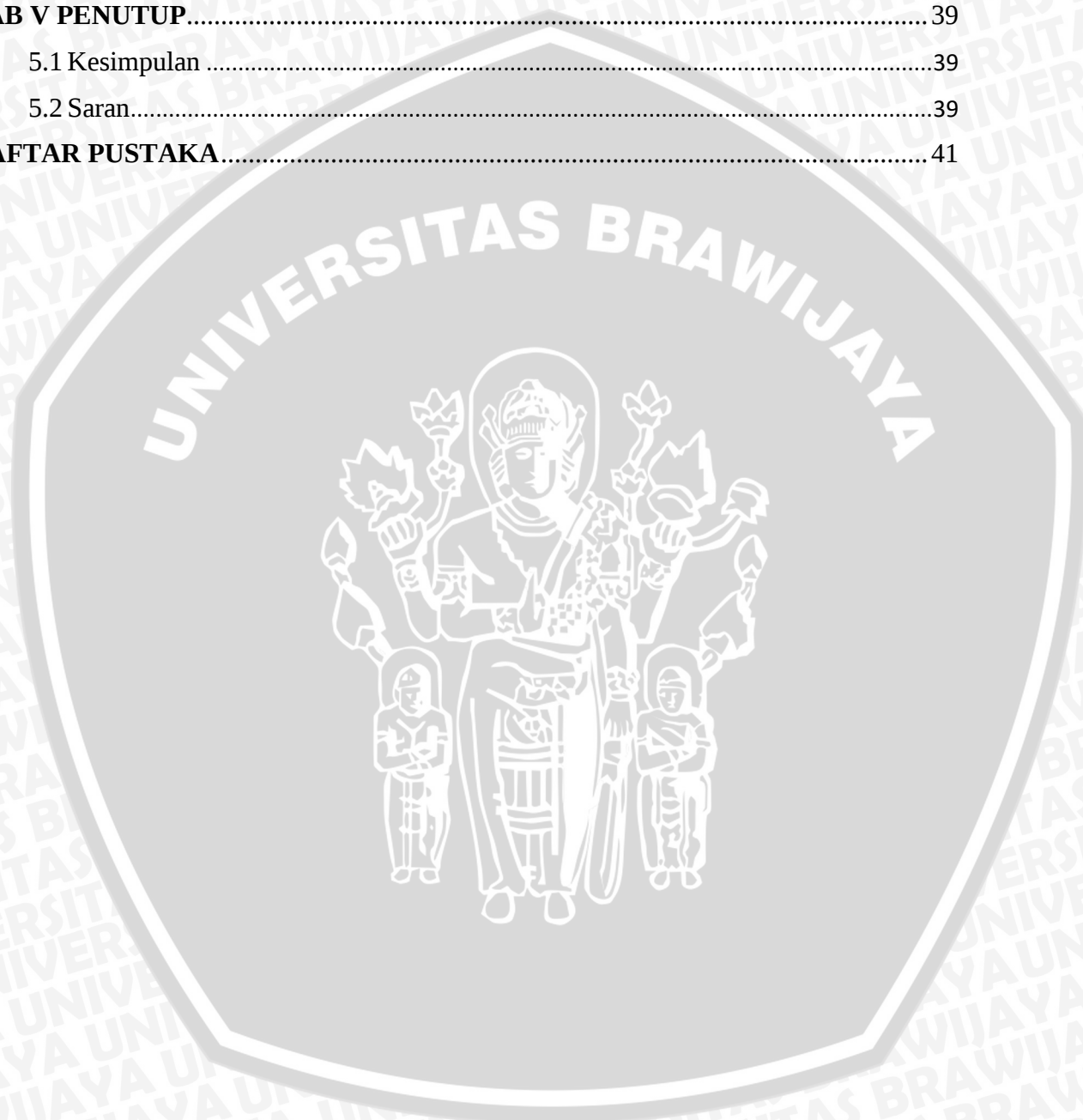


DAFTAR ISI

BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	2
BAB II TINJAUAN PUSTAKA	5
2.1 Ketel uap (<i>Boiler</i>)	5
2.1.1 Ketel dengan pembakaran di dalam (<i>internally fired steam boiler</i>).....	5
2.1.2 Ketel dengan pembakaran di luar (<i>outernally fired steam boiler</i>).	6
2.2 Turbin Uap (<i>Steam Turbinee</i>)	7
2.3 Mikrokontroler Arduino UNO R3.....	8
2.4 Motor DC Servo MG995	10
2.5 PWM (Pulse With Modulation)	11
2.6 Sensor Kecepatan <i>Rotary encoder</i>	11
2.7 Loop Pengendalian	13
2.7.1. Sistem Pengendalian <i>Loop</i> Terbuka (<i>Open Loop Control System</i>).....	13
2.7.2. Sistem Pengendalian <i>Loop</i> Tertutup (<i>Close Loop Control System</i>).....	13
2.8 Kontroler	14
2.8.1. Kontrol Proporsional (P).....	14
2.8.2. Kontroler Integral (I).....	15
2.8.3. Kontroler Proporsional Integral (PI)	15
2.9 Metode Hand Tuning.....	16
BAB III METODOLOGI PENELITIAN	17
3.1 Perancangan Blok Diagram Sistem.....	17
3.2 Pembuatan Perangkat Keras	17
3.3 Spesifikasi Alat	20
3.4 Pin Arduino Uno	22
3.5 <i>Flowchart</i> Perancangan Perangkat Lunak.....	23
3.6 Pengujian <i>Rotary encoder</i>	25
3.7 Pengujian Motor DC Servo	27
3.8 Perancangan Kontroler.....	28

3.9 Pengambilan Kesimpulan dan Saran	23
BAB IV HASIL DAN PEMBAHASAN	25
4.1 Penentuan nilai penguatan (Gain)	32
4.2 Pengujian Keseluruhan Sistem tanpa Gangguan.....	35
4.3 Pengujian Keseluruhan Sistem dengan Gangguan.....	37
BAB V PENUTUP.....	39
5.1 Kesimpulan	39
5.2 Saran.....	39
DAFTAR PUSTAKA.....	41



DAFTAR GAMBAR

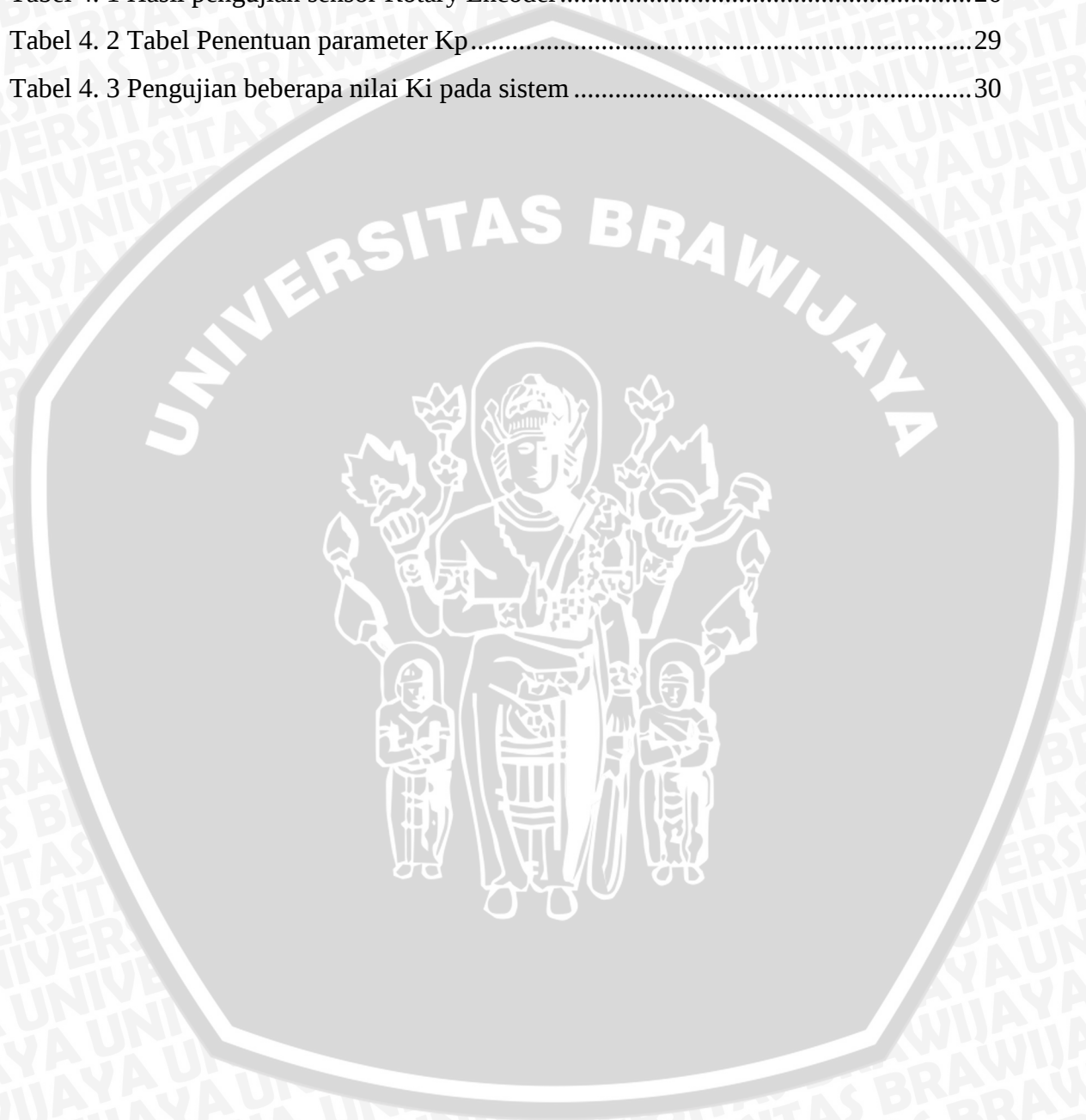
Gambar 2. 1 Internally fired steam boiler	6
Gambar 2. 2 Externally fired steam boiler	7
Gambar 2. 3 Prinsip kerja Turbin Uap	8
Gambar 2. 4 Arduino Uno R3	9
Gambar 2. 5 Motor DC Servo MG995	10
Gambar 2. 6 Pengaturan Sudut Motor Servo	11
Gambar 2. 7 Blok penyusun rotary encoder	12
Gambar 2. 8 Rangkaian Optocoupler	13
Gambar 2. 9 Blok Diagram Sistem Loop Terbuka	13
Gambar 2. 10 Block Diagram Close Loop	14
Gambar 2. 11 Diagram Blok Kontroler Proporsional	15
Gambar 2. 12 Diagram Blok Kontroler Integral	15
Gambar 2. 13 Diagram Blok Kontroler Proporsional Integral	15
Gambar 3. 1 Blok Diagram Sistem	17
Gambar 3. 2 Skema perancangan perangkat keras	18
Gambar 3. 3 Skema perancangan model miniatur Steam Turbine	19
Gambar 3. 4 Rancangan Sistem	19
Gambar 3. 5 Mikrokontroler Arduino Uno	20
Gambar 3. 6 Kompor gas portable	20
Gambar 3. 7 Boiler	21
Gambar 3. 8 Motor DC Servo MG 995	21
Gambar 3. 9 Mini Steam Turbine	21
Gambar 3. 10 Sensor Rotary encoder	22
Gambar 3. 11 Flowchart keseluruhan sistem	23
Gambar 4. 1 Hasil Pengujian Sensor Kecepatan	27
Gambar 4. 2 Blok Diagram pengujian Motor DC Sevo	28
Gambar 4. 3 Hasil percobaan pengujian motor DC Servo	28
Gambar 4. 4 Grafik ketika $K_p=1$, $K_i=0$	29
Gambar 4. 5 Grafik ketika $K_p=1,5$ dan $K_i=0$	30
Gambar 4. 6 Grafik ketika $K_p=2, K_i=0$	30

Gambar 4. 7 Grafik ketika $K_P=1,5$ $K_I=4,25$	31
Gambar 4. 8 Grafik ketika $K_P=1,5$ $K_I=4,5$	31
Gambar 4. 9 Grafik ketika $K_P=1,5$, $K_I=4.75$	32
Gambar 4. 10 Grafik perubahan suhu saat sudut servo 65°	33
Gambar 4. 11 Grafik perubahan suhu saat sudut servo 80°	33
Gambar 4. 12 Grafik perubahan suhu saat sudut servo 95°	34
Gambar 4. 13 Grafik perubahan suhu saat sudut servo 125°	34
Gambar 4. 14 Grafik perubahan Kec Turbin saat sudut servo 65° , 80° , 95° , dan 125°	34
Gambar 4. 15 Grafik perubahan sudut servo dan perubahan kecepatan turbin	35
Gambar 4. 16 Respon Sistem Tanpa Gangguan	36
Gambar 4. 17 Respon Sistem diberi Gangguan	38



DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino UNO R3	9
Tabel 3. 1 Fungsi pin Arduino	23
Tabel 4. 1 Hasil pengujian sensor Rotary Encoder	26
Tabel 4. 2 Tabel Penentuan parameter Kp	29
Tabel 4. 3 Pengujian beberapa nilai Ki pada sistem	30





BAB I PENDAHULUAN

1.1 Latar Belakang

Pembangkit Listrik Tenaga Uap (PLTU) adalah salah satu instalasi penghasil listrik yang melayani kebutuhan listrik masyarakat. Salah satu komponen utama yang memegang peranan penting pada PLTU adalah *boiler*. Ketel uap atau yang lebih dikenal dengan *boiler* merupakan bagian penting dan berfungsi menghasilkan uap untuk berbagai proses operasi di industri, khususnya sebagai penggerak turbin generator sebagai penghasil energi listrik.

Secara umum *boiler* terdiri dari beberapa sistem, diantaranya adalah sistem air umpan, sistem steam dan sistem pembakaran yang terintegrasi menjadi satu kesatuan untuk menghasilkan kinerja yang maksimal dari *boiler*. Sistem air umpan berfungsi sebagai penyedia air untuk *boiler* yang bekerja secara otomatis sesuai kebutuhan steam dan kemampuan dari *boiler* itu sendiri. Sedangkan sistem steam berfungsi sebagai penyedia uap air untuk proses pada plant yang lain. Pada saat *boiler* bekerja, air didalamnya akan terus berkurang sering berjalannya waktu sehingga diperlukan pengisian air kedalam *boiler*, karena air yang dimasukkan memiliki temperatur normal mengakibatkan air yang dipanaskan didalam *boiler* yang awalnya sudah mencapai temperatur untuk berubah menjadi uap bertekanan mengalami penurunan suhu sehingga laju aliran uap menjadi turun. Pengontrolan sistem pembakaran pada *boiler* sangatlah penting karena untuk menjaga laju aliran yang masuk ke *Steam Turbine*.

Untuk mengatasi hal tersebut, dapat dibuat suatu pengontrolan pembakaran *boiler* secara otomatis yang dapat mengatur besar kecilnya pembakaran pada *boiler* ketika pengisian air sedang berlangsung. Pada skripsi kali ini plant yang penulis gunakan adalah berupa miniature *Steam Turbine* dimana sistem pembakarannya berupa kompor gas yang tuasnya dikopel oleh Motor DC Servo. Penelitian skripsi ini bertujuan untuk menjaga agar kecepatan *Steam Turbine* tetap berada pada setpoint yang diinginkan, yaitu sebesar 22 rotations per second.

Pengendalian kecepatan *Steam Turbine* ini dilakukan dengan menggunakan kontroler PI. Kontroler PI merupakan gabungan dari kontroler proporsional dan kontroler integral. Diharapkan dengan menggunakan kontroler PI, performa sistem yang didapatkan menjadi stabil yaitu dapat mempertahankan sudut servo dan menaikkan sudut servo saat terjadi pengisian air pada *boiler*.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada point berikut:

1. Bagaimana cara merancang dan membuat sistem pengendalian kecepatan mini *Steam Turbine* menggunakan kontroler PI?
2. Bagaimana respon sistem tanpa kontroler dan dengan penambahan kontroler jika diberi gangguan?

1.3 Batasan Masalah

- a) *Boiler* yang digunakan adalah electric cup heater yang telah dimodifikasi sedemikian rupa sehingga mampu bekerja sebagai *boiler* dengan ukuran tinggi x diameter yaitu 12,5 cm x 12 cm.
- b) *Steam Turbine* yang digunakan adalah Akrilik berbentuk lingkaran dengan diameter 7 cm dengan 12 baling-baling.
- c) Generator yang digunakan adalah motor DC 12 volt.
- d) Mikrokontroler yang digunakan adalah Arduino Uno.
- e) Sensor yang digunakan adalah sensor LM393 dan piringan *rotary encoder* yang dikopel pada poros *Steam Turbine*.
- f) Pembahasan ditekankan pada penggunaan kontroler PI pada sistem, untuk masalah tekanan, suhu dan level pada *boiler* tidak dibahas secara mendalam.

1.4 Tujuan

Tujuan dari penelitian ini adalah mampu merancang sistem yang dapat mempertahankan kecepatan *Steam Turbine* meskipun terdapat gangguan berupa penambahan air dengan temperature normal dengan tetap mempertahankan sudut Motor Servo agar sesuai dengan setpoint yang ditentukan dengan menggunakan kontroler PI.

1.5 Sistematika Penulisan

Skripsi ini terdiri dari lima bab dengan sistematika pembahasan sebagai berikut:

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Menjelaskan dasar teori yang mendukung dalam pembuatan alat dan perancangan algoritma yang terdiri atas mikrokontroler, sensor kecepatan LM393 dan piringan encoder, Arduino Uno, Motor DC servo, kontroler PI, dan metode *Hand Tuning*.

BAB III METODELOGI PENELITIAN

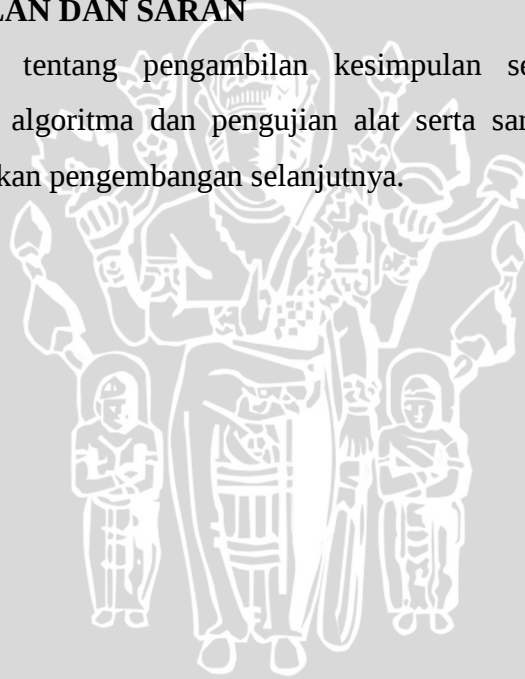
Bab ini menjelaskan bagaimana melakukan pengujian alat-alat yang digunakan untuk membuat sistem pengendali intensitas cahaya pada miniatur ruang kuliah menggunakan kontroler PI dengan metode *Hand Tuning*.

BAB IV HASIL DAN PEMBAHASAN

Membahas hasil pengujian sistem, simulasi sistem dan analisis data secara keseluruhan terhadap alat yang telah direalisasikan.

BAB V KESIMPULAN DAN SARAN

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perancangan algoritma dan pengujian alat serta saran yang diperlukan untuk dilakukan pengembangan selanjutnya.





BAB II

TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang di buat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini. Teori penunjang yang akan di bahas dalam bab ini adalah:

- Plant *Boiler*
- Mini *Steam Turbine*
- Mikrokontroler Arduino UNO R3
- Motor DC Servo
- PWM (*Pulse Width Modulation*)
- Sensor Kecepatan *rotary encoder*
- Kontroller PI
- Metode Kontroler

2.1 Ketel uap (*Boiler*)

Ketel Uap (*Boiler*) adalah alat untuk menghasilkan uap air, yang akan digunakan untuk pemanasan atau tenaga gerak. Bahan bakar pendidih bermacam-macam dari yang populer batubara dan minyak bakar, sampai listrik, gas, biomasa, nuklir dan lain-lain. *Boiler* merupakan bagian terpenting dari penemuan mesin uap yang merupakan pemicu lahirnya revolusi industri.

Dilihat dari letak pembakarannya, ketel uap dibagi menjadi :

2.1.1 Ketel dengan pembakaran di dalam (*internally fired steam boiler*)

Internally fired steam boiler atau ketel dengan pembakaran di dalam merupakan jenis *boiler* yang pembakarannya terdapat di dalam pipa *boiler* sehingga pembakarannya dapat dilakukan di *shell* tersebut. *boiler* dengan pembakaran di dalam memiliki ruang bakar (*furnace*) dibagian dalam *boiler*. Sistem ketel dengan pembakaran di dalam digunakan pada *boiler* jenis ketel pipa api atau biasa disebut dengan *fire tube boiler*.

Pada *boiler* pipa api (*fire tube boiler*), fluida yang mengalir adalah gas nyala dari hasil pembakaran yang membawa kalor, kemudian mentransfernya ke air *boiler* melalui bidang pemanas. Tujuan pipa-pipa api ini adalah untuk memudahkan distribusi kalor ke air *boiler*.

Keuntungan dan kerugian ketel dengan pembakaran dalam (*internally fired tube boiler*) yang diaplikasikan pada ketel pipa api :

➤ Keuntungan :

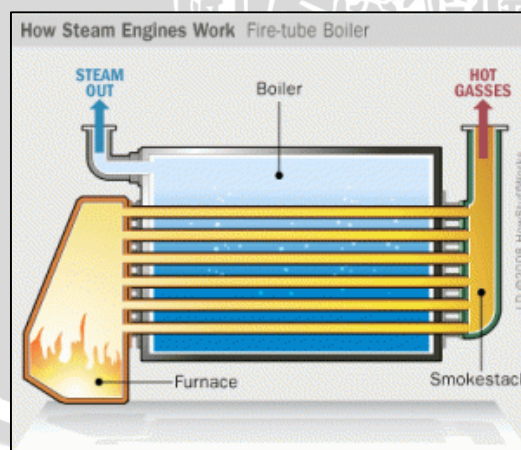
1. Ketel dengan pembakaran dalam (*internally fired tube boiler*) akan menghasilkan steam dengan tekanan lebih tinggi dari pada ketel dengan pembakaran luar (*externally fired tube boiler*).
2. Memiliki laju aliran uap lebih rendah.
3. Bentuknya portable (mudah dipindahkan) karena komponennya dapat diurai.
4. Memiliki permukaan pemanas yang lebih efektif karena kalor mengalir tegak lurus ke atas.
5. Biaya pembuatan *boiler* relatif murah.

➤ Kerugian

1. Tekanan operasi uap terbatas pada tekanan rendah.
2. Menimbulkan kerak yang dapat memecahkan pipa *boiler*.
3. Pipa *boiler* sulit untuk dibersihkan.
4. Effisiensinya rendah, karena banyak kalor yang terbuang..

➤ Contoh Internally fired steam *boiler* antara lain :

1. Ketel Lancashire dan Corn Wall
2. Ketel Lokomotif



Gambar 2. 1 Internally fired steam boiler

Sumber : <http://www.agussuwasono.com/artikel/mechanical/566-prinsip-dasar-turbin-uap.html>

2.1.2 Ketel dengan pembakaran di luar (*outernally fired steam boiler*).

Externally fired steam boiler atau ketel dengan pembakaran di luar merupakan jenis *boiler* yang pembakarannya berada di bagian luar pipa pada *boiler*. Ketel jenis ini

mempunyai dapur/ruang bakar (*furnace*) dibagian luar *boiler*. Sistem ketel dengan pembakaran di luar digunakan pada *boiler* jenis ketel pipa air atau biasa disebut dengan water tube *boiler*. Pada *boiler* pipa air (water tube *boiler*) fluida yang mengalir dalam pipa adalah air, kalor yang ditransfer dari luar pipa yang berasal dari ruang bakar/dapur (*furnace*) ke air ketel.

Pada ketel jenis ini, dapur/ruang bakar (*furnace*) ditempatkan di bawah ketel yaitu di dalam ruangan yang dikelilingi oleh bata api. Dapur/ruang bakar (*furnace*) pada *externally fired steam boiler* memiliki ukuran yang besar dan letak ruang bakar terpisah dari *boiler* tersebut, sehingga ketel jenis ini memiliki kemampuan untuk menyalurkan panas lebih besar. Contoh *externally fired steam boiler* yaitu pada *Boiler Babcock and Wilcox*.

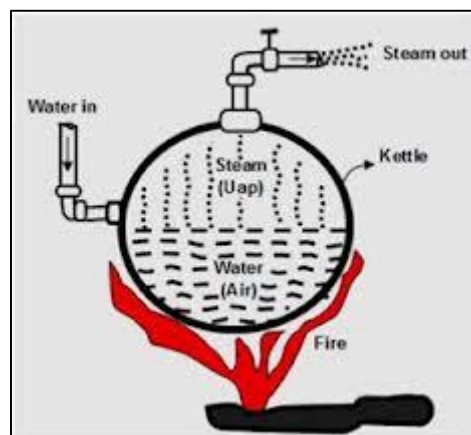
Keuntungan dan kerugian ketel dengan pembakaran luar (*externally fired tube boiler*) yang diaplikasikan pada ketel pipa air :

➤ Keuntungan

1. Nilai effisiensinya relatif tinggi.
2. Konstruksinya sederhana.
3. Memiliki ruang penampung uap yang besar.

➤ Kerugian

1. Biaya pembuatannya lebih mahal.
2. Memerlukan waktu lama untuk mendapat steam dengan kualitas yang baik.



Gambar 2. 2 Externally fired steam boiler

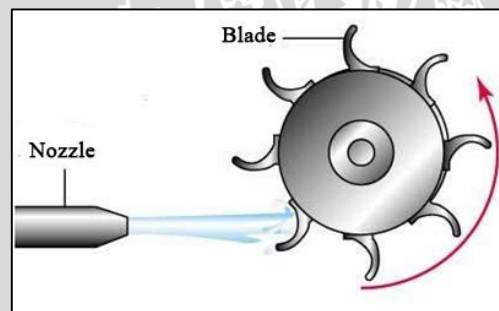
Sumber : <http://www.agussuwasono.com/artikel/mechanical/566-prinsip-dasar-turbin-uap.html>

2.2 Turbin Uap (Steam Turbine)

Turbin uap merupakan suatu penggerak mula yang mengubah energi potensial uap menjadi energi kinetik dan selanjutnya diubah menjadi energi mekanis dalam bentuk

putaran poros turbin. Poros turbin, langsung atau dengan bantuan roda gigi reduksi, dihubungkan dengan mekanisme yang akan digerakkan. Tergantung pada jenis mekanisme yang digunakan, turbin uap dapat digunakan pada berbagai bidang seperti pada bidang industri, untuk pembangkit tenaga listrik dan untuk transportasi. Pada proses perubahan energi potensial menjadi energi mekanisnya yaitu dalam bentuk putaran poros dilakukan dengan berbagai cara.

Prinsip kerja turbin uap dimulai dengan Uap masuk kedalam turbin melalui *nozzle*. Didalam *nozzle* energi panas dari uap dirubah menjadi energi kinetis dan uap mengalami pengembangan. Tekanan uap pada saat keluar dari *nozzle* lebih kecil dari pada saat masuk ke dalam *nozzle*, akan tetapi sebaliknya kecepatan uap keluar *nozzle* lebih besar dari pada saat masuk ke dalam *nozzle*. Uap yang memancar keluar dari *nozzle* diarahkan ke sudu-sudu turbin yang berbentuk lengkungan dan dipasang disekeliling roda turbin. Uap yang mengalir melalui celah-celah antara sudu turbin itu dibelokkan kearah mengikuti lengkungan dari sudu turbin. Perubahan kecepatan uap ini menimbulkan gaya yang mendorong dan kemudian memutar poros turbin.

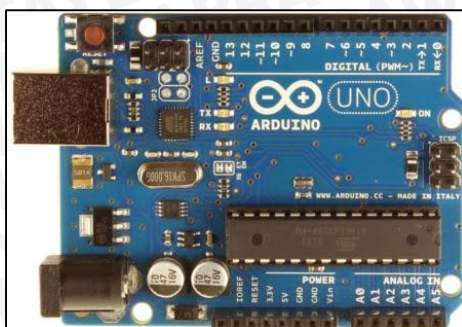


Gambar 2. 3 Prinsip kerja Turbin Uap

Sumber : http://www.academia.edu/6552176/Turbin_uap_kuliah

2.3 Mikrokontroler Arduino UNO R3

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Uno memiliki 14 pin digital input / output (dimana 6 dapat digunakan sebagai output PWM), 6 input analog, resonator keramik 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan power USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai.



Gambar 2. 4 Arduino Uno R3

Sumber: <http://ndoware.com/apa-itu-arduino-uno.html>

Tabel 2. 1 Spesifikasi Arduino UNO R3

Sumber: <https://www.arduino.cc/en/Main/arduinoBoardUno>

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Sumber tegangan arduino UNO R3 dapat diaktifkan melalui koneksi USB atau dengan daya eksternal. Eksternal (non-USB) daya dapat berasal baik dari adaptor maupun baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jackpusatpositif ukuran 2.1 mm konektor power. Ujung kepala dari baterai dapat dimasukkan kedalam ground dan vcc pin header dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 V sampai 20 V. Namun jika menggunakan lebih dari 12 V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 V sampai 12 V. Arduino juga menyediakan *Software Serial Library* memungkinkan komputer untuk berkomunikasi secara serial pada salah satu pin digital pada board arduino UNO R3.

2.4 Motor DC Servo MG995

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

Motor servo mampu bekerja dua arah (*CW* dan *CCW*) dimana arah dan sudut pergerakan rotasinya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Gambar 2.4 Menunjukkan gambar motor servo.



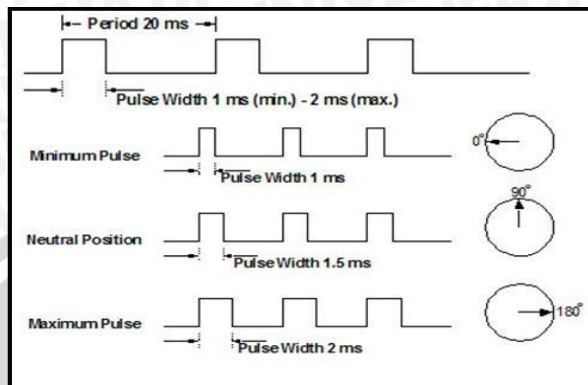
Gambar 2. 5 Motor DC Servo MG995

Spesifikasi MG-995 Servo Motor :

- Modulasi: Analog
- Torsi: 10 kg.cm
- Kecepatan: 0,16 detik / 60° (pada 6 Volt)
- Kecepatan reaksi: 53 - 62 rpm
- Dimensi: 40,6 x 19,8 x 42,9 mm (berat 55g)
- Tipe gir: Metal
- Rentang Rotasi: 180°
- Siklus pulsa 20 ms dengan lebar pulsa $\pm 1,5$ ms
- Catu daya 4,8 ~ 7,2 Volt DC (tipikal 6V, 4x baterai 1,5V disusun seri)

2.5 PWM (Pulse With Modulation)

PWM (*Pulse Width Modulation*) memiliki fungsi untuk mengatur sudut putaran motor servo. Teknik PWM (*pulse width modulation*) untuk mengatur sudut motor servo dapat dilihat dalam Gambar 2.5.



Gambar 2. 6 Pengaturan Sudut Motor Servo

Sumber : www.electroons.com

Berdasarkan Gambar 2.5 dapat diasumsikan bahwa saat diberikan sinyal periodik dengan lebar sebesar 1 ms maka motor servo akan bergerak dengan sudut 0° . Apabila diberikan sinyal sebesar 1.5 ms maka motor servo akan bergerak dengan sudut 90° . Begitu pula jika diberi sinyal 2 ms, maka motor servo akan bergerak dengan sudut 180° . Perhitungan rumus motor servo akan ditunjukkan Persamaan 2-1 sebagai berikut:

$$S = D \left(\frac{1000}{180} \right) + 1000 \mu s \dots \dots \dots (2-1)$$

Dimana:

S = Lebar Pulsa dalam μs

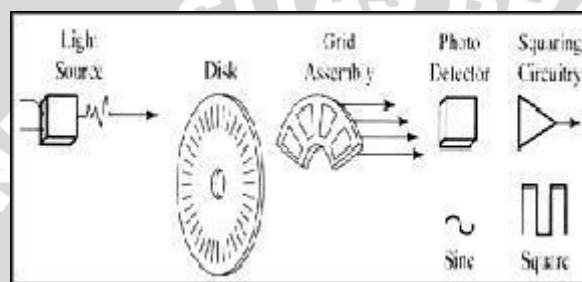
D = Sudut putar servo dalam derajat

2.6 Sensor Kecepatan Rotary encoder

Rotary encoder adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *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 tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, 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 1 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.

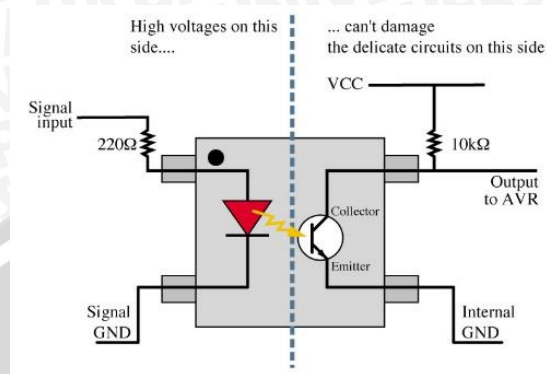


Gambar 2. 7 Blok penyusun rotary encoder

Sumber : www.konversi.wordpress.com

Rotary encoder pada perancangan sistem digunakan untuk mendeteksi kecepatan gerak dari mikrokontroler. *Rotary encoder* dibuat dari rangkaian sensor optocoupler yang berbentuk “U” dan piringan berbentuk bulat yang telah dilubangi. Didalam optocoupler terdapat diode penghasil cahaya, led (light emitting dioda), dan sebuah transistor yang peka terhadap intensitas cahaya (photo-transistor). Jika lubang U optocoupler tertutup maka akan menghasilkan pulsa keluaran high, dan jika Lubang optocoupler terbuka maka akan menghasilkan pulsa keluaran low. Dengan menghitung lama perubahan pulsa, dan jarak perubahan lubang pada piringan inilah yang digunakan untuk menghitung kecepatan gerakan motor. Lubang dalam piringan mempengaruhi banyaknya pendeteksian pulsa pada *rotary encoder* untuk satu putaran yang dilakukan oleh shaft yang berputar. Karena banyaknya pulsa atau lubang akan mempengaruhi perumusan perhitungan kecepatan. Semakin banyak pulsa atau lubang maka akan meningkatkan resolusi pembacaan kecepatan.

- Rumus untuk menghitung kecepatan gerak dengan *rotary encoder* :
 - Kecepatan = (pulsa*keliling)
 - Pulsa= jumlah pulsa/banyaknya lubang (perdetik)



Gambar 2. 8 Rangkaian Optocoupler

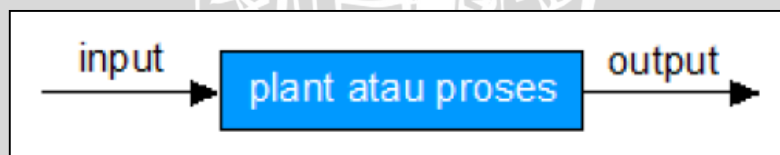
Sumber : www.konversi.wordpress.com

2.7 Loop Pengendalian

Secara umum bentuk *loop* sistem pengendalian dibagi menjadi dua macam, yaitu:

2.7.1. Sistem Pengendalian Loop Terbuka (*Open Loop Control System*)

Sistem pengendalian *loop* terbuka adalah sistem pengendalian yang keluarannya tidak dapat mempengaruhi aksi dari pengendaliannya. Jadi, pada sistem ini keluaran dari kontrolernya tidak diukur atau diumpanbalikkan untuk membandingkan dengan masukkannya (K. Ogata, 1985).

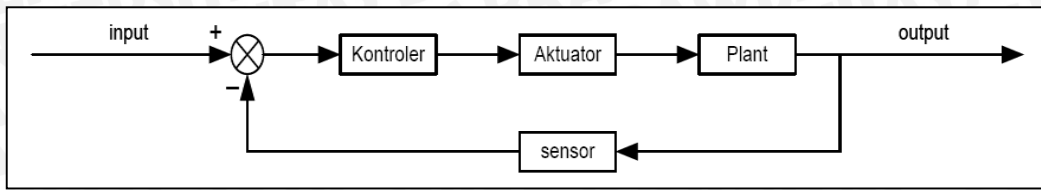


Gambar 2. 9 Blok Diagram Sistem Loop Terbuka

Sumber: K. Ogata, 1985

2.7.2. Sistem Pengendalian Loop Tertutup (*Close Loop Control System*)

Sistem pengendalian *loop* tertutup adalah sistem pengendalian yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengendaliannya (K. Ogata, 1985). Sinyal kesalahan yang bekerja yaitu antara sinyal masukan dan sinyal umpan balik yang disajikan ke kontroler disajikan sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki.



Gambar 2. 10 Block Diagram Close Loop

Sumber: K. Ogata, 1985

2.8 Kontroler

Kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Hal itu disebabkan karena tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, perubahan perilaku sistem hanya dapat dilakukan melalui penambahan kontroler.

Salah satu fungsi kontroler adalah mengurangi sinyal *error*, sinyal *error* adalah perbedaan nilai *setpoint* dengan nilai *output plant*. *Setpoint* adalah nilai referensi atau nilai yang diinginkan, sedangkan *output plant* adalah nilai aktual yang terukur pada *output plant*. Semakin kecil nilai sinyal *error* maka kinerja sistem kontrol dinilai semakin baik.

Prinsip kerja kontroler adalah membandingkan nilai *output plant* dengan nilai *setpoint*, menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata K,1995).

2.8.1. Kontrol Proporsional (P)

Kontroler proporsional memiliki *output* yang besarnya sebanding dengan besarnya sinyal *error*. *Output* kontroler merupakan perkalian antara penguatan proporsional dengan sinyal *error*, yang dapat dituliskan sebagai berikut:

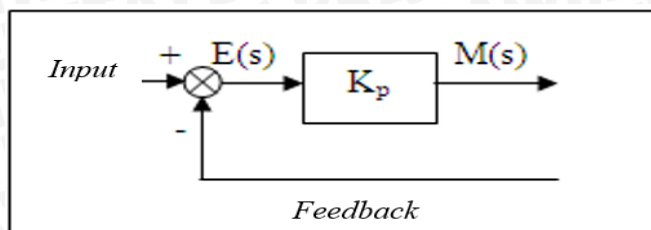
$$m(t) = K_p e(t) \dots \dots \dots (2.2)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots \dots \dots (2.3)$$

Di mana K_p adalah gain proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997).



Gambar 2. 11 Diagram Blok Kontroler Proporsional

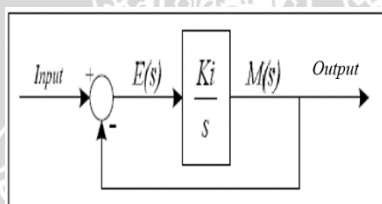
(Sumber: Ogata K., 1997)

2.8.2. Kontroler Integral (I)

Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$. Jadi,

$$\frac{dm(t)}{dt} = Kie(t) \dots\dots\dots(2.4)$$

Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* (Ogata K.,1997).

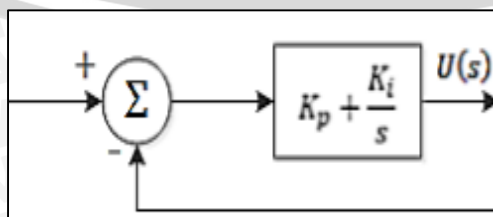


Gambar 2. 12 Diagram Blok Kontroler Integral

(Sumber: Ogata K., 1997)

2.8.3. Kontroler Proporsional Integral (PI)

Kontroler *Proportional Integral* (PI) memiliki sifat yang tidak mengeluarkan respon sebelum selang waktu tertentu, pada kontrol P suatu *plant* yang fungsi alihnya tidak mempunyai integrator $1/s$, terdapat *error steady state (offset)* pada masukan tangga atau *unit step*. *Offset* semacam ini dapat dihilangkan jika ditambahkan aksi integral pada kontroler. kontroler I mengakibatkan respon menjadi lambat walaupun dapat menghilangkan *error steady state (offset)*. Dalam memperbaiki respon yang lambat tersebut, umumnya kontroler I dipasang paralel dengan kontroler P.



Gambar 2. 13 Diagram Blok Kontroler Proporsional Integral

(Sumber: Ogata K., 1997)

Bentuk fungsi alih dari kontroler Proporsional Integral (PI) seperti pada Persamaan 2.5.

$$U(s) = \left[Kp + \frac{Ki}{s} \right] E(s) \dots\dots\dots(2.5)$$

Dengan $U(s)$: Sinyal Kontrol
 $E(s)$: Nilai *error* sistem
 Kp : *Gain* Kontroler Proporsional
 Ki : *Gain* Kontroler Integral

Karena kontroler PI merupakan penggabungan dari dua unit kontroler P dan I, semua kelebihan serta kekurangan yang ada pada kontroler P dan kontroler I masih tetap ada. Sifat kontroler P yang selalu meninggalkan *error steady state (offset)* dapat ditutupi oleh kelebihan kontroler I, sedangkan sifat kontroler I yang lambat dapat ditutupi oleh kontroler P, sehingga kontroler PI menghasilkan respon yang lebih cepat dari kontroler I tapi mampu menghilangkan *error steady state (offset)* yang dihasilkan kontroler P.

2.9 Metode Hand Tuning

Kontroler PID dapat di tuning dalam beberapa cara, antara lain *ziegler-nichols tuning*, *loop shaping*, metode analitis, optimasi, *pole placement*, *autotuning* dan *hand tuning* (Smith,1997; Astrom & Hagglund,2004). Dengan menggunakan metode *hand tuning*, nilai parameter P perlu diubah-ubah secara *trial* agar respon yang diperoleh sesuai dengan yang diinginkan.

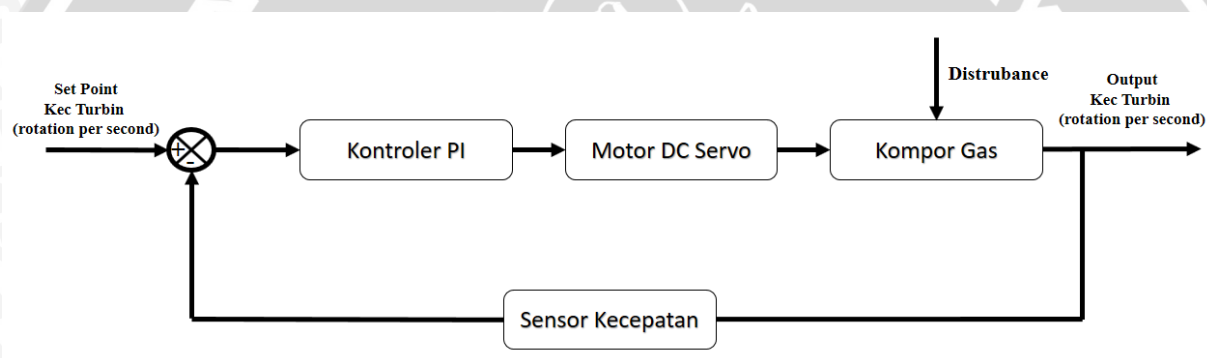
BAB III

METODOLOGI PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan informasi dengan tujuan dan manfaat tertentu. Metodologi penelitian disini berfungsi agar tujuan skripsi yang terdapat di bab pendahuluan dapat tercapai. Berikut ini adalah langkah-langkah yang akan digunakan dalam penelitian.

3.1 Perancangan Blok Diagram Sistem

Perancangan diagram blok sistem pengaturan suhu bertujuan untuk mempermudah dalam perancangan perangkat keras sistem maupun perancangan perangkat lunak sistem yang berhubungan dengan pengaturan suhu. Diagram blok yang dirancang untuk sistem pengujian keseluruhan dapat dilihat di Gambar 3.1



Gambar 3. 1 Blok Diagram Sistem

Keterangan :

1. *Setpoint* sistem berupa kecepatan turbin sebesar 22 rotation per second
2. Kontroler yang digunakan adalah kontroler Proportional Integral (PI) menggunakan perangkat keras Arduino UNO R3.
3. *Plant* yang digunakan adalah *Steam Boiler*.
4. Sensor *Rotary encoder* digunakan sebagai pembaca kecepatan turbin.
5. Gangguan pada sistem berupa penambahan air pada tabung *Steam Boiler*.

3.2 Spesifikasi Desain

Desain yang diinginkan pada perancangan pengontrolan kecepatan Mini Steam Turbin mempunyai spesifikasi :

1. *Error Steday State* <5%

Error Steady State <5% karena sistem yang baik memiliki output dengan batas error akhir 5% dari *setpoint*.

2. *Settling time* < 60 detik

Settling time < 60 detik, karena diharapkan penggunaan kontroler PI mampu mempercepat *Settling time* kurang dari 60 detik.

3. *Output* tanpa *Overshoot*

Penggunaan controller PI diharapkan mampu meminimalkan *Overshoot*.

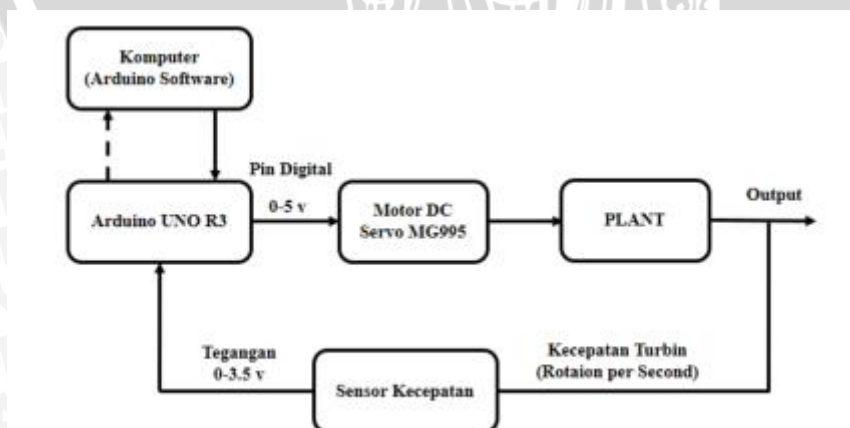
3.3 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem, hal ini bertujuan supaya *output* kecepatan *Steam Turbine* sesuai dengan *input* yang diinginkan dapat bekerja dengan baik sesuai deskripsi awal yang telah direncanakan.

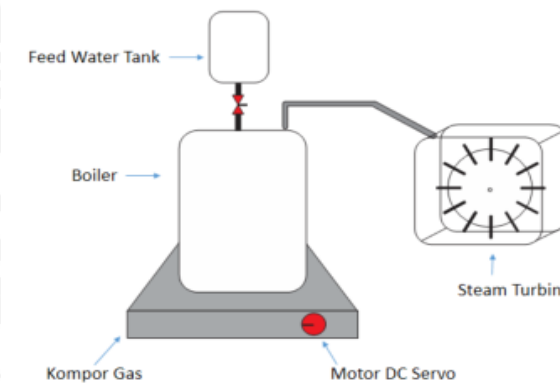
Pembuatan perangkat keras meliputi:

1. Skema perancangan perangkat keras (Gambar 3.2).
2. Skema perancangan model miniatur *Steam Turbine* (Gambar 3.3)
3. Penentuan modul elektronik yang digunakan meliputi:
 - Seperangkat Komputer.
 - Mikrokontroler Arduino UNO
 - Motor DC Servo MG995
 - Sensor *Rotary encoder*

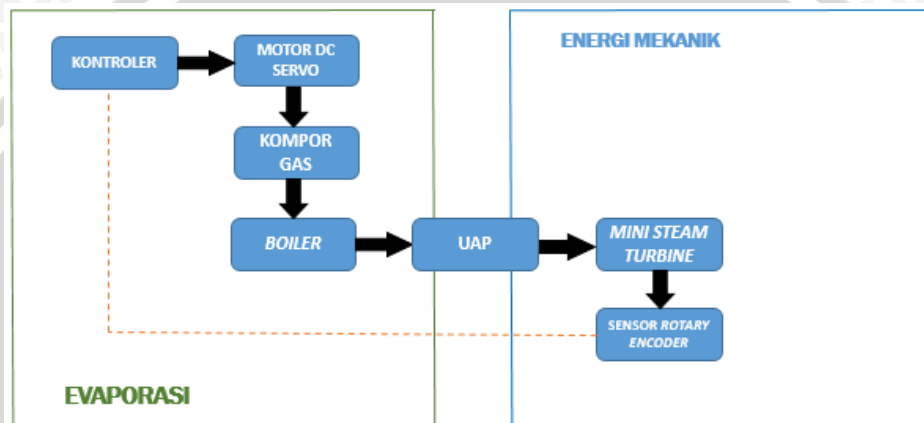
Perangkat keras diatas nantinya akan membentuk sebuah sistem dimana cara kerjanya sebagai berikut.



Gambar 3. 2 Skema perancangan perangkat keras



Gambar 3. 3 Skema perancangan model miniatur Steam Turbine



Gambar 3. 4 Rancangan Sistem

Penjelasan mengenai blok diagram sistem diatas adalah sebagai berikut :

1. Kontroler berfungsi mengatur sudut servo otomatis berdasarkan hasil pembacaan Sensor Rotary Encoder.
2. Kontroler mengirimkan sinyal kontrol ke Motor DC Servo yang sudah dikopel pada tuas kompor gas *portable*.
3. Kompor gas *portable* digunakan sebagai pemanas boiler.
4. Boiler berfungsi sebagai *vaporizer* dimana perubahan dari zat cair menjadi gas terjadi . Boiler memanaskan air hingga mencapai titik didihnya .
5. Uap adalah hasil keluaran dari boiler yang berupa gas. Uap ini akan dilewatkan melalui selang sebagai media penyambung antara *vaporizer* dan *mini steam turbine*.
6. Uap dari boiler akan masuk menuju melewati *nozzle* yang akan meniup baling baling yang menyebabkan *mini steam turbine* berputar.
7. Kecepatan *mini steam turbine* dibaca oleh sensor rotary encoder dan diumpkan balikkan ke kontroler.

3.4 Spesifikasi Alat

Spesifikasi alat yang digunakan meliputi komponen-komponen pendukung skripsi ini adalah sebagai berikut :

1. Komputer atau PC dengan *software* Arduino.
2. Perangkat kontroler yang digunakan adalah Arduino Uno (Gambar 3.4).
3. Kompor gas *portable* (Gambar 3.5).
4. *Boiler* (Gambar 3.6).
5. Motor DC Servo MG995 (Gambar 3.7).
6. Mini *Steam Turbine* (Gambar 3.8)
7. Sensor *Rotary encoder* (Gambar 3.9).



Gambar 3. 5 Mikrokontroler Arduino Uno

Sumber: www.caratekno.com

Mikrokontroler Arduino ini berfungsi sebagai penyimpan algoritma kontroler sistem yang dikirimkan melalui komputer atau PC, pengendali Motor DC Servo, dan mengolah data yang dikirimkan sensor.



Gambar 3. 6 Kompor gas portable

Kompor gas portable digunakan sebagai pemanas *Boiler*, dimana tuas kompor gas dikopel oleh Motor DC Servo.



Gambar 3. 7 Boiler

Ketel uap (*Boiler*) terbuat dari wadah *electric cup heater* berbahan *stainless steel* yang sudah didesain sedemikian rupa sehingga mampu bekerja sebagai *boiler*, adapun pada bagian atas *boiler* ditambah oleh wadah *stainless steel* lainnya yang dihubungkan oleh valve sebagai penambah air saat terjadi penurunan level air. *Boiler* berfungsi sebagai penghasil uap yang akan digunakan untuk mendorong turbin.



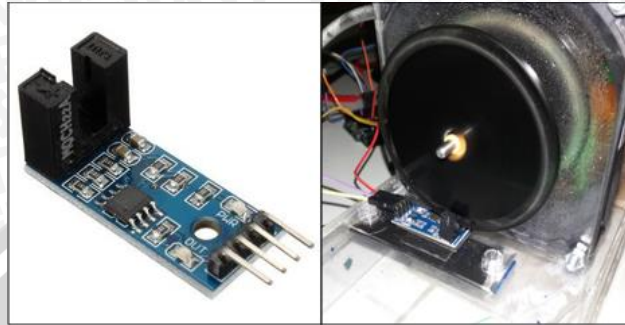
Gambar 3. 8 Motor DC Servo MG 995

Motor DC Servo MG995 berfungsi sebagai aktuator yang telah dikopel dengan tuas kompor gas yang nantinya akan mengatur perapian secara otomatis



Gambar 3. 9 Mini Steam Turbine

Mini *Steam Turbine* berupa miniature yang terbuat dari Akrilik berbentuk lingkaran dengan diameter 7 cm dengan 12 baling-baling. Dibagian luar dilengkapi piringan *rotary encoder* yang berfungsi untuk pembacaan sensor. Penutupnya terbuat dari chasing Brushless Motor DC dengan penutup disisi kiri dan kanan berupa Akrilik.



Gambar 3. 10 Sensor Rotary encoder

Rotary encoder berupa piringan tipis yang biasanya di kopel dengan poros yang berputar, umumnya di kopel langsung dengan shaft motor. Piringan tipis tersebut terdapat lubang di sepanjang pinggir lingkarannya. Di bagian sisi-sisi piringan terdapat sebuah led dan phototransistor di bagian bersebrangan. Fungsi dari lubang-lubang yang berada di sepanjang pinggir lingkaran tersebut akan menghantarkan cahaya led ke phototransistor, sebaliknya jika cahaya led tidak menembus lubang piringan maka cahaya akan tertahan. Piringan tersebut akan berputar sesuai dengan kecepatan putaran motor sehingga phototransistor akan saturasi ketika cahaya led menembus lubang-lubangnya. Pada saat saturasi phototransistor akan menghasilkan pulsa dengan range +0.5V s/d +5V. Semakin banyak lubang yang berada pada piringan tentu saja semakin banyak pulsa yang dihasilkan selama satu putaran.

3.5 Pin Arduino Uno

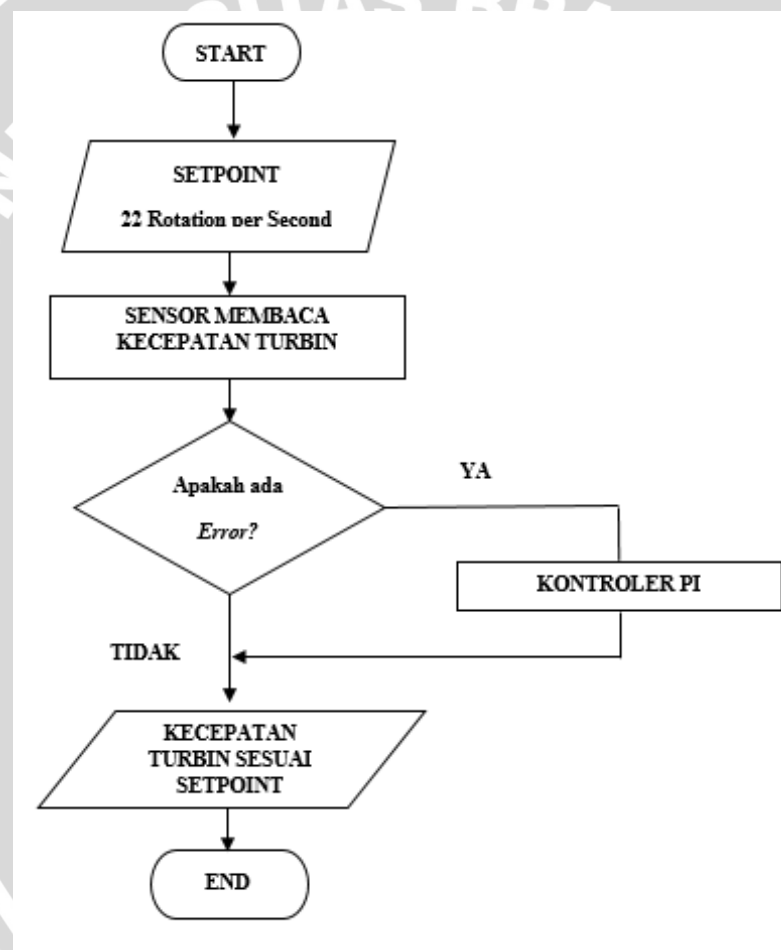
Arduino Uno adalah *board* mikrokontroler berbasis ATmega328 yang memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHZ osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Fungsi setiap pin ditunjukkan dalam tabel 3.1.

Tabel 3. 1 Fungsi pin Arduino

NO	Pin	Fungsi
1	5 Volt	Catu daya Motor DC Servo
2	3.3 Volt	Catu daya sensor RotaryEncoder
3	GND	Jalur masukan GND seluruh sistem
4	2	Digital pin untuk Sensor <i>Rotary encoder</i>
5	9	Digital pin untuk Motor DC Servo

3.6 Flowchart Perancangan Perangkat Lunak

Flowchart keseluruhan sistem ditunjukkan pada Gambar 3.11



Gambar 3. 11 Flowchart keseluruhan sistem

3.7 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang didapat dari hasil pengujian sistem secara keseluruhan. Apabila hasil yang didapatkan sesuai dengan yang direncanakan

sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan dapat dikembangkan untuk penelitian selanjutnya untuk disempurnakan.



BAB IV

HASIL DAN PEMBAHASAN

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat sesuai dengan perancangan dan berfungsi dengan baik. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu :

1. Pengujian *Rotary encoder*
2. Pengujian Motor DC Servo
3. Perancangan Kontroller PI
4. Penentuan nilai penguatan (Gain)
5. Pengujian keseluruhan sistem tanpa gangguan
6. Pengujian keseluruhan sistem dengan gangguan

4.1 Pengujian *Rotary encoder*

a. Tujuan

Tujuan pengujian sensor untuk mengetahui seberapa besar *error* sensor yang digunakan dengan cara membandingkannya dengan suhu yang tertera pada thermometer digital.

b. Peralatan yang dipakai:

1. Laptop
2. Sensor *rotary encoder*.
3. Arduino UNO R3.
4. Tachometer Digital.

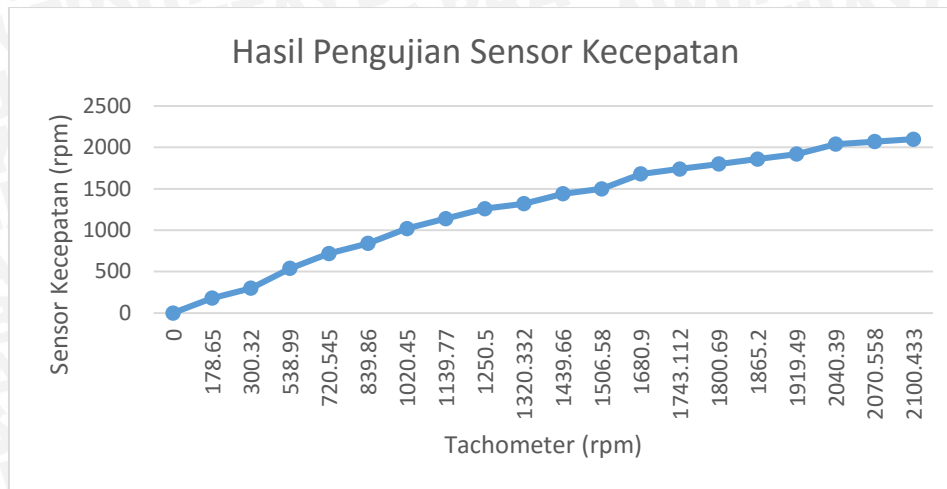
c. Langkah Pengujian

1. Membuat program pada Arduino.
2. Menyiapkan tachometer.
3. Mencatat hasil pengujian sensor dan tachometer dan dibandingkan.

Data hasil dari pengujian sensor kecepatan yang dibandingkan dengan tachometer digital dapat dilihat pada Tabel 4.1 dan garfik (Gambar 4.1).

Tabel 4. 1 Hasil pengujian sensor Rotary Encoder

Tachometer (rpm)	Sensor (rpm)
0	0
178.65	180
300.32	300
538.99	540
720.545	720
839.86	840
1020.45	1020
1139.77	1140
1250.5	1260
1320.332	1320
1439.66	1440
1506.58	1500
1680.9	1680
1743.112	1740
1800.69	1800
1865.2	1860
1919.49	1920
2040.39	2040
2070.558	2070
2100.433	2100



Gambar 4. 1 Hasil Pengujian Sensor Kecepatan

4.2 Pengujian Motor DC Servo

a. Tujuan

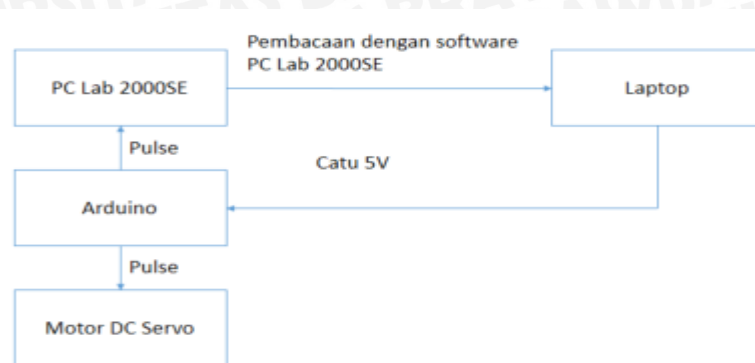
Tujuan Motor DC Servo untuk mengetahui karakteristik dari motor DC Servo MG995 itu sendiri. Pengujian ini dilakukan dengan cara memasang PC Lab 2000SE pada keluaran data Arduino.

b. Peralatan yang dipakai:

1. Laptop
2. PC Lab2000SE
3. Kabel penghubung PC Lab dan laptop
4. Arduino UNO R3
5. Software PC Lab 2000SE

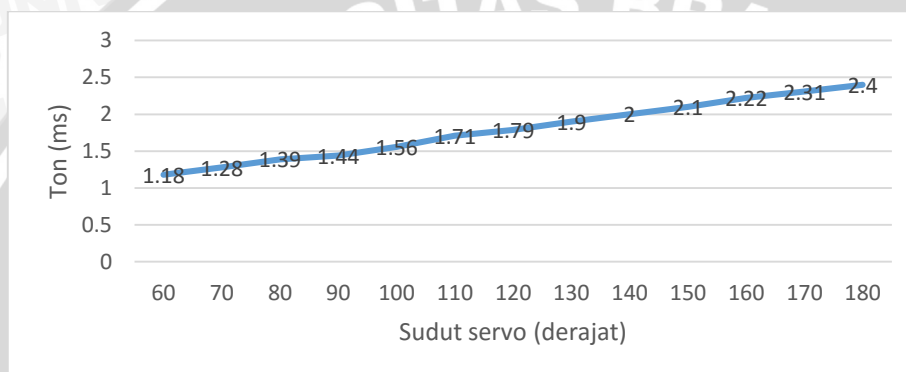
c. Langkah Pengujian

1. Menghubungkan antara PC Lab 2000SE pada channel 1 dengan keluaran data pada Arduino , dan juga ground yang digunakan pada arduino
2. Menghubungkan PC Lab 2000SE dengan Laptop
3. Menghubungkan Arduino UNO R3
4. Memasukkan algoritma motor DC servo untuk tiap sudut
5. Melihat hasil percobaan pada software PC Lab 2000SE



Gambar 4. 2 Blok Diagram pengujian Motor DC Sevo

Hasil dari percobaan motor DC Servo dapat dilihat pada Gambar 4.3 berikut :



Gambar 4. 3 Hasil percobaan pengujian motor DC Servo

4.3 Perancangan Kontroler PI

Pada penelitian kali ini untuk memenuhi tujuan performansi loop yang diinginkan, maka perlu ditambahkan kontroler dalam sistem. Kontrol yang dipilih adalah kontrol proporsional integral. Kontroler proposional integral dipilih karena karakteristik *plant* tersebut tidak membutuhkan kontrol yang terlalu cepat tetapi meminimalkan *overshoot*.

Pencarian parameter kontroler PI dapat dicari menggunakan beberapa cara, antara lain *ziegler-nichols tuning*, *loop shaping*, metode analitis, optimasi, *pole placement*, *autotuning* dan *hand tuning* (Smith,1997; Astrom & Hagglund,2004). Dengan menggunakan metode *hand tuning*, nilai parameter PI yaitu Kp dan Ki perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan yang diinginkan.

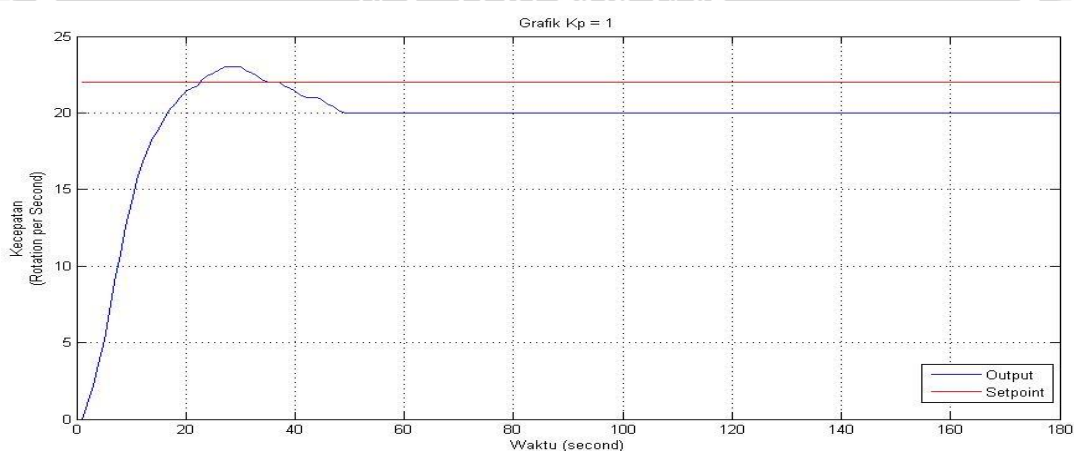
Metode *hand tuning* digunakan karena menghasilkan nilai sesuai dengan set point yaitu, nilai kp, ki dan kd yang sesuai dengan yang diharapkan. Meskipun menggunakan metode *hand tuning* adapun beberapa prosedur yang harus dilakukan untuk mendapatkan hasil yang sesuai diantaranya :

1. Atur nilai K_p dan hilangkan nilai K_i dengan memasukkan nilai $K_i = 0$ hingga sistem mencapai keadaan *setpoint* seperti pada Tabel 4.2.

Tabel 4. 2 Tabel Penentuan parameter K_p

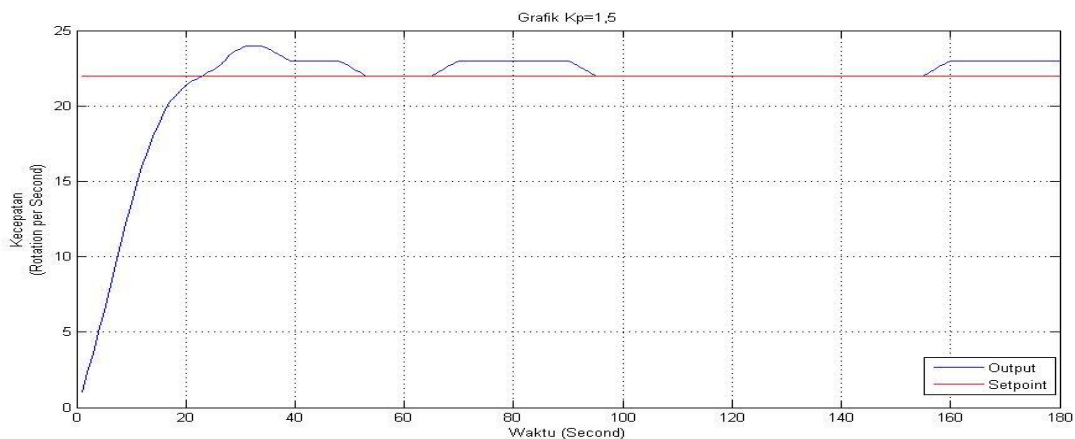
Pengujian Ke-	K_p	K_i	Settling Time (s)	Keterangan
1.	1	0	47	• Terjadi Overshoot 0,01 % • Terdapat Error 8,445% • Mencapai <i>setpoint</i> dan mengalami penurunan
2.	1,5	0	43	• Terjadi Overshoot 1,09% • Terdapat Error 1.678% • Mencapai <i>setpoint</i> dan <i>steady</i> di 22 rotation per second
3.	2	0	49	• Terjadi Overshoot 1,92% • Terdapat Error 1,89 • Mencapai <i>setpoint</i> dan <i>steady</i> di 22 rotation per second

Berdasarkan Tabel 4.2 dapat dilihat dimana ketika nilai $K_p=1$ dan nilai $K_i=0$ sistem tidak mampu mencapai *setpoint* yang diinginkan dan terjadi *overshoot* seperti yang ditunjukkan dalam Gambar 4.4.



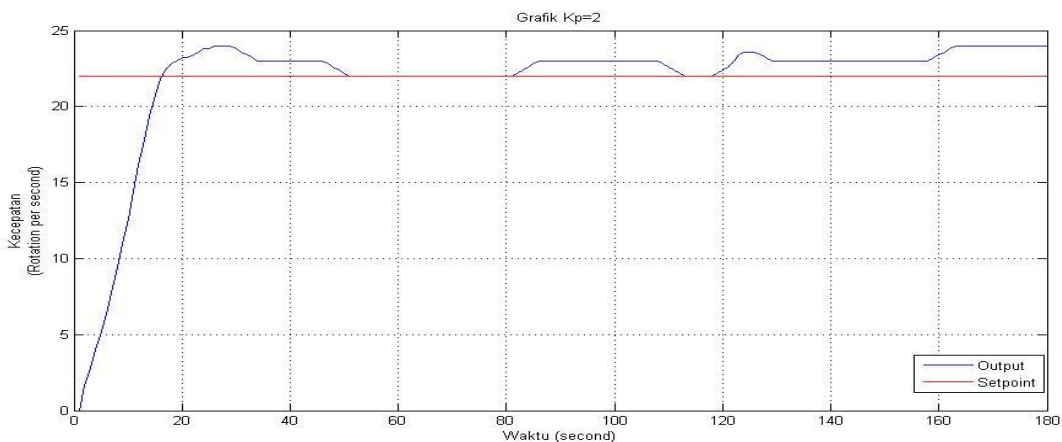
Gambar 4. 4 Grafik ketika $K_p=1$, $K_i=0$

Sedangkan bila K_p diberi nilai sebesar 1,5 sistem mampu mencapai *setpoint* yang diinginkan meskipun terdapat sedikit *error* seperti yang ditunjukkan dalam Gambar 4.5.



Gambar 4. 5 Grafik ketika $K_p=1,5$ dan $K_i=0$

Tetapi bila nilai K_p diberikan nilai sebesar 2, sistem mampu melewati *setpoint* tetapi terdapat *error* yang cukup besar bila dibandingkan dengan nilai K_p sebesar 1,5 seperti yang ditunjukkan dalam Gambar 4.6.



Gambar 4. 6 Grafik ketika $K_p=2, K_i=0$

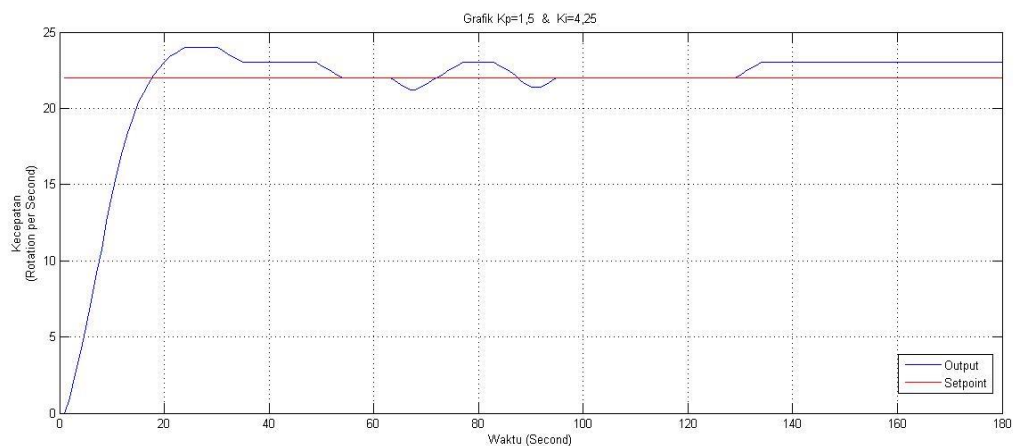
- Atur nilai K_i untuk mengurangi *error* hingga mendapatkan respon yang diinginkan. Setelah mengatur beberapa nilai K_i , didapatkan data seperti yang ditunjukkan dalam Tabel 4.3.

Tabel 4. 3 Pengujian beberapa nilai K_i pada sistem

Pengujian ke-	K_p	K_i	Settling Time (s)	Mencapai <i>Setpoint</i>	Keterangan
1.	1,5	4,25	52	Ya	Ess(%)= 1,867 %

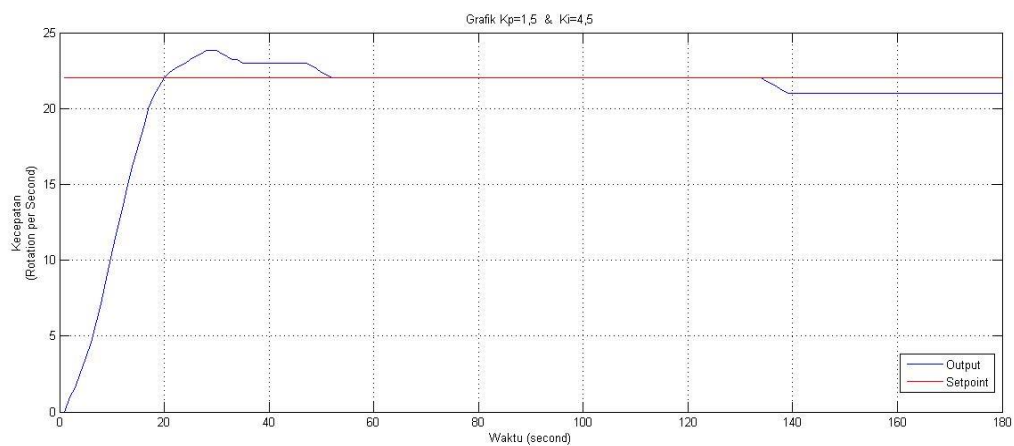
2.	1,5	4,5	51	Ya	Ess(%) =1,526 %
3.	1,5	4,75	36	Ya	Ess(%)= 1,369 %

Berdasarkan Tabel 4.3, bila sistem diberikan nilai Ki sebesar 4,25 sistem mampu mencapai *setpoint* pada detik ke 33, tetapi terdapat *error steady state* sebesar 1.867% seperti yang ditunjukkan dalam Gambar 4.7.



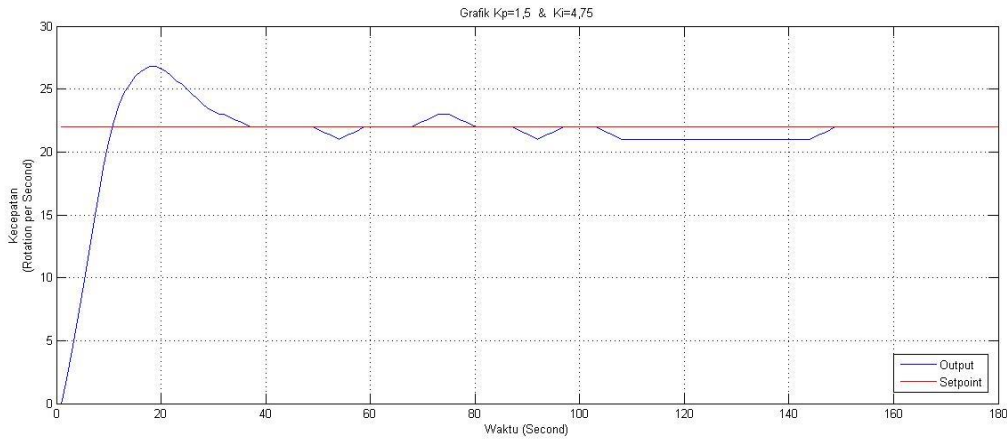
Gambar 4. 7 Grafik ketika KP=1,5 KI=4,25

Berdasarkan Tabel 4.3, bila sistem diberikan nilai Ki sebesar 4,5 sistem mampu mencapai *setpoint* pada detik ke 52, tetapi terdapat *error steady state* sebesar 1,526 % seperti yang ditunjukkan dalam Gambar 4.8.



Gambar 4. 8 Grafik ketika Kp=1,5 Ki=4,5

Berdasarkan Tabel 4.3, bila sistem diberikan nilai Ki sebesar 4,75 sistem mampu mencapai *setpoint* pada detik ke 36, tetapi terdapat *error steady state* sebesar 1,369% seperti yang ditunjukkan dalam Gambar 4.9.



Gambar 4. 9 Grafik ketika $K_p=1,5$, $K_i=4,75$

4.4 Penentuan nilai penguatan (Gain)

a. Tujuan

Pengujian ini bertujuan untuk mencari nilai gain yang berfungsi sebagai penguat.

b. Peralatan yang dipakai:

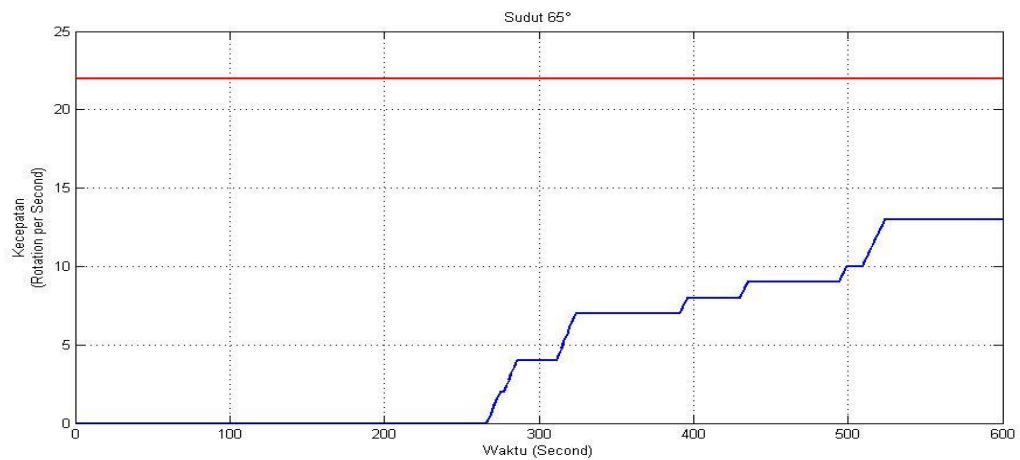
- Laptop yang sudah terinstall program Arduino
- Kompor gas portable
- Motor DC servo
- Arduino Uno
- Mini Steam Turbine
- Sensor Rotary encoder

c. Prosedur Pengujian

1. Mengkoppel motor Servo dengan tuas kompor gas
2. Menghubungkan Motor Servo MG995 ke pin V_{cc} , GND, dan pin 9 yang ada pada Arduino Uno
3. Menghubungkan Sensor Rotary encoder ke pin V_{cc} , GND, dan pin D0
4. Membuat program arduino buat tiap pengujian.
5. Mengunduh program sesuai dengan tipe pengujiannya.
6. Catat, amati, dan analisis respon dari sistem setelah mengunduh program sesuai dengan tipe pengujiannya.

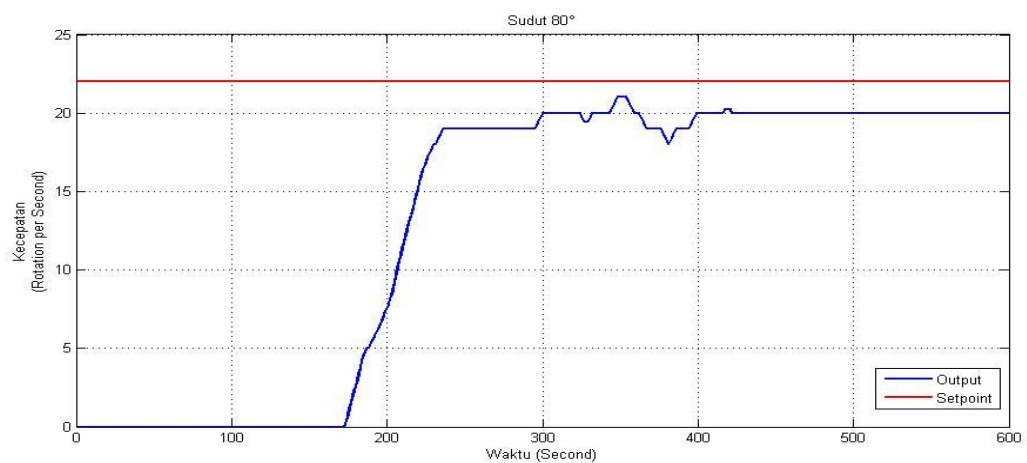
d. Hasil Pengujian

- Grafik hasil pengujian selama 10 menit ketika servo diatur dengan sudut 65° ditunjukkan dalam Gambar 4.10.



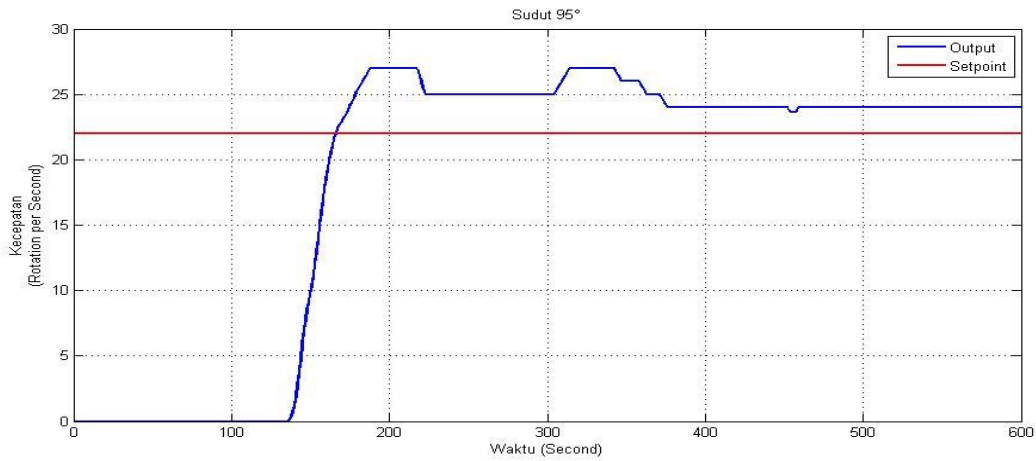
Gambar 4. 10 Grafik perubahan suhu saat sudut servo 65°

- Grafik hasil pengujian selama 10 menit ketika servo diatur dengan sudut 80° ditunjukkan dalam Gambar 4.11.



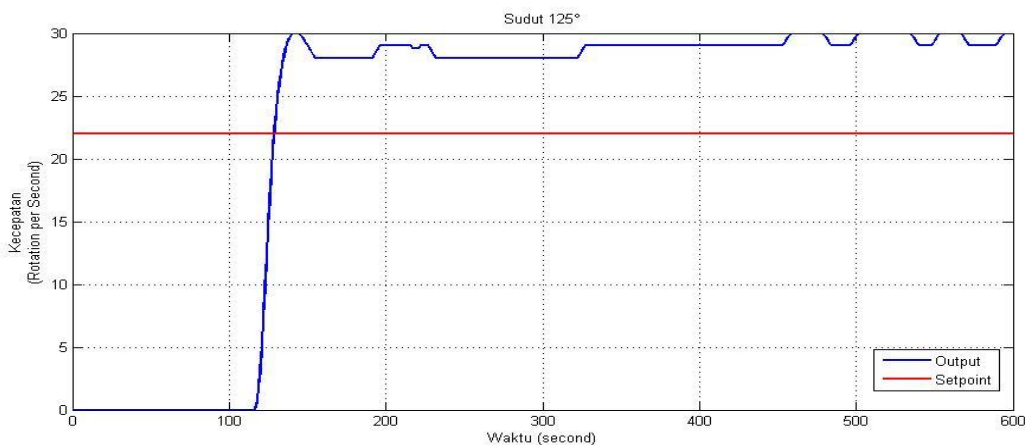
Gambar 4. 11 Grafik perubahan suhu saat sudut servo 80°

- Grafik hasil pengujian selama 10 menit ketika servo diatur dengan sudut 95° ditunjukkan dalam Gambar 4.12.

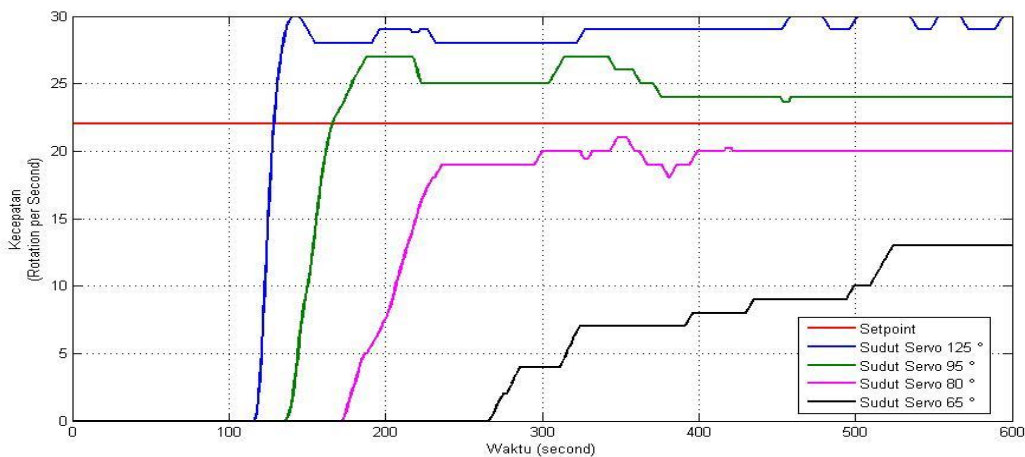


Gambar 4. 12 Grafik perubahan suhu saat sudut servo 95°

- Grafik hasil pengujian selama 10 menit ketika servo diatur dengan sudut 125° ditunjukkan dalam Gambar 4.13.



Gambar 4. 13 Grafik perubahan suhu saat sudut servo 125°

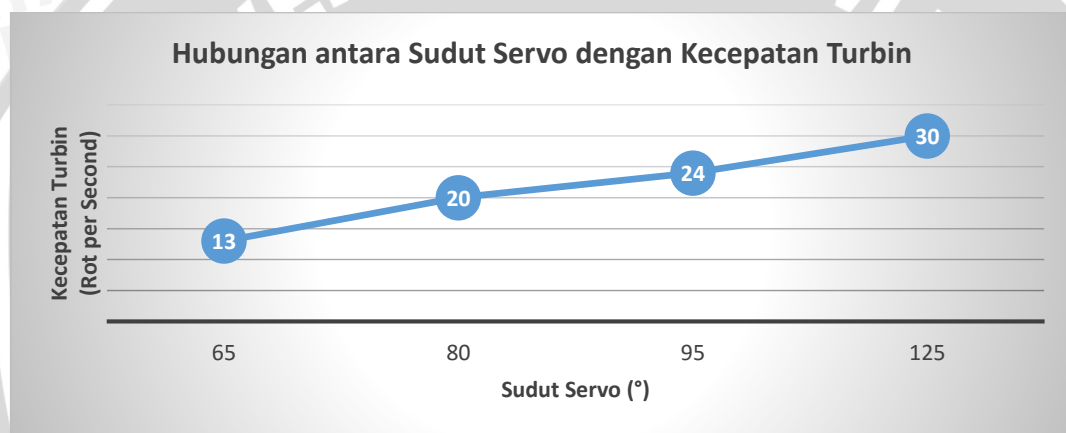


Gambar 4. 14 Grafik perubahan Kecepatan Turbin saat sudut servo 65°, 80°, 95°, dan 125°

Berdasarkan Gambar 4.14 dapat disimpulkan bahwa semakin besar sudut servo yang diberikan maka semakin besar pula kecepatan maksimal turbin yang dicapai dan juga semakin cepat pula proses terbentuknya uap

Tabel 4.4 Grafik perubahan kecepatan turbin terhadap perubahan sudut servo

Sudut	Kecepatan Turbin
65	13
80	20
95	24
125	30



Gambar 4. 15 Grafik hubungan antara perubahan sudut servo dan perubahan kecepatan turbin

Dari perhitungan sesuai dengan grafik dan tabel diatas maka di dapatkan nilai *gain* adalah :

$$m = \frac{y_4 - y_1}{x_4 - x_1} = \frac{30 - 13}{125 - 65} = 0.283333$$

4.5 Pengujian Keseluruhan Sistem tanpa Gangguan

a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja sistem dengan mengamati respon sistem terhadap setpoint ketika mendapatkan gangguan berupa perubahan suhu yang turun drastis.

b. Peralatan yang digunakan:

- Laptop yang sudah terinstall program Arduino
- Kompor gas portable

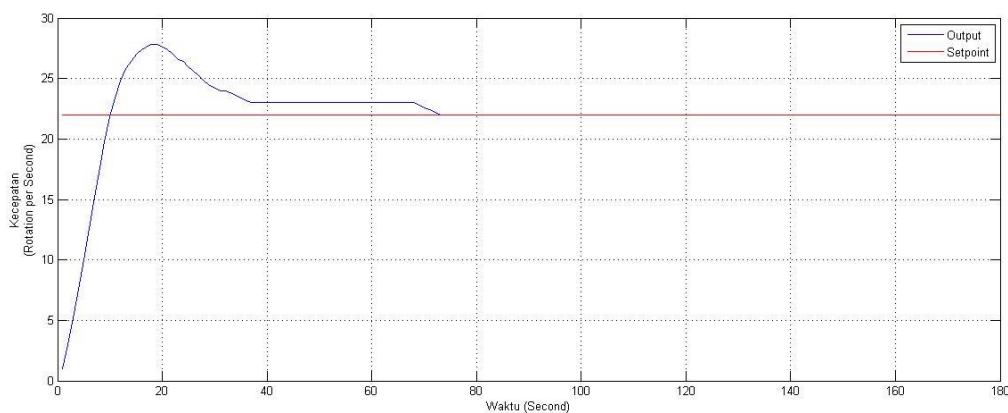
- Motor DC servo
- Arduino Uno
- Mini *Steam Turbine*
- Sensor *Rotary encoder*

c. Prosedur Pengujian

1. Mengkoppel motor Servo dengan tuas kompor gas
2. Menghubungkan Motor Servo MG995 ke pin V_{cc} , GND, dan pin 9 yang ada pada Arduino Uno
3. Menghubungkan Sensor *Rotary encoder* ke pin V_{cc} , GND, dan pin D0
4. Membuat program arduino buat tiap pengujian.
5. Mengunduh program sesuai dengan tipe pengujiannya.
6. Catat, amati, dan analisis respon dari sistem setelah mengunduh program sesuai dengan tipe pengujiannya.

d. Hasil Pengujian

Setelah melaksanakan prosedur pengujian didapatkan respon sistem dari *plant*. Grafik hasil respon sistem seperti pada Gambar 4.16.



Gambar 4. 16 Respon Sistem Tanpa Gangguan dengan Menggunakan $K_p = 1,5$ dan $K_i = 4,75$

Sesuai dengan hasil grafik dalam Gambar 4.16 dapat disimpulkan bahwa sistem dengan nilai K_p 1.5 dan $K_i=4,75$ mencapai *setpoint* dengan *settling time* 41 detik. Dari respon sistem di atas terjadi *overshoot* sebesar 1.272727 %.

Berdasarkan hasil pengujian dalam Gambar 4.16, diperoleh kinerja sistem antara lain:

1. t_s (*settling time*) yaitu waktu yang diperlukan sistem untuk mencapai nilai akhir ketika *steady*. t_s berdasarkan pengujian adalah 36 detik. *Settling time* didapat ketika kecepatan *turbine* telah mencapai 22 rotation per second.
2. Perhitungan nilai *error steady state* pada sistem didapatkan nilai sebesar 1,1208%

$$\%Ess = \frac{\text{rata} - \text{rata kecepatan steady} - \text{Setpoint}}{\text{Setpoint}} \times 100\%$$

$$\%Ess = \frac{22.24658 - 22}{22} \times 100\%$$

$$\%Ess = 1.1208\%$$

4.6 Pengujian Keseluruhan Sistem dengan Gangguan

a. Tujuan

Pengujian ini bertujuan untuk mengetahui kinerja sistem dengan mengamati respon sistem terhadap setpoint ketika mendapatkan gangguan berupa perubahan suhu yang turun drastis.

b. Peralatan yang digunakan:

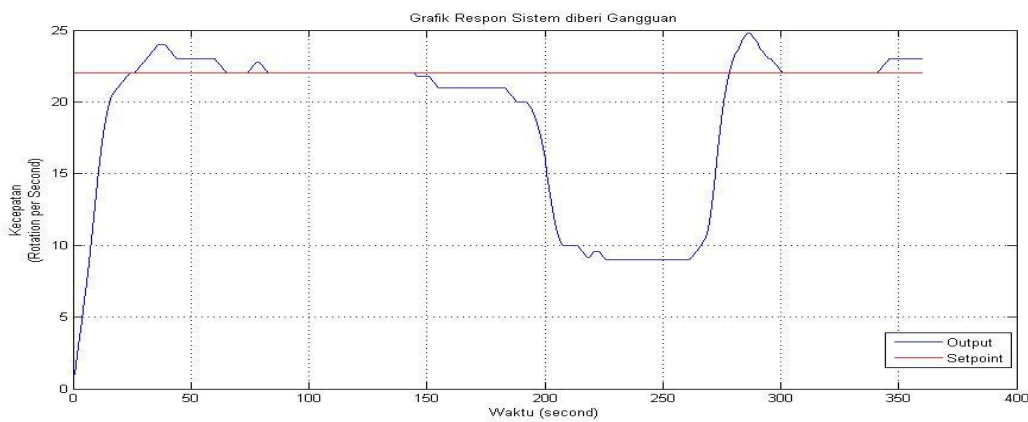
- Laptop yang sudah terinstall program Arduino
- Kompor gas portable
- Motor DC servo
- Arduino Uno
- Mini Steam Turbine
- Sensor Rotary encoder

c. Prosedur Pengujian

1. Mengkopel motor Servo dengan tuas kompor gas
2. Menghubungkan Motor Servo MG995 ke pin V_{cc} , GND, dan pin 9 yang ada pada Arduino Uno
3. Menghubungkan Sensor Rotary encoder ke pin V_{cc} , GND, dan pin D0
4. Membuat program arduino buat tiap pengujian.
5. Mengunduh program sesuai dengan tipe pengujiannya.
6. Catat, amati, dan analisis respon dari sistem setelah mengunduh program sesuai dengan tipe pengujiannya.

d. Hasil Pengujian

Setelah melaksanakan prosedur pengujian didapatkan respon sistem dari *plant*. Grafik hasil respon sistem seperti pada Gambar 4.17.



Gambar 4. 17 Respon Sistem diberi Gangguan dengan Menggunakan $K_p = 1,5$ dan $K_i = 4,75$

Sesuai dengan hasil grafik dalam Gambar 4.17 dapat disimpulkan bahwa sistem dengan nilai K_p 1.5 dan $K_i=4,75$ mengalami perubahan kondisi setelah diberi gangguan dengan menambahkan air dengan temperature normal kedalam boiler sebanyak 300ml. Sistem akan kembali ke keadaan mantap ketika kecepatan turbin sudah sesuai dengan *setpoint*. *Recovery time* pada sistem adalah 151 detik dan error steady state sebesar 0.982%.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil perancangan dan pengujian alat yang telah dilakukan, maka didapatkan:

1. Sistem pengendalian kadar gas dengan menggunakan metode *hand tuning* untuk menentukan nilai parameter kontroler P, didapatkan parameter P terbaik yaitu $K_p = 1.5$ dan parameter I terbaik yaitu $K_i = 4,75$ dengan nilai *setpoint*-nya 22 Rotation per Second.
2. Alat ini telah mencapai hasil sesuai dengan yang diharapkan, di mana kecepatan turbin dapat dikendalikan sesuai dengan *setpoint*. Waktu yang diperlukan sistem untuk mencapai keadaan mantap adalah selama 41 detik dengan *error steady state* 1,1208% dibandingkan tanpa menggunakan kontroler yang membutuhkan waktu sebesar 125 detik untuk mencapai keadaan mantap.

5.2 Saran

Dalam perancangan dan pembuatan alat ini tentunya tidak terlepas dari kesalahan maupun kekurangan. Diharapkan untuk kedepannya diperbaiki kinerja alat dan pengembangan lebih lanjut agar dapat mencapai sistem yang optimal. Contohnya seperti, dengan menggunakan *boiler* dengan material mild steel dan bejana benar benar tertutup agar tidak ada kebocoran uap yang terjadi sehingga uap yang dihasilkan lebih bertenaga dan mampu menggerakkan turbin yang dikopel dengan mini generator seperti motor DC. Menambahkan pengendalian suhu, level air atau tekanan *boiler* dan pengaruhnya terhadap listrik yang dihasilkan oleh mini *Steam Turbine Generator*.

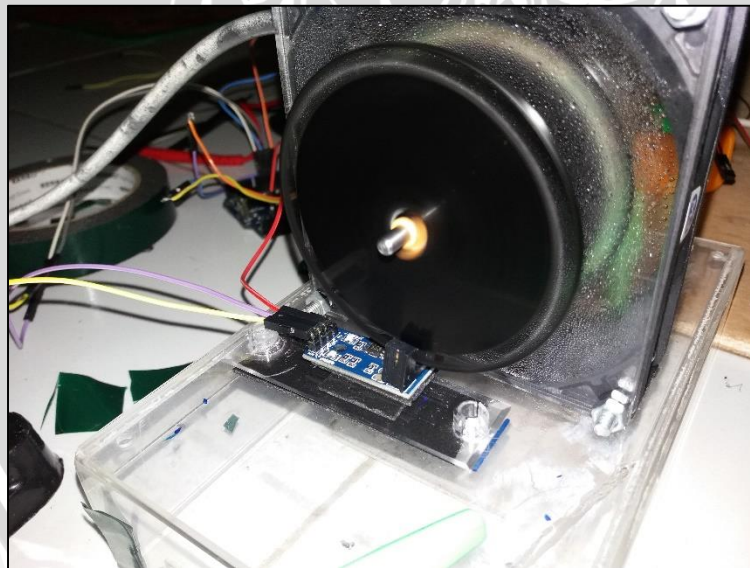


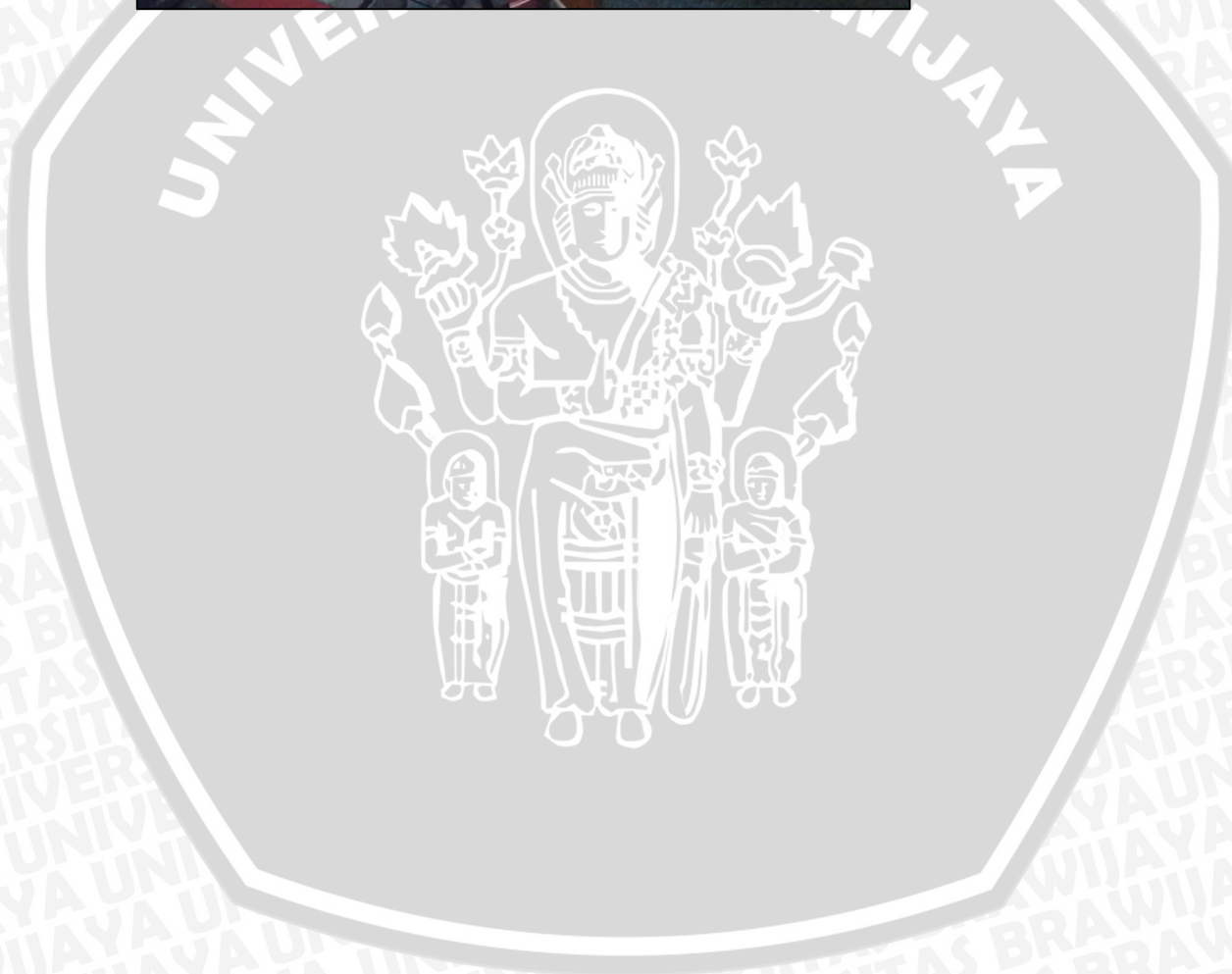
DAFTAR PUSTAKA

1. Laksono,Edi. 1995. TEKNIK KONTROL AUTOMATIK (SISTEM PENGATURAN) Jilid 1. Bandung : Penerbit Erlangga.
2. Philip, C. L. & Harbor, R. D. 1996. *Feedback Control System*. Diterjemahkan oleh Widodo.R.J. Prentice Hall. New Jersey.
3. Puspawardiana,N. 2014. *PENGATURAN POSISI MOTOR SERVO PADA MINIATUR ROTARY PARKING*. Laporan Skripsi, Teknik Elektro. Universitas Brawijaya Malang.
4. Ogata, Katsuhiko. 1985. Teknik Kontrol Automatik (Sistem Pengaturan) Jilid 1. Jakarta: Erlangga.
5. Mu'amar,A., Cordova,H., dan Fitri,A. PERANCANGAN SISTEM CONTROL LEVEL DAN PRESSURE PADA *BOILER* DI WORKSHOP INSTRUMENTASI BERBASIS DCS CENTUM CS3000 YOKOGAWA. Teknik Fisika. Institut Teknologi Surabaya. Surabaya.
6. HIMA Teknik Pengelasan PPNS. 2016. <http://hima-t1.ppns.ac.od/?p=1249/> (Pada tanggal 1 November 2016)
7. Anonim. <http://brainy-bits.com/tutorials/speed-sensor-with-arduino/> (Pada tanggal 28 juli 2016)



LAMPIRAN I FOTO ALAT





LAMPIRAN II

Listing Program

1. PROGRAM PEMBACAAN KECEPATAN

```
#define pinpwm 3
volatile int pulsa = 0;
float rpm = 0;
void setup() {
    pinMode(pinpwm, OUTPUT); //inisialisasi timer
    noInterrupts();
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;

    OCR1A = 62500; // compare match register 16MHz/256/50Hz/1000ms
    TCCR1B |= (1 << WGM12); // CTC mode
    TCCR1B |= (1 << CS12); // 256 prescaler
    TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
    attachInterrupt(0, rpm_fan, FALLING);
    interrupts();
    Serial.begin(9600);
}

ISR(TIMER1_COMPA_vect) // timer compare interrupt service routine
{
    detachInterrupt(0);
    rpm = (pulsa/60);
    Serial.println(rpm);

    pulsa = 0;
    attachInterrupt(0, rpm_fan, FALLING);
}
```

```

    }

    void loop()
    {
        // put your main code here, to run repeatedly:
    }

```

```

    void rpm_fan()
    {
        pulsa++;
    }

```

2. PROGRAM KESELURUHAN

```

#include <PID_v1.h>
#include <Servo.h>
#define pinpwm 3
volatile int pulsa = 0;
float rpm = 0;
int tick_timer;
int tick_servo;
Servo myservo;

unsigned long time;
double Setpoint, Input, Output; // these are just variables for storing values
// Tuning parameters
float Kp=1.5; //Initial Proportional Gain
float Ki=4.75; //Initial Integral Gain
float Kd=0; //Initial Differential Gain
float rpm;
float K=0.28;
PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);

// Timestamp
unsigned long serialTime; //this will help us know when to talk with processing

```

```

const int sampleRate = 1; // Variable that determines how fast our PID loop runs
const long serialPing = 500; //This determines how often we ping our loop
unsigned long now = 0; //This variable is used to keep track of time
unsigned long lastMessage = 0; //This keeps track of when our loop last spoke to serial

```

```

void setup() {
  pinMode(pinpwm, OUTPUT);
  //inisialisasi timer
  noInterrupts();
  ASSR=(0<<EXCLK) | (0<<AS2);
  TCCR2A=(0<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) | (0<<COM2B0) |
  (1<<WGM21) | (0<<WGM20);
  TCCR2B=(0<<WGM22) | (1<<CS22) | (1<<CS21) | (0<<CS20);
  TCNT2=0x00;
  OCR2A=0xFF;
  OCR2B=0x00;
  attachInterrupt(0, rpm_fan, FALLING);
  interrupts();
  Serial.begin(9600);
  myservo.attach(9);

  // Timer/Counter 2 Interrupt(s) initialization
  TIMSK2=(0<<OCIE2B) | (1<<OCIE2A) | (0<<TOIE2); // pengaturan timer !
}

```

```

ISR(TIMER2_COMPA_vect)      // setiap 4ms sekali
{
  tick_timer++;
  //Serial.println(rpm);
  if(tick_timer>=250) // 250 x 4 ms = 1000 ms = 1 s
  { detachInterrupt(0);
    rpm = (pulsa/60);
    Serial.println(rpm);
    pulsa = 0;
  }
}

```

```

attachInterrupt(0, rpm_fan, FALLING);
tick_timer=0;

Input = rpm;
Setpoint = 30; // Read the SetPoint value from the potentiometer
myPID.SetMode(AUTOMATIC); // Turn on the PID loop as automatic control
myPID.SetOutputLimits(0, 1300);
myPID.SetSampleTime(sampleRate); // Sets the sample rate
lastMessage = millis(); // timestamp
}
}

void loop() {
// tick_servo++;
//Serial.println(rpm);
attachInterrupt(0, rpm_fan, FALLING); // pelajarin fungsi gunanya
myservo.write(65 + rpm*2);
delay(10);
// if(tick_servo>180){
// tick_servo=0;
// }
}

void rpm_fan()
{
pulsa++;

```