2 Motor BLDC (*Brushless Direct Current*)

Bicopter membutuhkan penggerak berupa baling-baling yang diputar oleh motor. Spesifikasi yang harus dipenuhi oleh sistem gerak ini adalah torsi, efisiensi dan getaran yang ditimbulkan oleh berputarnya motor dan baling-baling. Motor dengan getaran yang terlalu besar dapat mengganggu sensor-sensor yang digunakan pada *Attitude Heading Reference System* (AHRS). AHRS merupakan integrasi dari beberapa sensor dan menggunakan perhitungan tertentu untuk memadukan data dari sensor-sensor tersebut. Efisiensi motor berkaitan dengan durabilitas terbang dari pesawat. Mengingat sumber daya (*battery*) yang digunakan terbatas.

Brushless Direct Current Motor atau biasa disebut BLDC adalah motor DC yang proses komutasinya tidak menggunakan sikat seperti motor DC pada umumnya. Dibandingkan dengan motor DC dengan sikat, BLDC memiliki beberapa kelebihan yaitu, efisiensi tinggi, kecepatan dan torsi yang tinggi, respon dinamis yang tinggi, masa operasi yang panjang dan operasi tanpa *noise*. Sehingga dengan kelebihan-kelebihan tersebut, BLDC banyak digunakan pada aplikasi *aeromodelling* dan termasuk pada *bicopter*.

Motor BLDC adalah tipe motor sinkron. Artinya medan magnet yang dihasilkan oleh stator dan medan magnet yang dihasilkan oleh rotor mempunyai frekuensi yang sama. Rotor (Bagian motor yang berputar) pada BLDC terdiri dari

magnet permanen, sedangkan stator terdiri dari kumparan. Berbeda dengan motor DC dengan sikat, dimana rotor berupa lilitan dan stator berupa magnet tetap. Bentuk motor BLDC ditunjukkan pada Gambar 2.2.

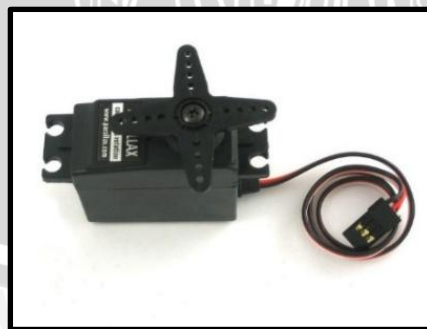


Gambar 2.2 Motor BLDC

Sumber: <http://www.rcspark.com>

2.3 Motor Servo

Motor *servo* adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (*servo*), sehingga dapat di *set-up* atau di atur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Bentuk motor servo ditunjukkan pada Gambar 2.3.



Gambar 2.3 Motor Servo

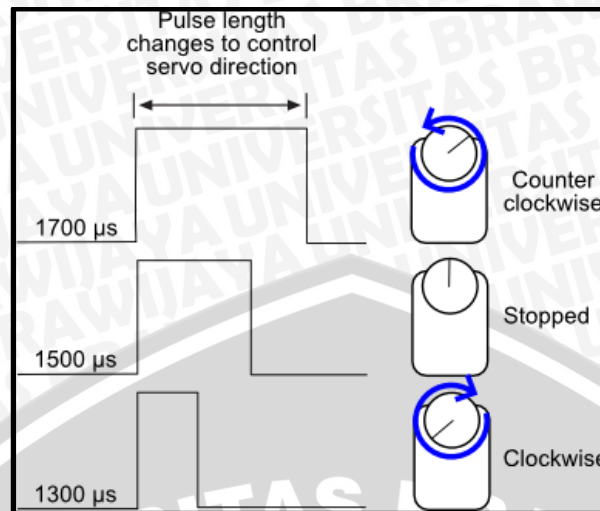
Sumber : Adafruit, 2014

Penggunaan sistem kontrol *loop* tertutup pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo. Penjelasan sederhananya begini, posisi poros *output* akan di sensor untuk mengetahui posisi poros sudah tepat seperti yang di inginkan atau belum, dan jika belum, maka kontrol *input* akan mengirim sinyal kendali untuk membuat posisi poros tersebut tepat pada posisi yang diinginkan.

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu motor servo rotation 180° dan servo *rotation continuous*.

1. Motor servo standar (*servo rotation* 180°) adalah jenis yang paling umum dari motor servo, dimana putaran poros keluarannya terbatas hanya 90° kearah kanan dan 90° kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180° .
2. Motor servo *rotation continuous* merupakan jenis motor servo yang sebenarnya sama dengan jenis servo standar, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Wide Modulation / PWM*) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo *continuous*. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo *continuous* akan berhenti. Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah arah jarum jam, sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo *continuous* akan berputar ke arah berlawanan jarum jam. Lebih jelasnya perhatikan Gambar 2.4.



Gambar 2.4 Cara Kerja Motor Servo

Sumber : parallax.com

2.4 Catu Daya

Sumber tegangan atau catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah *payload*. Tanpa bagian ini *payload* tidak akan berfungsi. Begitu pula dengan pemilihan sumber tegangan yang tidak tepat, maka *payload* tidak akan bekerja dengan baik.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh beberapa faktor, diantaranya :

1. Tegangan

Setiap aktuator atau motor tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap desain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

2. Arus

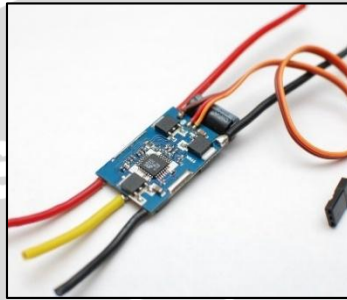
Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin lama daya tahan baterai bila digunakan pada beban yang sama.

3. Teknologi Baterai

Baterai isi ulang ada yang dapat diisi kapan saja, dan ada pula yang harus diisi ulang sebelum batas tegangan minimum.

2.5 ESC (Electronic Speed Control)

Motor *brushless* memiliki sebuah ESC (*Electronic Speed Control*) yang berfungsi sebagai pengatur kecepatan motor, selain itu juga berfungsi untuk menaikan jumlah arus yang diperlukan oleh motor (Chmelai, 2011). Kecepatan untuk motor yang keluar dari ESC diatur melalui pulsa dari mikrokontroler. Bentuk ESC ditunjukkan pada Gambar 2.5.



Gambar 2.5 ESC

Sumber: electronics.stackexchange.com

2.6 PWM Digital Controller

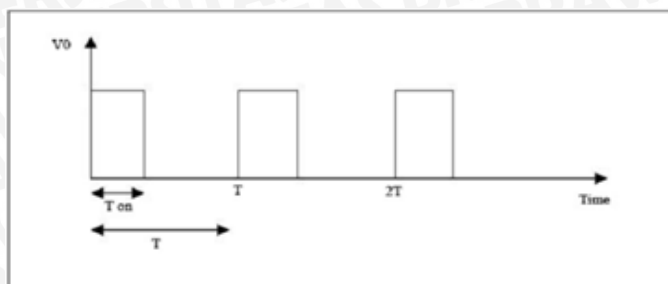
PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. Bentuk PWM ditunjukkan pada Gambar 2.6.



Gambar 2.6 PWM Digital Controller

Sumber: Dokumen Pribadi

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat pada Gambar 2.7 :



Gambar 2.7 Gambar Sinyal PWM Secara Umum

Sumber: *electronics-scheme.com*

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \dots \dots \dots (2-1)$$

Dengan :

T_{on} = Periode logika tinggi

T = Periode keseluruhan

$$V_{dc} = Duty\ cycle \times V_{cc} \dots (V) \dots \dots \dots (2-2)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{0n} = \frac{f_{clk} / 0}{N \cdot 256} \dots (Hz) \dots \dots \dots (2-3)$$

2.7 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik plant harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu sub sistem, yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem kontrol mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

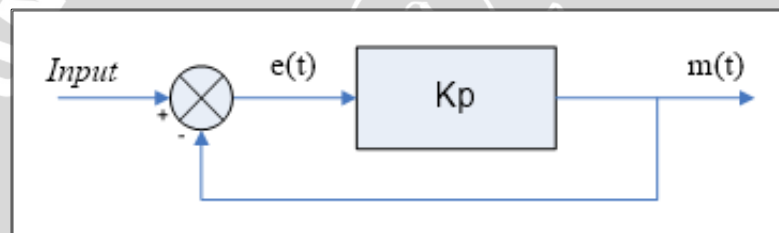
Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengamati perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi *plant*. Dengan demikian

sistem secara cepat mengubah keluaran plant sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plant dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1996).

2.7.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan/*error*. Dapat dikatakan bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya. Untuk lebih jelasnya dapat dilihat pada blok diagram pada Gambar 2.8



Gambar 2.8 Diagram blok kontroler proporsional

Sumber: Ogata, 1995, 157

Pada Gambar 2.8 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.

Pada pengendali proporsional hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan $e(t)$ adalah sebagai berikut :

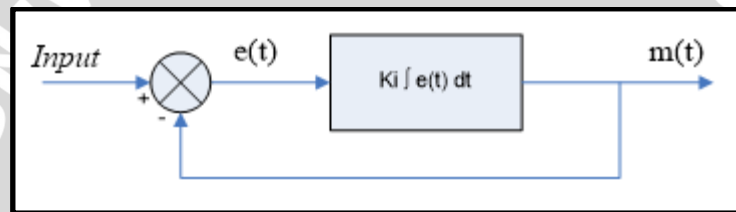
$$m(t) = K_p e(t) \quad (2 - 4)$$

Dengan K_p adalah penguatan proporsional, keluaran $m(t)$ hanya bergantung pada K_p dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan K_p akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap.

2.7.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah plant tidak memiliki unsur integrator (1/s), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Pada Gambar 2.9 menunjukkan blok diagram kontroler integral.



Gambar 2.9 Blok diagram kontroler integral

Sumber: Ogata, K, 2002

Nilai keluaran kontroler $m(t)$ sebanding dengan integral sinyal kesalahan $e(t)$, Sehingga

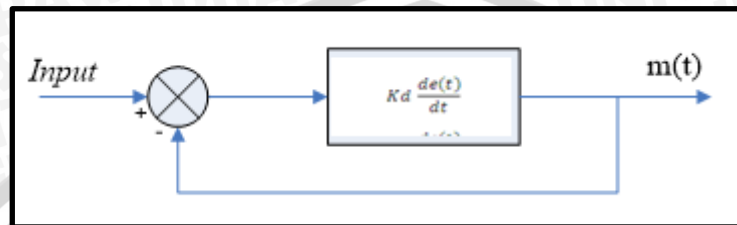
$$\frac{dm(t)}{dt} = Ki \cdot e(t) \tag{2 - 5}$$

$$m(t) = Ki \int_0^t e(t)dt \tag{2 - 6}$$

dengan Ki adalah konstanta integral. Jika sinyal kesalahan $e(t)=0$, maka laju perubahan sinyal kendali integral $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi control integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

2.7.3 Kontroler Differensial

Kontroler differensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.10. berikut menunjukkan blok diagram pada kontroler differensial.



Gambar 2.10 Blok diagram kontroler differensial

Sumber: Ogata, K., 2002

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$. Hubungan ini dapat ditulis sebagai :

$$m(t) = Kd \frac{de(t)}{dt} \quad (4)$$

Kontroler diferensial akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

2.7.4 Kontroler Proporsional Integral (PI)

Aksi kontrolnya dinyatakan dalam persamaan :

$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \int_0^t e(t) dt \quad (5)$$

kontroler ini menghasilkan sinyal kesalahan $\int e(t) dt$ kemudian ditambahkan dengan sinyal kesalahan $e(t)$.

2.7.5 Kontroler Proporsional Derivative (PD)

Aksi kontrolnya dinyatakan dalam persamaan :

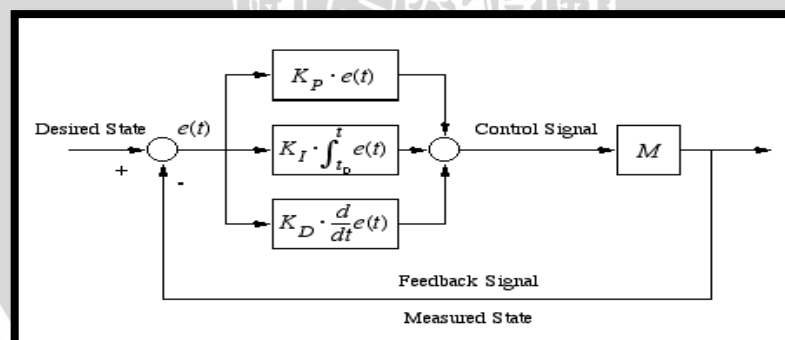
$$m(t) = K_p \cdot e(t) + K_p \cdot T_d \frac{de(t)}{dt} \quad (6)$$

Kontroler PD selalu mengukur kemiringan (slope) sinyal kesalahan $\frac{de(t)}{dt}$ dan memperkirakan akan besar *overshoot* yang akan terjadi serta memberikan koreksi sebelum terjadi lewatannya sehingga diperoleh *maximum overshoot* yang kecil.

Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunan terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat perubahan kesalahan, kontroler PD akan mengurangi besar kesalahan keadaan mantap. Jadi kontrol PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai *maximum overshoot* yang berlebihan tanpa memperhitungkan kecepatan responnya.

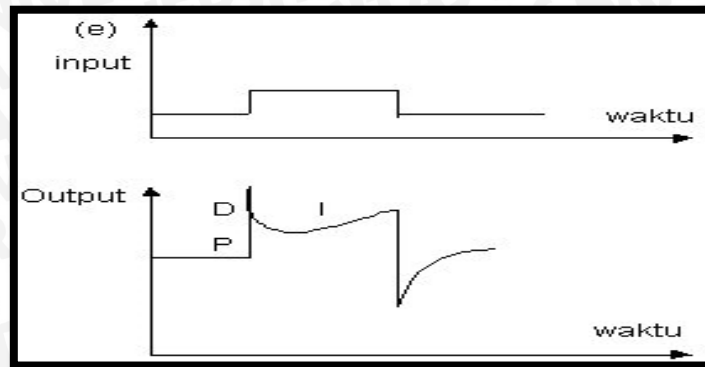
2.7.6 Kontroler Proportional Integral Derivative (PID)

Setiap kekurangan dan kelebihan masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel yang ditujukan pada Gambar 2.11.



Gambar 2.11 Diagram blok kontroler PID

Kontroler proporsional integral derivative (PID). Elemen-elemen P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan pada Gambar 2.12.



Gambar 2.12 Hubungan fungsi waktu antara sinyal keluaran dan masukan kontroler PID

Sumber: Gunterus, 1994:8-11

Aksi kontrolnya dapat dinyatakan sebagai berikut :

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (7)$$

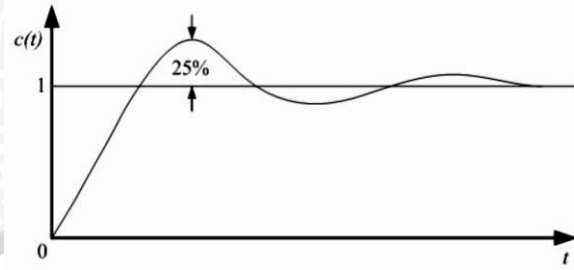
Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon dan mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional integral dan derivative. Gambar 2.12 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , K_i dan K_d akan mengakibatkan penjonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibandingkan yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.7.7 Metode Perancangan Kontroler Proporsional Integral Defereensial (PID) Menggunakan Metode Ziegler-Nichols

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan. (Ogata, K., 1997)

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, seperti ditunjukkan pada Gambar 2.13.

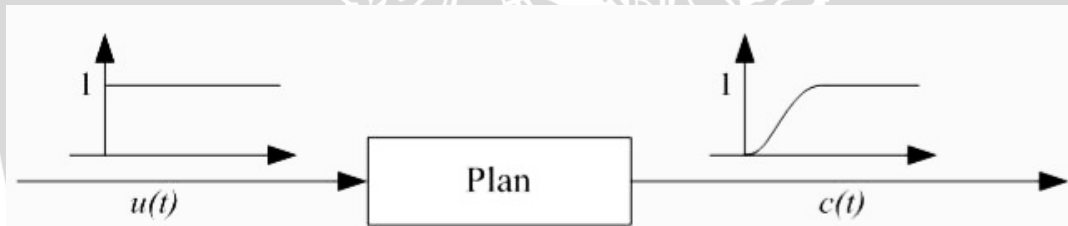


Gambar 2.13 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot

Sumber: Ogata, K., 1997

a). Metode Pertama

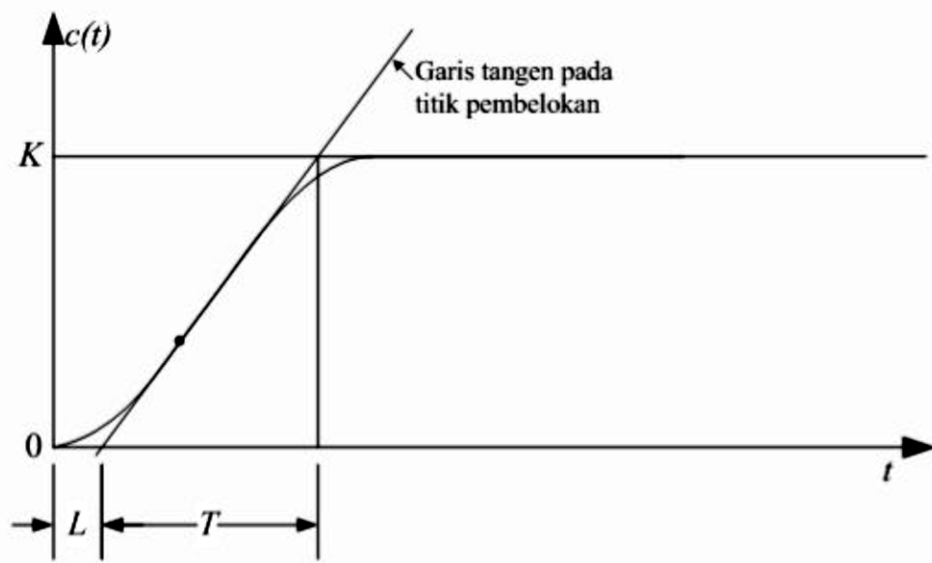
Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan pada Gambar 2.14



Gambar 2.14 Respon Plant Terhadap Masukan Berupa Unit Step

Sumber: Ogata, K. 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti dalam Gambar 2.15 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



Gambar 2.15 Kurva Respon yang Berbentuk S

Sumber: Ogata, K. 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.15 Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut :

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1}$$

Sumber: Ogata, K. 1997

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1. (Ogata, K., 1997).

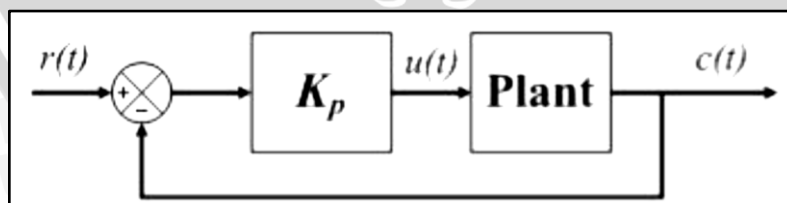
Tabel 2.1. Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan

Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5 L$

Sumber: Ogata, K. 1997

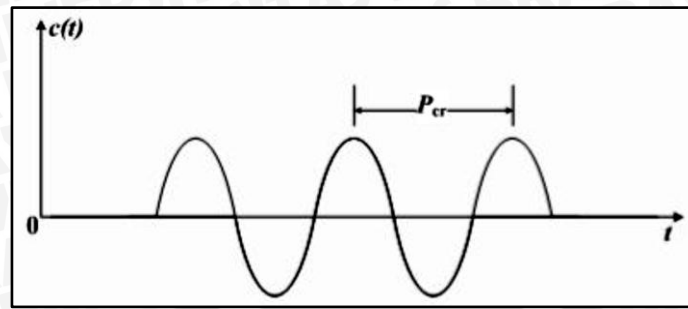
b). Metode Kedua

Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat $T_i = \infty$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.16. dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.17. Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i , T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.1 (Ogata, K., 1997).



Gambar 2.16 Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Ogata, K., 1997



Gambar 2.17 Osilasi Berkesinambungan dengan Periode P_{cr}

Sumber: Ogata, K., 1997

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain K_{cr} dan Critical Period P_{cr}

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Sumber: Ogata, K., 1997

2.8 Mikrokontroler

Mikrokontroler populer yang pertama dibuat oleh Intel pada tahun 1976, yaitu mikrokontroler 8-bit Intel 8748. Mikrokontroler tersebut adalah bagian dari keluarga mikrokontroler MCS-48. Sebelumnya, Texas instruments telah memasarkan mikrokontroler 4-bit pertama yaitu TMS 1000 pada tahun 1974. TMS 1000 yang mulai dibuat sejak 1971 adalah mikrokomputer dalam sebuah *chip*, lengkap dengan RAM dan ROM.

Pengendali mikro (*microcontroller*) adalah sistem mikroprosesor lengkap yang terkandung didalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O.

Berbeda dengan CPU serba-guna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah *pin* yang lebih sedikit.

Sebuah *chip* mikrokontroler umumnya memiliki fitur :

- a) *central processing unit* - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
- b) *input/output* antarmuka jaringan seperti *port serial* (UART)
- c) antarmuka komunikasi serial lain seperti *I²C*, *Serial Peripheral Interface* and *Controller Area Network* untuk sambungan sistem
- d) periferal seperti *timer* dan *watchdog*
- e) RAM untuk penyimpanan data
- f) ROM, EPROM, EEPROM atau *Flash memory* untuk menyimpan program komputer
- g) pembangkit *clock* - biasanya berupa resonator rangkaian RC
- h) pengubah *analog* ke *digital*

2.8.1 Mikrokontroler ATmega328

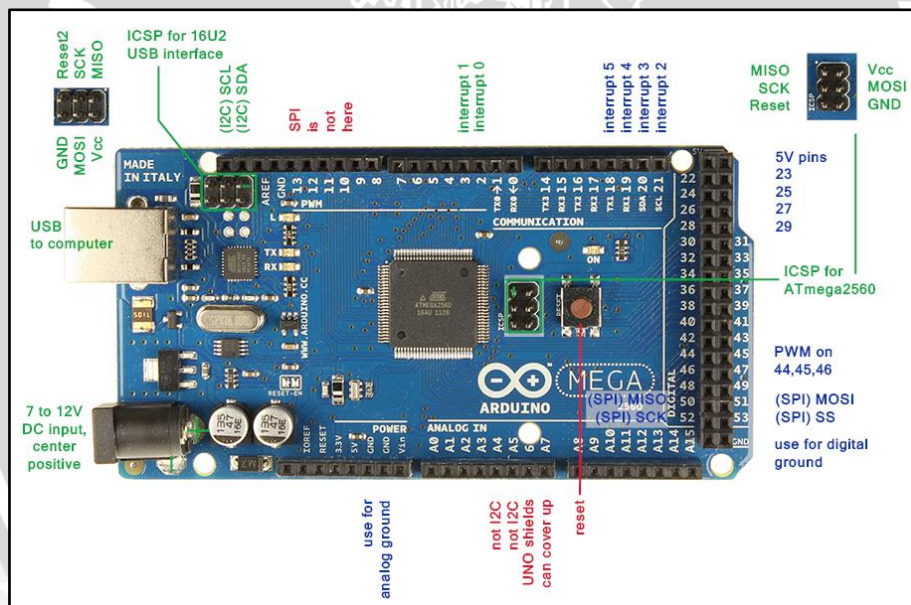
Atmel ATmega328 adalah mikrokontroler CMOS 8-bit berdaya rendah berbasis AVR yang arsitektur RISCnya telah ditingkatkan. Hampir semua instruksi dieksekusi dalam satu siklus *clock*, mempunyai *throughput* mendekati 1 MIPS per MHz membuat desain sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

Atmel ATmega328 memiliki beberapa fitur antara lain 8Kbytes *In-system Programmable Flash with Read-While-Write*, 1K bytes EEPROM, 2K bytes SRAM, 23 jalur I/O untuk tujuan umum, 32 *working registers* untuk tujuan umum, tiga *timer/counter* yang fleksibel dengan *compare mode*, internal dan external *interrupt*, sebuah serial *programmable USART*, sebuah *byte-oriented 2-wire Serial Interface*, sebuah *port SPI serial*, sebuah 6-channel 10-bit ADC, sebuah *Watchdog Timer* yang *programmable* dengan internal osilator.

2.9 Arduino Mega

Arduino Mega adalah sebuah board mikrokontroler berbasis Atmega 2560. Arduino Mega mempunyai 54 pin *digital input / output* (15 di antaranya dapat digunakan sebagai *output* PWM), 16 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino Mega memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya pada sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah *adaptor* AC ke DC atau menggunakan baterai untuk memulainya.

Mega berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter *USB-to-serial* berbeda dengan *board* sebelumnya yang menggunakan *chip* FTDI *driver* *USB-to-serial*. Yang ditunjukkan pada Gambar 2.18.



Gambar 2. 18 Board Arduino Mega

Sumber : electroschematics.com

BAB III METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem pengendalian menggunakan PID yang bertujuan agar dapat menampilkan informasi sistem yang sesuai dengan yang direncanakan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut :

3.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditentukan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Dengan perencanaan sebagai berikut.

1. Baling-baling adalah alat yang mengubah gerak putar menjadi daya dorong.
2. Menggunakan sensor Potensiometer.
3. Motor BLDC sebagai pemutar baling-baling.
4. Motor *brushless* memiliki sebuah ESC (*Electronic Speed Control*) yang berfungsi sebagai pengatur kecepatan motor.
5. Kontrol yang digunakan adalah Kontrol PID
6. Menggunakan Arduino Mega sebagai tempat pemrograman Kontrol PID.

3.2 Studi Literatur

Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi :

1. Motor *BLDC*
2. Sensor *Potensiometer*
3. Kontrol PID
4. Motor servo
5. Arduino mega
6. Propeller

3.3 Perancangan dan Realisasi Pembuatan Alat

3.3.1 Perancangan Perangkat Keras dan Realisasi Pembuatan Alat

- a. Pembuatan diagram blok
- b. Penentuan dan Perhitungan komponen yang akan digunakan dalam perancangan alat
- c. Merakit perangkat keras (*hardware*) untuk masing-masing blok.

3.3.2 Perancangan dan Perhitungan Komponen yang akan Digunakan

Setelah merancang perangkat keras, maka langkah selanjutnya adalah merancang perangkat lunak guna mengendalikan dan mengatur kerja daripada alat. Desain dan parameter yang telah dirancang kemudian diterapkan pada Arduino mega dengan menggunakan *software* Arduino ERW 1.0.5.

3.3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dimulai dari pembuatan *flowchart*, kemudian penulisan *listing code* program PID pada *software* Arduino ERW 1.0.5.

3.4 Pengujian Alat

Setelah semua komponen pada alat sudah terealisasi sesuai dengan diagram blok yang telah dirancang dan perangkat lunak pendukung sistem sudah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujiannya adalah sebagai berikut :

1. Pengujian Sensor

Pengujian sensor dilakukan dengan cara mensimulasikan rangkaian sensor dan hasil pemodelan rangkaian sensor. Pengujian ini bertujuan untuk memastikan sensor dan hasil pemodelan sensor dapat bekerja sesuai dengan perancangan dan memberikan analisis terhadap hasil pengujian. Terdapat rangkaian sensor utama yang akan diuji, yaitu sensor Potensiometer sebagai pengukur keseimbangan *frame* pada *bicopter*

2. Pengujian Sinyal Kontrol BLDC

Pengujian sinyal kontrol BLDC ini bertujuan untuk melihat bagaimana bentuk sinyal saat berada pada posisi sudut yang telah ditentukan untuk menggerakkan aktuator serta melihat tegangan yang dikeluarkan untuk setiap perubahan sudut propeller pada motor *BLDC*.

3. Pegujian Sinyal Kontrol Motor *Servo*

Pengujian sinyal kontrol motor *servo* ini bertujuan untuk melihat bagaimana bentuk sinyal saat berada pada posisi sudut yang telah ditentukan untuk menggerakkan *propeller* serta melihat tegangan yang dikeluarkan untuk setiap perubahan sudut motor *servo*.

4. Pengujian Motor *Servo* terhadap *Motor BLDC*

Pengujian ini dilakukan untuk mengetahui pengaruh perubahan pergerakan motor *servo* terhadap kecepatan putaran pada *Motor BLDC*, sehingga dapat menentukan besarnya perubahan kecepatan putaran tiap perubahan derajatnya

5. Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk mengetahui bagaimana kinerja sistem secara keseluruhan dan mengamati respons kontroler PID terhadap *setpoint* dan ketika mendapatkan gangguan. Dengan adanya gangguan maka dapat melihat sistem pengendalian telah berjalan dengan baik atau tidak.

3.5 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan hasil data yang diperoleh dari pengujian keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan, maka sistem kendali tersebut telah berhasil memenuhi harapan dan untuk penyempurnaan mendalam perlu adanya pengembangan lebih lanjut.



BAB IV PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini membahas mengenai perancangan dan pembuatan alat bantu desain *frame bicopter* secara statis satu *frame* dengan Kontroler Proporsional Integral Diferensial. Perancangan perangkat tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak. Sedangkan pembuatan bertujuan untuk menghasilkan semua perangkat pendukung maupun alat secara keseluruhan.

4.1 Perancangan Sistem

Perancangan alat ini dilakukan bertahap dalam bentuk diagram blok sehingga memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan sistem. Perancangan ini terdiri atas :

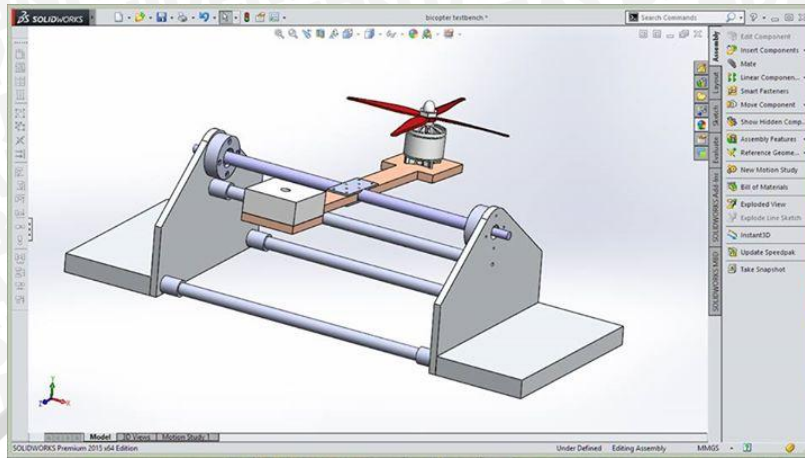
1. Perancangan perangkat keras (sensor *Potensiometer*, *propeller* (balin-baling), dan modul Arduino mega)
2. Perancangan perangkat lunak (perancangan algoritma kontrol PID pada *software* Arduino mega)

4.1.1 Perancangan Sistem Mekanik

Untuk mendesain *frame bicopter* secara lengkap diperlukan alat uji satu *frame*. Alat bantu desain *frame bicopter* merupakan model satu *frame* dengan satu aktuator motor dan propeller. Sisi lainnya merupakan beban uji yang dapat diubah-ubah beratnya. Dengan menggabungkan kontroler PID, dapat diuji kestabilan yang dihasilkan dengan mengubah-ubah parameter, panjang *frame*, spesifikasi motor dan propeller dengan konstanta PID tertentu. Parameter yang diperoleh dari alat uji untuk membantu desain *bicopter* akan dapat digunakan sebagai dasar desain *bicopter* secara keseluruhan. Spesifikasi mekanik *frame* adalah sebagai berikut dan gambar konstruksi perancangan dapat dilihat pada Gambar 4.1 :

1. Panjang : 95 cm
2. Lebar : 40 cm
3. Tinggi : 34 cm

Konstruksi alat yang dirancang dapat dilihat dalam Gambar 4.1.



Gambar 4.1 Gambar skema alat uji satu *frame*



Gambar 4.2 Foto alat uji satu *frame*

Keterangan dari Gambar 4.2 :

1. Motor Servo
2. Motor BLDC 1200 KV
3. Beban 300 gram

Frame menyerupai jungkat-jungkit dengan kedua sisi sama panjang, satu sisi terdapat *propeller* dan sisi lainnya merupakan beban uji yang dapat diubah beratnya dengan cara menggeser ke arah dalam untuk menambah berat beban uji dan ke arah luar untuk mengurangi beban uji. Selanjutnya kecepatan putaran motor diatur secara *konstan* dan *pitch* (*sudut*) *propeller* diatur dengan kontroler PID agar memiliki *pitch* yang diinginkan. Umpan balik yang digunakan berupa sensor *potensiometer* yang diletakkan pada tepat ditengah antara kedua sisi.

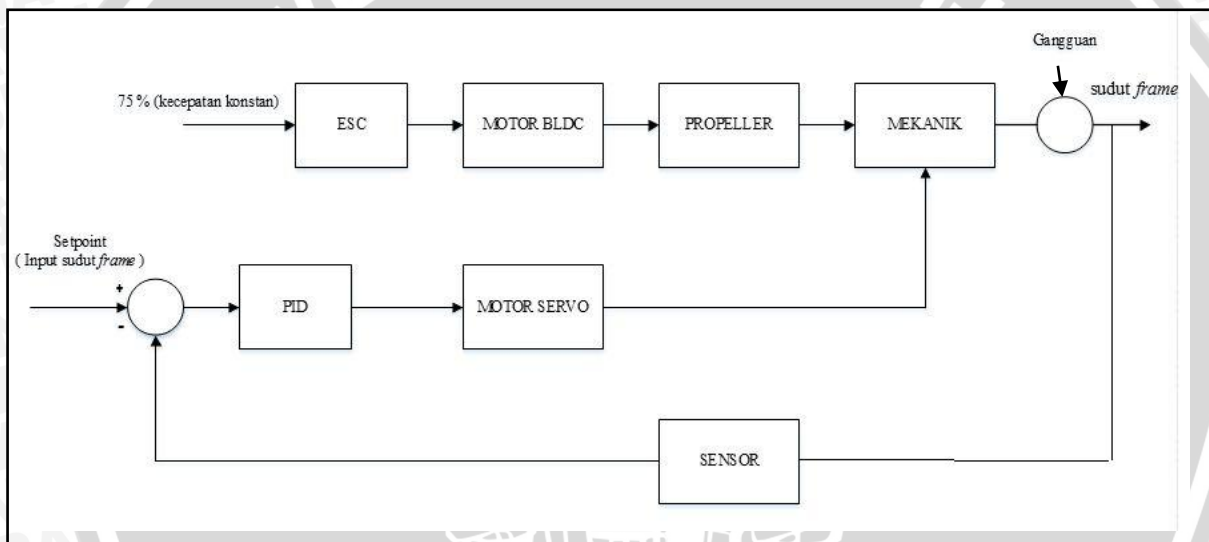
4.1.2 Perancangan Sistem Kontroler

Spesifikasi alat yang dirancang adalah sebagai berikut :

1. Arduino mega 2560 dan *software* Arduino ERW 1.0.5 untuk pengembangan kontroler Proporsional Integral Derferensial (PID).
2. Sensor yang digunakan berupa sensor *potensiometer* berfungsi sebagai *feedback* pengontrolan posisi.

4.2 Diagram Blok Sistem

Dalam skripsi ini dibuat diagram blok agar dalam mengerjakan dapat dilakukan sesuai dengan rancangan sistem. adapun diagram blok tersebut dapat dilihat pada Gambar 4.3



Gambar 4. 3 Diagram Blok Sistem

Keterangan dari diagram blok pada Gambar.4.3 :

- Masukan / *setpoint* berupa sudut (*pitch*) diberikan melalui *program* pada Arduino mega.
- Kemudian *input* diolah dan menghasilkan sinyal kontrol berupa PWM yang kemudian akan mejadi masukan motor servo untuk mengerjakan sudut propeller pada mekanik.
- Motor BLDC menggunakan ESC untuk mengatur jumlah arus yang diperlukan oleh motor yang dicatu oleh *power suply*. Sedangkan untuk mengatur kecepatan motor BLDC, ESC diberi *input* sebesar 75% dari kecepatan *maximum* karena putaran Motor BLDC diatur untuk konstan.

- Putaran motor BLDC tadi kemudian menggerakkan propeller agar dapat berputar sesuai *input* dari ESC sebesar 75 % dari kecepatan *maximum*. sekitar 10800rpm dari maksimal 14400rpm.
- Keluaran dari putaran *propeller* sebagai *input* dari mekanik. Dalam mekanik ini propeller berputar secara konstan tetapi sudut atau *pitch* yang berubah sesuai *output* dari motor *servo* dikarenakan adanya *distrub* (gangguan) yang berubah – ubah.
- Keluaran dari mekanik kemudian dibaca oleh sensor potensiometer. Keluaran dari sensor yang berupa analog kemudian diubah menjadi digital oleh *converter* ADC.
- Hasil akhir pembacaan sensor kemudian dikurangkan dengan *input* / *setpoint* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi.

4.2.1 Alasan menggunakan *input* kecepatan sebesar 75% (10800rpm)

Dikarenakan pada hasil percobaan perancangan keseimbangan gerak *pitch* pada *bicopter* secara statis dengan metode mengubah-ubah kecepatan motor BLDC yang dilakukan oleh Dandy Muhammad, diperoleh keadaan setimbang pada kecepatan 7300rpm, yang terdapat pada Tabel 4.1. Oleh karena itu kecepatan diatur tidak maksimal tetapi diberi sebesar 10800rpm (75%) dari maksimal 14400rpm, dikarenakan pada skripsi ini hanya mengatur *pitch propeller* dengan kecepatan yang telah ditentukan agar mendapat posisi setimbang.

Tabel 4.1 kecepatan terhadap sudut *frame*

PWM	rpm	Sudut	130	7341.18	0
28	1581.18	-30	147	8301.18	5
45	2541.18	-25	164	9261.18	10
62	3501.18	-20	181	10221.18	15
79	4461.18	-15	198	11181.18	20
96	5421.18	-10	215	12141.18	25
113	6381.18	-5	228	12875.29	30

4.3 Perancangan Perangkat Keras

4.3.1 Propeller dan Motor BLDC

Propeller yang digunakan pada ujung motor BLDC adalah propeller tipe 10 inch. Pemilihan ini didasarkan pada rekomendasi pabrik jika digunakan pada dalam RC (*Radio Control*) aeromodelling. *Propeller* digunakan untuk mengubah sudut (*pitch*) pada alat uji satu *frame* ini sehingga mempengaruhi daya angkat (*trust*) pada *frame*. dan Motor *BLDC* yang digunakan dalam perancangan kali ini berguna sebagai penggerak baling – baling (*propeller*). Motor BLDC ini juga dapat langsung terhubung ke Arduino mega tanpa menggunakan *driver* karena bekerja pada maksimum tegangan masukan 4,8 V dan dengan putaran yang *kontans*. *Propeller* dan motor BLDC ditunjukkan pada Gambar 4.4.



Gambar 4.4 Propeller dan Motor BLDC

Sumber : Dokumen Pribadi

4.3.2 Sensor Potensiometer

Sensor yang digunakan pada perancangan alat ini adalah sensor potensiometer tipe *linear wire-wound* 10 k Ω yang bekerja sebagai pengontrolan posisi. parameter yang terdapat pada kontroler PID yaitu konstanta proporsional (K_p), konstanta integral (K_i) dan konstanta diferensial (K_d) menggunakan metode 1 Ziegler-Nichols. Sensor potensiometer ditunjukkan pada Gambar 4.5.

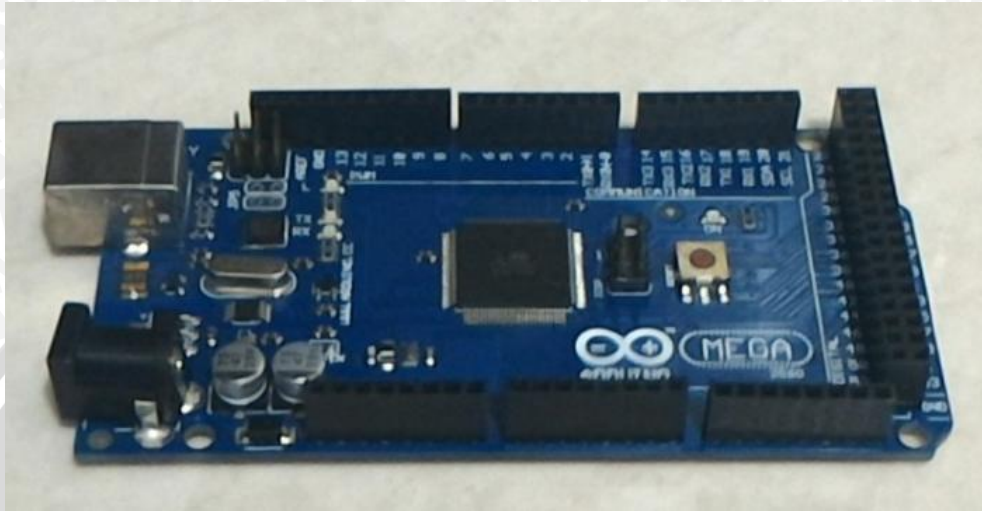


Gambar 4.5 Sensor Potensiometer

Sumber : Dokumen Pribadi

4.3.3 Modul Arduino Mega

Pada perancangan alat ini Arduino mega berbasis mikrokontroler Atmega 2560 digunakan sebagai pusat pengolah utama dalam melakukan proses pengendalian. Modul Arduino Mega ditunjukkan pada Gambar 4.6.



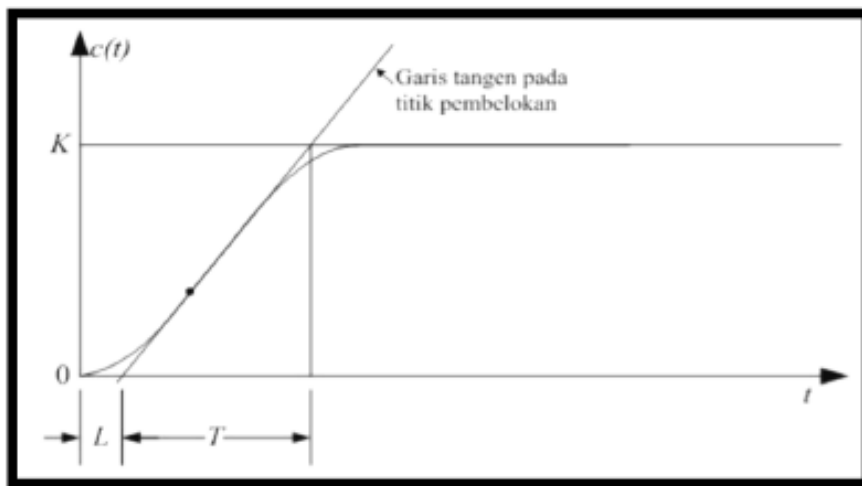
Gambar 4.6 arduino mega

Sumber : electroschematics.com

4.4 Perancangan Kontrol PID

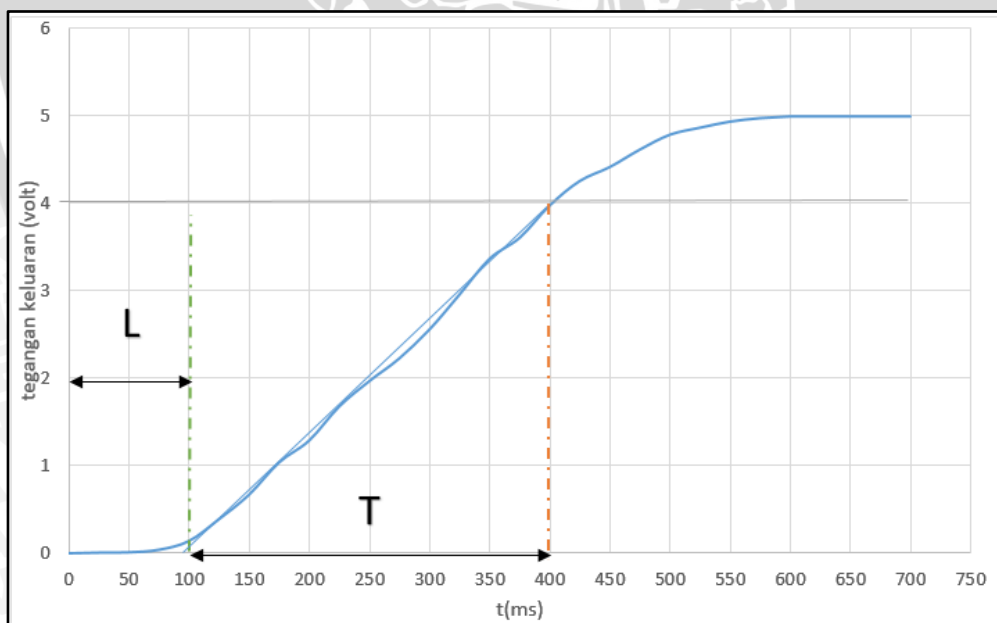
4.4.1 Penentuan Nilai Penguatan Kontroler

Untuk menentukan penguatan kontroler digunakan metode Ziegler dan Nichols yang mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivative T_d berdasarkan karakteristik respon transien dari plant yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan *plan* (Ogata, K., 1997). Pada pengerjaan skripsi ini digunakanlah metode *Ziegler Nichols* yang pertama sehingga mendapatkan sinyal kurve S seperti pada Gambar 4.7



Gambar 4.7 Kurva Respon yang Berbentuk S (Ogata, K., 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$. Berikut kurva hasil perancangan kontroler menggunakan metode *Ziegler Nichols* yang ditunjukkan pada Gambar 4.8.



Gambar 4.8 Karakteristik Respon *Dimming* (dokumen pribadi)

Langkah menentukan *tuning* parameter dengan metode 1 *Ziegler-Nichols* adalah sebagai berikut :

1. Menarik garis *tangent* pada titik infleksi pada grafik karakteristik *plant* seperti pada Gambar 4.8
2. Menentukan perpotongan garis *tangent* terhadap sumbu waktu *t* untuk mendapatkan nilai *L*.
3. Menentukan perpotongan garis *tangent* terhadap sumbu *steady* untuk mendapatkan nilai *T*.
4. Setelah mendapatkan nilai *L* dan *T* kemudian menentukan nilai *K_p*, *T_i* dan *T_d* sesuai ketentuan pada Tabel 4.2

Tabel 4.2 Aturan Metode 1 *Ziegler-Nichols* (Ogata, K., 1997)

Tipe Kontrol	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

Diperoleh nilai $L = 100\text{ms}$ dan $T = 300\text{ms}$. Dan diperoleh nilai K_p , K_i , K_d :

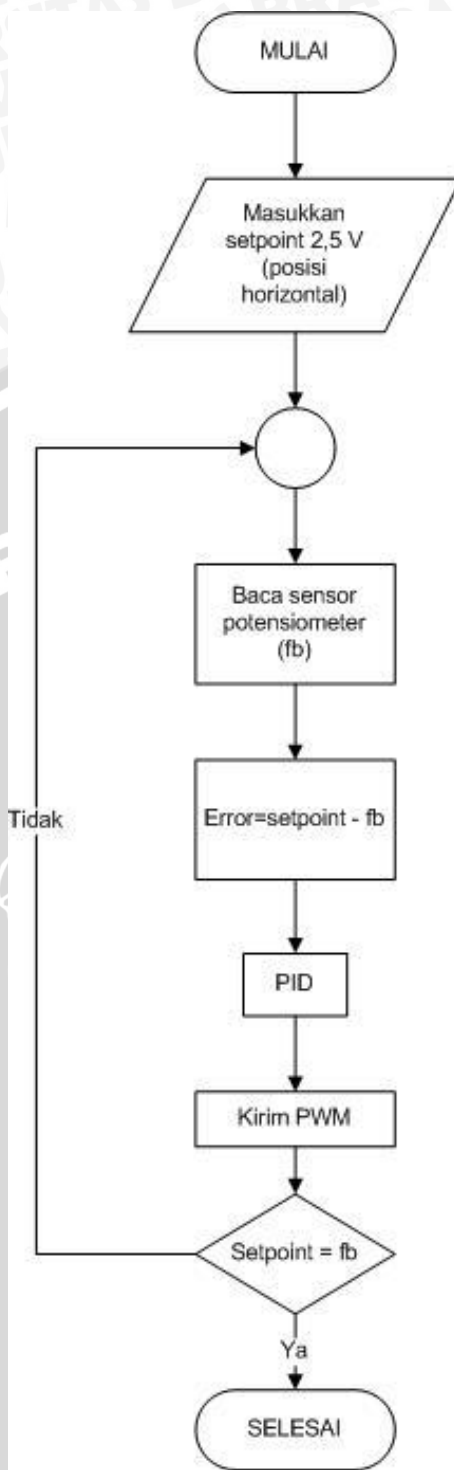
$$K_p = 1.2 \times \frac{T}{L} = 1.2 \times \frac{0,30}{0,10} = 3,6$$

$$K_i = \frac{K_p}{T_i} = \frac{3,6}{2 \times L} = \frac{3,6}{2 \times 0,1} = 18$$

$$K_d = K_p \times T_d = 3,6 \times (0.5 \times 0,1) = 3,6 \times 0,05 = 0,24$$

4.5 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada pengendalian ini menggunakan bahasa pemrograman C++ dengan menggunakan *software* Arduino ERW 1.0.5. *Tuning* kontroler PID adalah dengan memasukkan nilai K_p , K_i , dan K_d dari hasil perhitungan metode *Ziegler-Nichols*. *Flowchart* perancangan perangkat lunak dapat dilihat pada Gambar 4.9.



Gambar 4.9 Flowchart Perangkat Lunak



BAB V PENGUJIAN DAN ANALISIS

Tujuan pengujian sistem ini adalah untuk menguji apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menguji unjuk kerja untuk mempermudah analisis pada sistem apakah sesuai dengan perancangan. Adapun langkah – langkah pengujian yang dilakukan adalah :

1. Pengujian sensor potensiometer
2. Pengujian aktuator motor dc *brushless*
3. Pengujian keseluruhan sistem

5.1 Pengujian Sensor

a. Tujuan

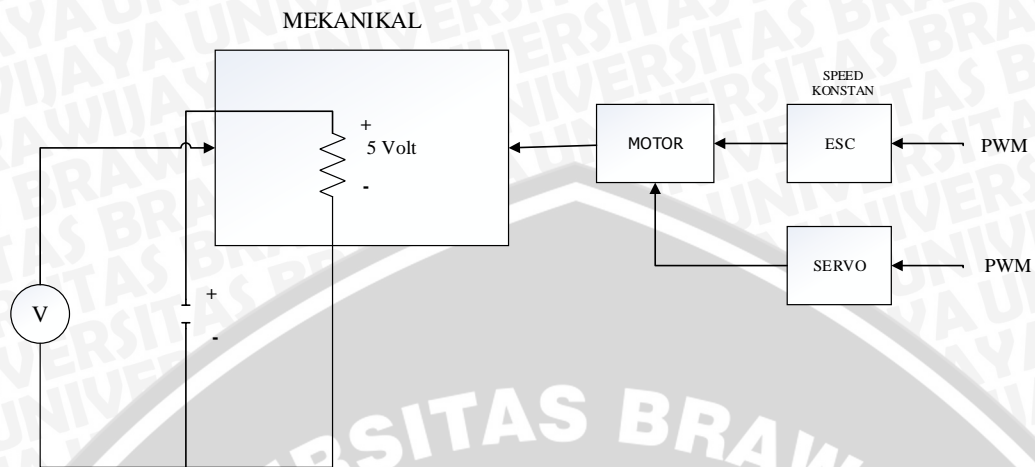
Menguji tegangan keluaran sensor *potensiometer* elevasi *bicopter*.

b. Peralatan yang digunakan

- Sensor *potensiometer*
- Voltmeter
- Mekanikal alat uji *bicopter*
- Baterai 12 VDC ESC catu daya 5 VDC

c. Langkah pengujian

1. Merangkai peralatan seperti pada Gambar 5.1
2. Berikan beban penyeimbang 300 gram
3. Set PWM *generator* = \emptyset
4. Hidupkan catu daya
5. Set elevasi mulai minimal sampai maksimal
6. Ukur tegangan keluaran sensor setiap elevasi yang diberikan.



Gambar 5. 2 Diagram perancangan Pengujian Sensor

d. Hasil Pengujian

Hasil pengujian diperlihatkan dalam Tabel 5.1 :

Tabel 5.1 : Hasil Pengujian Sensor dengan beban

PWM	Pitch propeller	Sudut Frame	Tegangan Keluaran (Volt)				
				130	15,29 ⁰	0 ⁰	2,50
				147	17,29 ⁰	5 ⁰	2,92
28	3,29 ⁰	-30 ⁰	0	164	19,29 ⁰	10 ⁰	3,33
45	5,29 ⁰	-25 ⁰	0,42	181	21,29 ⁰	15 ⁰	3,75
62	7,29 ⁰	-20 ⁰	0,83	198	23,29 ⁰	20 ⁰	4,17
79	9,29 ⁰	-15 ⁰	1,25	215	25,29 ⁰	25 ⁰	4,58
96	11,29 ⁰	-10 ⁰	1,67	228	26,82 ⁰	30 ⁰	5,00
113	13,29 ⁰	-5 ⁰	2,08				

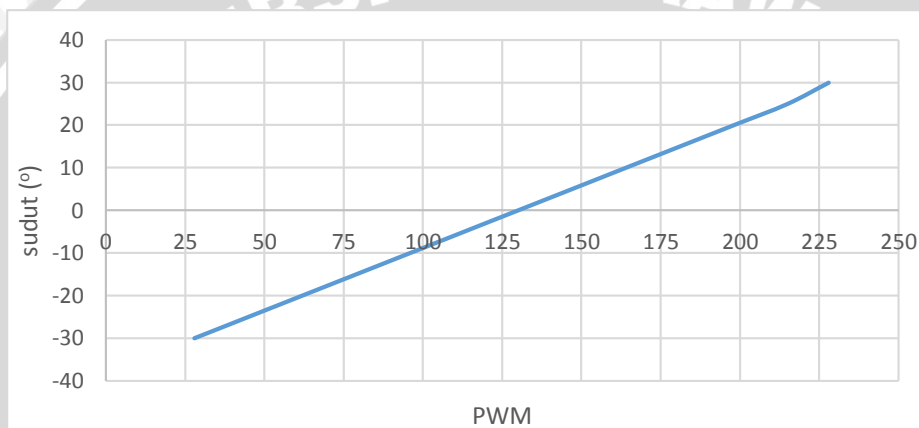
Hasil pengujian berupa data PWM, Sudut *frame*, dan Tegangan keluaran sensor *potensiometer* ditunjukkan pada Tabel 5.1. Selanjutnya PWM 0 – 255 ekuivalen dengan sudut *pitch* 0 - 30°. Sebagai contoh untuk PWM 130 pada Tabel 5.1 maka diperoleh *pitch propeller* sebesar :

$$\text{Pitch propeller} = \frac{\text{PWM}}{\text{PWM max}} \times \text{pitch propeller max}$$

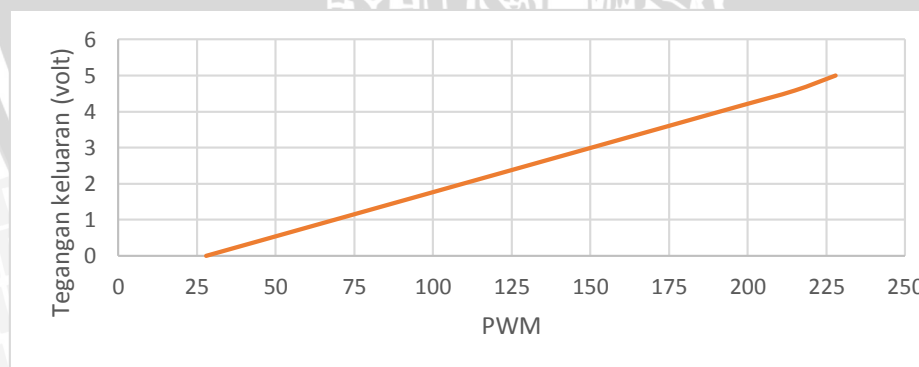
$$\text{Pitch propeller} = \frac{130}{255} \times 30^\circ = 15,29^\circ$$

Dengan cara yang sama data *pitch propeller* diperoleh pada Tabel 5.1.

Secara grafik hasil pengujian ditunjukkan dalam Gambar 5.2 dan 5.2.2 :



Gambar 5.2 Grafik perbandingan antara PWM terhadap sudut



Gambar 5.2.2 Grafik perbandingan antara PWM terhadap tegangan keluaran

Dari hasil pengujian yang dilakukan, sensor dapat bekerja dengan maksimal dan terlihat kelinieran yang baik sehingga ideal untuk digunakan sebagai sudut kemiringan pada *frame*.

5.2 Pengujian respon aktuator Motor DC *Brushless*

a. Tujuan

Menguji respon plan secara *open loop* untuk mendapatkan kurva S.

b. Peralatan yang digunakan

- ESC Catu Daya 5V DC.
- Pembangkit PWM
- Mekanik alat uji *bicopter*
- PWM Generator
- Data *logger*

c. Langkah pengujian

1. Merangkai peralatan seperti Gambar 5.1.
2. Berikan beban penyeimbang 300 gram.
3. Hidupkan catu daya.
4. Berikan PWM maksimal seketika.
5. Rekam respon menggunakan *logger*.

d. Hasil Pengujian

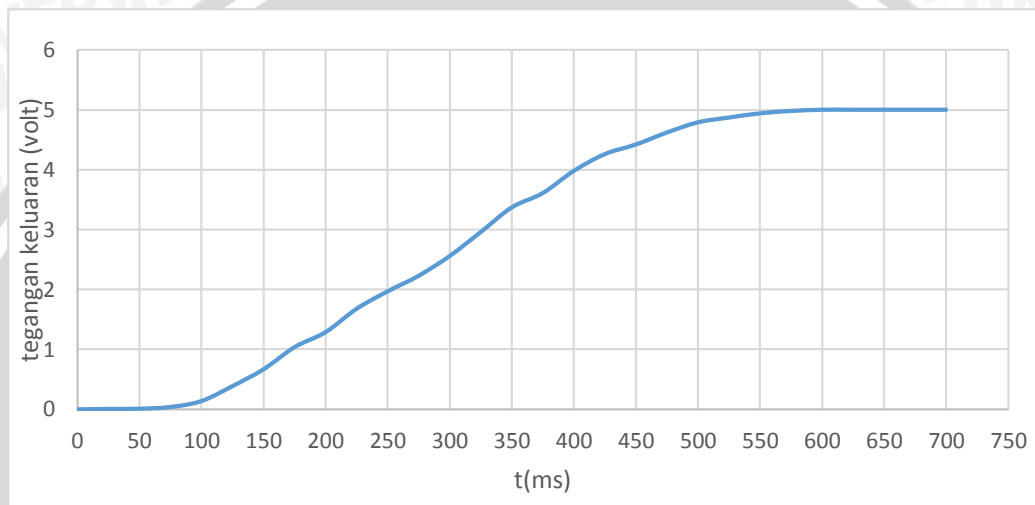
Setelah melakukan pengujian sesuai dengan langkah pengujian , didapatkan respon aktuator ditunjukkan dalam Tabel 5.2.

Tabel 5. 2 respon aktuator

t(ms)	Tegangan Keluaran (Volt)
0	0
25	0,007
50	0,011
75	0,04
100	0,14
125	0,39
150	0,67
175	1,04
200	1,29
225	1,68
250	1,97
275	2,23
300	2,56
325	2,96
350	3,37
375	3,61
400	3,98
425	4,26
450	4,42
475	4,62
500	4,79
525	4,87
550	4,94
575	4,98
600	5,00
625	5,00
650	5,00
675	5,00
700	5,00

Dari Tabel 5.2 dapat terlihat bahwa pada saat motor dihidupkan dengan kecepatan sebesar 75 % dari kecepatan *maximum* tetapi *pitch propeller* masih 0° , *propeller* masih belum menghasilkan daya dorong (*trush*). sehingga ketika diberi *input pitch propeller* maksimal seketika sebesar 30° atau PWM sebesar 255, sistem mampu mencapai tegangan 5V atau sama dengan kemiringan maksimal pada saat 550 ms dan dapat dilihat hasil respon seperti Gambar 5.3.

Hasil pengujian respon aktuator secara grafik ditunjukkan dalam Gambar 5.3 :

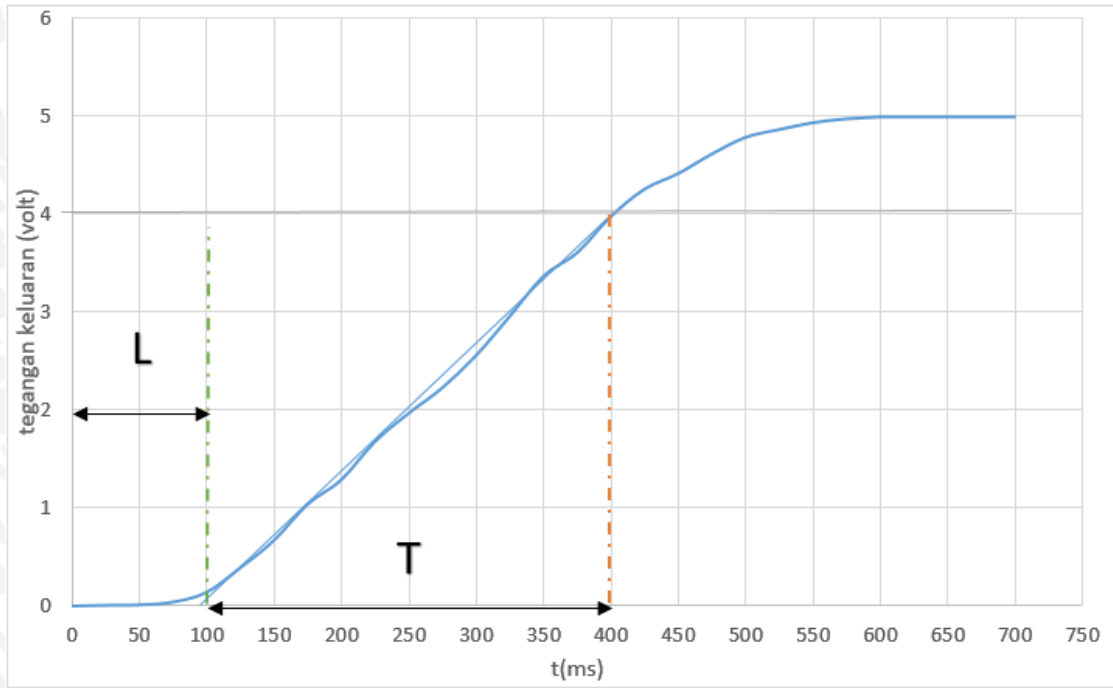


Gambar 5.3 : respon aktuator dalam grafik

5.3 Analisis

Dari hasil pengujian yang diperlihatkan dalam Gambar 5.3 berupa kurva S akan digunakan menentukan parameter tuning PID dengan metode 1 Ziegler-Nichols seperti yang diperlihatkan dalam Gambar 5.3 Adapun langkah-langkah yang dilakukan sebagai berikut :

1. Menarik garis tangent pada titik infleksi grafik karakteristik open loop seperti yang diperlihatkan dalam Gambar 5.3.1
2. Menentukan perpotongan garis tangent terhadap sumbu waktu t untuk mendapatkan L
3. Menentukan perpotongan garis tangent terhadap sumbu steady untuk mendapatkan nilai
4. Nilai L dan T digunakan untuk menentukan nilai K_p T_i dan T_d sesuai dengan Tabel 5.3.



Gambar 5.3.1 : respon aktuator dalam grafik

Tabel 5.3 Aturan Metode 1 Ziegler-Nichols (Ogata K., 1997)

Tipe Kontrol	K_p	τ_i	τ_d
P	$\frac{\tau}{L}$	∞	0
PI	$0,9 \frac{\tau}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{\tau}{L}$	$2L$	$0,5L$

Dari Gambar 5.3.1 diperoleh besarnya $L = 100\text{ms}$ dan $T = 400 - 100 = 300\text{ms}$.

Dengan demikian parameter kontroler diperoleh sebagai berikut :

$$K_p = 1,2 \frac{\tau}{L} = 1,2 \times \frac{0,30\text{s}}{0,10\text{s}} = 3,6$$

$$\tau_i = 2L = 2 \times 0,1\text{s} = 0,2$$

$$\tau_d = 0,5L = 0,5 \times 0,1\text{s} = 0,05$$

Selanjutnya akan diperoleh K_i dan K_d sebagai berikut :

$$K_i = \frac{K_p}{\tau_i} = \frac{3,6}{0,2} = 18 \quad K_d = K_p \times \tau_d = 3,6 \times 0,05 = 0,18$$

5.4 Pengujian sistem keseluruhan.

a. Tujuan

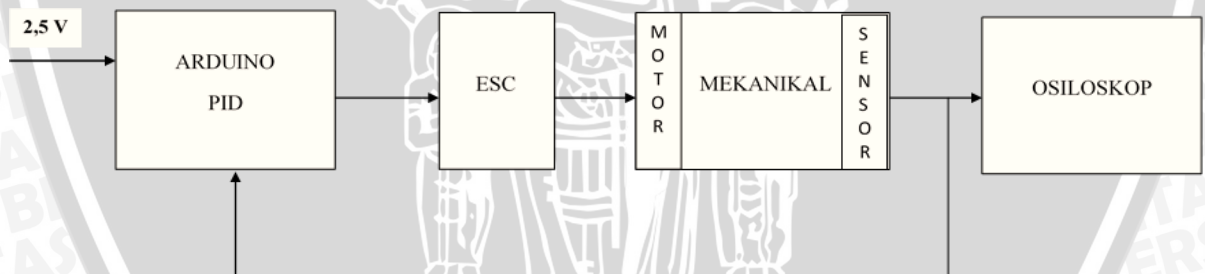
Pengujian ini bertujuan untuk mengetahui respon sistem secara closed loop dengan memberikan gangguan sementara.

b. Peralatan yang digunakan

- Catu Daya 5VDC.
- Mikrokontroler Arduino mega
- Komputer
- *Digital Oscilloscope* Velleman PCSU1000 dan *software* PC Lab 2000SE.
- Program dan *software* Arduino.

c. Langkah Pengujian

1. Merangkai peralatan seperti Gambar 5.4.
2. Mengunduh program pengatur sudut *servo* pada *software* Arduino ERW 1.0.5.
3. Menjalankan *software* PC Lab 2000SE.
4. Memilih mode *osiloskop* lalu pilih *Run* untuk menjalankan *osiloskop*.
5. Mengamati sinyal kontrol dan parameter motor *servo* pada *osiloskop*.



Gambar 5. 4 Diagram Perancangan Pengujian keseluruhan sistem

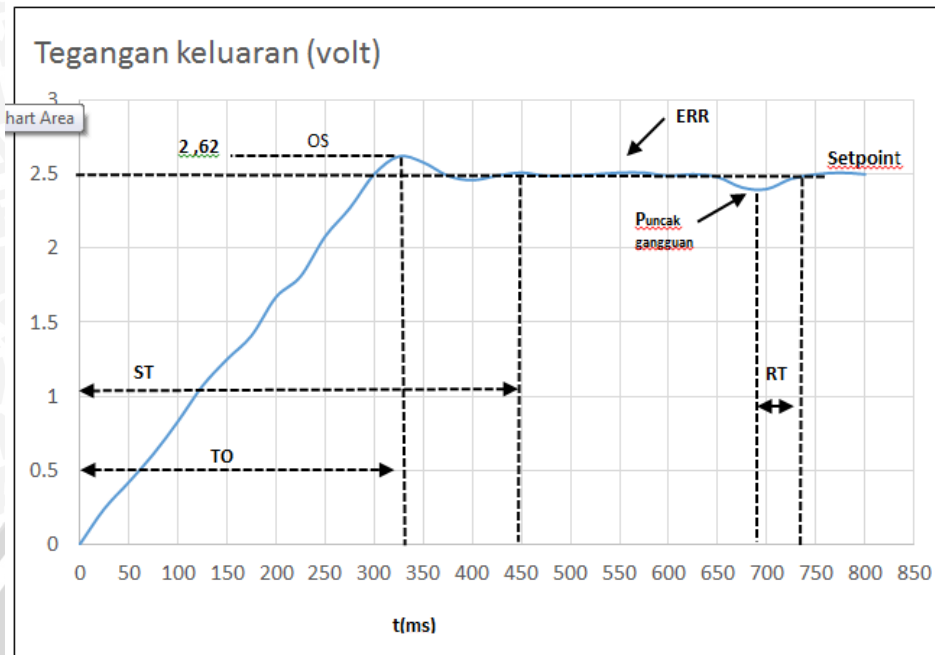
d. Hasil Pengujian

Setelah melakukan pengujian sesuai dengan langkah pengujian, didapatkan data yang ditunjukkan dalam Tabel 5.4 :

Tabel 5. 4 Hasil Pengujian keseluruhan

t (ms)	Tegangan keluaran (volt)		
		400	2,46
0	0	425	2,49
25	0,24	450	2,51
50	0,42	475	2,49
75	0,61	500	2,49
100	0,83	525	2,50
125	1,07	550	2,51
150	1,25	575	2,51
175	1,41	600	2,49
200	1,67	625	2,50
225	1,81	650	2,48
250	2,08	675	2,41
275	2,27	700	2,40
300	2,50	725	2,47
325	2,62	750	2,50
350	2,58	775	2,51
375	2,49	800	2,50

Secara grafik hasil pengujian di perlihatkan dalam Gambar 5.5.



Gambar 5.5 Grafik hasil pengujian keseluruhan

Dari hasil pengujian secara *closed loop* setelah diberi beban sebesar 300gr sistem mampu mencapai steady state pada waktu 450ms yang diperlihatkan dalam Gambar 5.5, kemudian diberi gangguan dengan cara menggeser beban secara *random* (acak). Sehingga menghasilkan *error* sebesar 4% dan respon mampu mengembalikan kembali pada keadaan *steady* (setimbang) dengan *recovery time* sebesar 50ms. Untuk lebih jelasnya parameter pada kerja sistem yang diperoleh sebagai berikut :

1. *Time overshoot* (TO) terjadi pada sekitar 325 ms
2. *Steady State* terjadi pada sekitar 450 ms
3. Besarnya *Overshoot* (OS) sebesar $\frac{2,62-2,5}{2,5} \times 100\% = 4,8\%$
4. Besarnya *Error* (ERR) sebesar $\frac{2,5-2,49}{2,5} \times 100\% = 4\%$
5. *Recovery Time* (RT) setelah adanya gangguan = 50 ms.

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian ini adalah sebagai berikut:

1. Alat yang dirancang sudah sesuai dengan yang diharapkan, dimana berdasarkan data respon sistem yang diperoleh dari pengujian dengan menggunakan metode 1 Ziegler-Nichols, maka parameter PID dapat ditentukan dengan gain $K_p = 3,6$; $K_i = 18$; $K_d = 0,24$.
2. Pada pengujian dengan memberikan gangguan berupa perubahan beban sementara, terjadi perubahan pada kemiringan sudut *frame* dari kondisi setimbang dengan nilai *time overshoot* sebesar 325 ms, waktu mencapai *steady state* yaitu 450 ms, *overshoot* sebesar 4,8% dan *error* sebesar 4%, serta secara keseluruhan sistem dapat kembali pada keadaan *steady* dan mampu memberikan respon sistem yang baik ketika terjadinya gangguan dengan *recovery time* sebesar 50 ms

6.2 Saran

Sebagai pengembangan selanjutnya terdapat beberapa saran-saran sebagai berikut

1. Sistem dapat dikembangkan untuk aksis yang lain dengan metode kontroler yang sama.
2. Dapat menggunakan kontroler selain PID.



DAFTAR PUSTAKA

Chmelai, Pavel. 2011. *Building And Controlling The Quadrocopter*:Online:
http://pernerscontacts.upce.cz/24_2011/Chmelar.pdf.

Gunterus, Frans. 1994. *Falsafah Dasar : Sistem Pengendalian Proses*. Jakarta: Elex Media Komputindo.

Kristianto, Daniel. 2012. *Rancang Bangun Pesawat Mandiri Tanpa Awak Dengan Empat Baling-Baling Penggerak (Autonomous Quadcopter)*. Salatiga: Teknik Elektronika dan Informatika Universitas Kristen Satya Wacana.

Ogata, K. 2002. *Modern Control Engineering*. New Jersey : Prentice-Hall, Inc.

Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1*. Jakarta: Penerbit Erlangga.

Santoso, Ahmad Dzikri. 2014. *Analisis Pengaruh Kecepatan Motor Pada Quadcopter Terhadap Kestabilan Multicopter Aerial Cam*. Teknik Elektro Universitas Pancasila.

Setiawan, Iwan. *Kontrol PID untuk Proses Industri*. Elex Media Komputindo, Jakarta, 2008.

Yusuf, Sam dan Fatoni ,Ali. 2012. *Perancangan Dan Implementasi Kontroler PID Untuk Autonomus Moving Forward Manuever Pada Quadrocopter*. Surabaya: Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS).

Dharmawan, Abe.2009. *Pengendalian Motor Brushless DC dengan Metode PWM Sinusoidal menggunakan ATMEGA 16.*, Jakarta, 2009.



LAMPIRAN 1

FOTO ALAT





Gambar Alat Keseluruhan dengan Beban



Gambar Alat Motor BLDC dan Motor Servo

LAMPIRAN 2

LISTING PROGRAM



```

/*****

```

```

* TUGAS AKHIR

```

```

* JUDUL:

```

```

* PERANCANGAN KESEIMBANGAN GERAK PITCH PADA

```

```

* BICOPTER SECARA STATIS DENGAN METODE MENGUBAH-UBAH

```

```

* PITCH PROPELLER MENGGUNAKAN KONTROLER PID

```

```

* NAMA: ARISKI FADILLAH

```

```

* NIM : 105060307111007

```

```

*****/

```

```

#include <PID_v1.h>

```

```

// Mendefinisikan kanal adc setpoint Input dan Output

```

```

double Setpoint, Input, Output;

```

```

double Kp, Ki, Kd

```

```

// setting parameter PID

```

```

Kp = 3.6;

```

```

Ki = 18;

```

```

Kd = 0.24;

```

```

PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);

```

```

void setup()

```

```

{

```

```

// inisialisasi

```

```

// baca feedback kanal adc 0

```

```

Input = analogRead(0);

```

```

// baca setpoint kanal adc 1

```

```

Setpoint = analogRead(1);

```

```
//turn the PID on
myPID.SetMode(AUTOMATIC);
}

void loop()
{
  // baca feedback
  Input = analogRead(0);
  // baca setpoint
  Setpoint = analogRead(1);
  myPID.Compute();
  // keluarkan hasilnya
  analogWrite(3,Output);
}
```



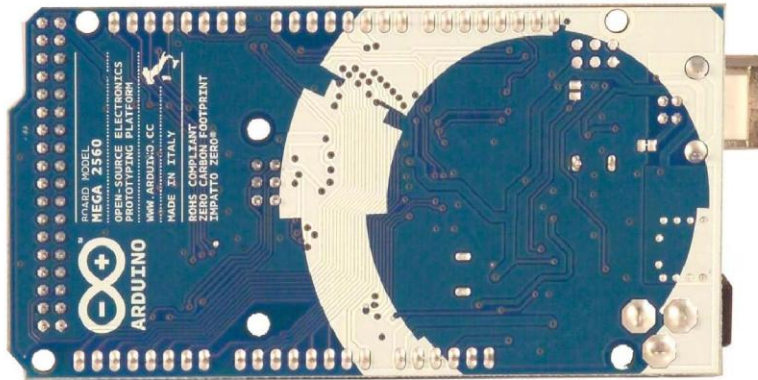
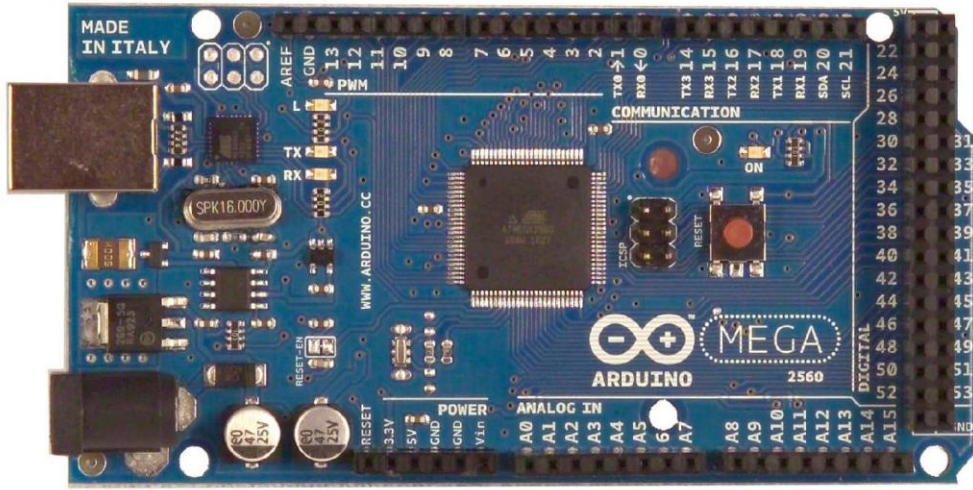


LAMPIRAN 3

DATASHEET



Arduino Mega 2560



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V,

however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

+ VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

+ 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

+ 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

+ GND. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

+ Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

+ External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured

to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

+ **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

+ **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

+ **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

+ **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the

Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function. There are a couple of other pins on the board:

+ **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

+ **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available [in the Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the

Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

Arduino™ Mega 2560 Reference Design

DESIGN DOCUMENT FOR THE ARDUINO MEGA 2560 REFERENCE DESIGN. ALL OTHERS REPRODUCING THIS DESIGN WITHOUT THE WRITTEN PERMISSION OF AT&T INTELLECTUAL PROPERTY ARE PROHIBITED. ANY REPRODUCTION OF THIS DESIGN WITHOUT THE WRITTEN PERMISSION OF AT&T INTELLECTUAL PROPERTY IS PROHIBITED. THE PRODUCT DESIGN IS SUBJECT TO CHANGE WITHOUT NOTICE. SEE THE AT&T WEBSITE FOR THE LATEST VERSION OF THIS DESIGN.

