BAB IV
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan dalam skripsi ini bertujuan untuk merancang beberapa perangkat maupun alat secara keseluruhan. Perancangan perangkat tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak. Sedangkan pembuatan bertujuan untuk menghasilkan semua perangkat pendukung maupun alat secara keseluruhan.

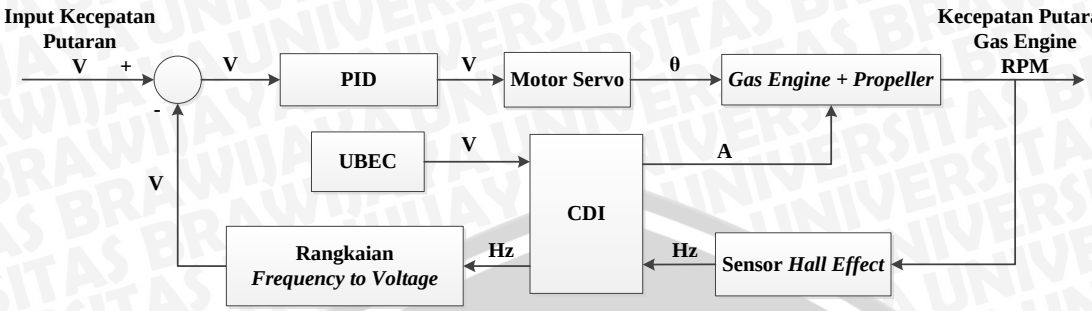
4.1 Spesifikasi Sistem

Spesifikasi alat yang di rancang adalah sebagai berikut:

1. *Gas Engine* 2 tak dengan *displacement* sebesar 9 CC
2. CDI (Capacitor Discharge Ignition) sebagai alat bantu sistem pengapian pada mesin pembakaran dalam.
3. UBEC (*Universal Battery Elimination Circuit*) sebagai pengkondisi sinyal tegangan masukan CDI sebesar 4,8-6 V.
4. Putaran *Gas Engine* didapatkan dari proses pembakaran dalam melalui campuran udara dan bahan bakar cair (oktan 90).
5. RPM yang dihasilkan *Gas Engine* yaitu 4.000-10.000.
6. *Propeller* pada ujung *shaft* motor menggunakan Type S2 Series 11 x 5 inch.
7. Arduino Uno berbasis mikrokontroler ATmega328.
8. Menggunakan sensor Sensor *Hall effect.*
9. Motor *Servo* dengan maksimum tegangan masukan 4,8 V dan memiliki torsi sebesar 3,1 kg-cm.
10. *Windtunnel* sebagai alat pengujian sistem.

4.2 Diagram Blok Sistem

Dalam skripsi ini dibuat diagram blok agar dalam pengerjaan dapat dilakukan sesuai dengan rangcangan sistem. Adapun diagram blok tersebut dapat dilihat pada Gambar 4.1:



Gambar 4.1 Diagram Blok Sistem

Keterangan dari diagram blok dalam Gambar.4.1:

- *Input/ setpoint* berupa kecepatan putaran diberikan melalui *program* pada Arduino Uno.
- Kemudiaan input diolah dan menghasilkan sinyal kontrol berupa pwm yang kemudian akan mejadi masukan untuk mengerakan motor *servo.*
- *Gas Engine* menggunakan CDI sebagai sistem pengapian pada proses pembakaran dalam yang dicatu oleh UBEC sebagaimana digunakan untuk mengondisikan tegangan agar lebih stabil.
- Sinyal dari motor *servo* tadi kemudian menggerakkan *throttle* pada *Gas Engine* sehingga mengatur putaran sesuai *setpoint* yang kemudian akan menggerakkan *propeller.*
- Keluaran putaran (*present value*) kemudian di baca oleh sensor *Hall effect* yang kemudian diproses melalui CDI hingga mendapatkan keluaran hasil pembacaan dalam bentuk digital.
- Keluaran dalam bentuk digital tadi diubah menjadi bentuk analog oleh rangkaian *Frequency to Voltage.*
- Hasil akhir pembacaan sensor kemudian dikurangkan dengan *input/ setpoint* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi.

4.3 Perancangan Perangkat Keras

Perangkat keras terdiri dari *Gas Engine*, sensor *hall effect*, rangkaian *frequency to Analog*, CDI (*Capacitor Discharge Ignition*), UBEC (*Universal Battery Elimination Circuit*), *propeller* (baling-baling), *motor servo*, dan *Windtunnel* (terowongan angin), penentuan parameter penguatan kontroler, modul Arduino Uno.

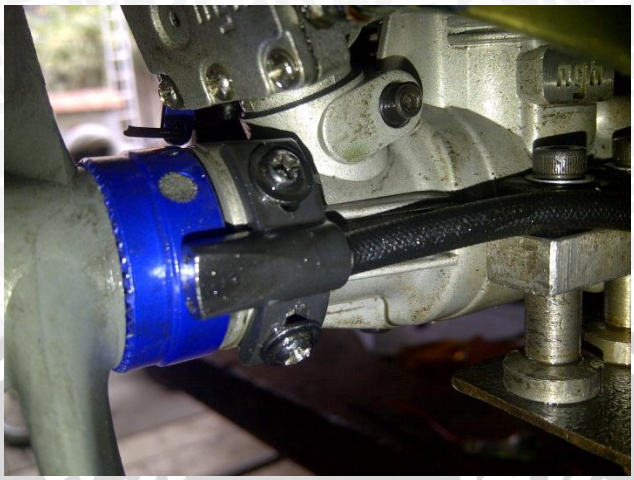
4.3.1 Gas Engine

Gas Engine merupakan motor yang menghasilkan putaran melalui proses pembakaran dalam, yaitu pencampuran antara bahan bakar cair (oktan 90) dengan udara. Pada dasarnya putarannya dikendalikan dengan megatur buka-tutup katup melalui *throttle*-nya. Spesifikasi yang dipilih berdasarkan kebutuhan standar untuk *aeromodeling*, mesin dengan tipe 2 tak, kapasitas *displacement* 9cc ini memiliki kekuatan maksimal 0,8 HP/ 15.000 RPM. Pada motor ini sensor *hall effect* sudah terpasang menjadi satu.

***Gambar 4.2 Gas Engine* 2-tak dengan displacement 9cc.**

4.3.2 Sensor *Hall effect*

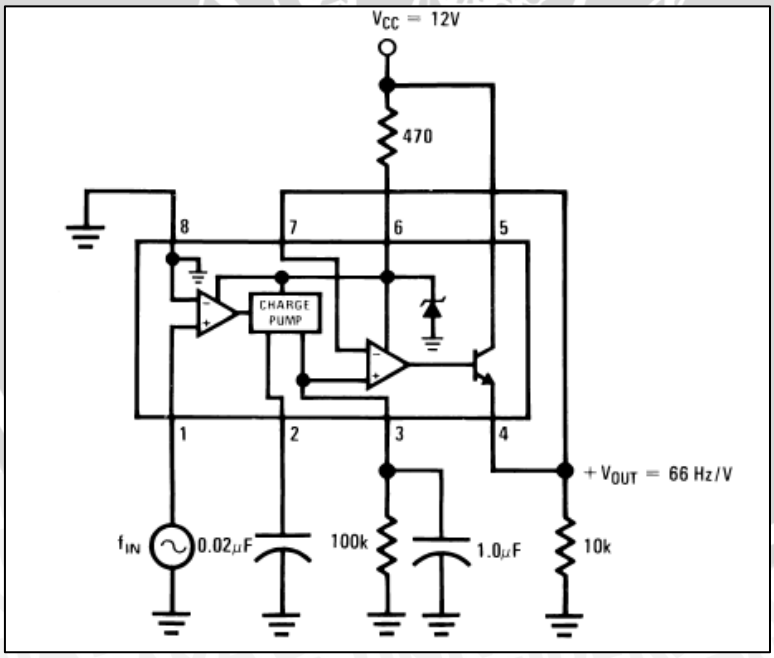
Seperti yang dijelaskan diatas, sensor *hall effect* yang digunakan pada perancangan alat ini terpasang menjadi satu pada *Gas Engine*, bekerja sebagai pengolah sinyal yang dihasilkan dari magnet/ reluktor yang berputar. Sinyal kemudian diolah oleh CDI menghasilkan keluaran berupa sinyal digital, kemudian diolah kembali oleh rangkaian *Frequency to Voltage* untuk menghasilkan keluaran berupa sinyal analog.



Gambar 4.3 Sensor *Hall effect* pada *Gas Engine*

4.3.3 Rangkaian *Frequency to Voltage*

Rangkaian *Frequency to Voltage* digunakan untuk mengubah sinyal digital dari keluaran sensor *hall effect* yang diproses oleh CDI menjadi sinyal analog agar dapat dimasukkan pada board Arduino Uno. Rankaian ini menggunakan *integrated circuit* (IC) LM2917 yang memiliki tegangan kerja +12 volt DC hingga +24 volt DC. Skema rangakainnya dapat ditunjukkan pada Gambar 4.4.



Gambar 4.5 Skematik Rangkaian *Frequency to Voltage*
Sumber: *Datasheet LM2917*

Berdasarkan pada *datasheet* dengan rangkaian seperti pada gambar 4.4, maka tegangan keluaran dapat dihitung dengan menggunakan persamaan:

$$V_{out} = \frac{f_{in}}{66} \quad \text{.....(4-1)}$$

Atau secara umum perhitungan yang digunakan pada rangkaian yang mengunakan IC LM2907/2917 dapat dijelaskan pada persamaan berikut.

$$V_{out} = f_{in} \times V_{cc} \times R_1 \times C_1 \quad \text{.....(4-2)}$$

dimana :

Fin = Frekuensi sinyal input (Hz)

Vcc = Tegangan sumber yang digunakan (volt)

R1 = Resistor pada pin 3 IC LM2917 (Ohm)

C1 = Kapasitor pada pin 2 IC LM2917 (Farad)

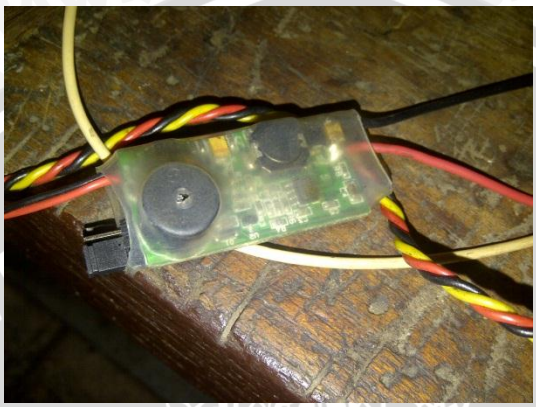
4.3.4 CDI (*Capacitor Discharge Ignition*)

CDI atau *Capacitor Discharge Ignition* disini digunakan untuk membantu sistem pengapian pada proses pembakaran dalam, sehingga semakin maksimal pengapian maka busi juga akan memantik campuran gas dalam ruang bakar secara maksimal juga. Bekerja pada tegangan 4,8 – 6 V. Disamping itu CDI juga berfungsi sebagai pengolah sinyal yang dihasilkan dari sensor *hall effect*.

Gambar 4.6 *Capacitor Discharge Ignition* (CDI)

4.3.5 UBEC (*Universal Battery Elimination Circuit*)

UBEC (*Universal Battery Elimination Circuit*) berfungsi sebagai pengondisi sinyal tegangan agar lebih stabil ketika disalurkan pada CDI. Rangkaian ini bekerja pada tegangan 6 - 23V dan menghasilkan output 5,1 atau 6,1 V.



Gambar 4.7 UBEC (*Universal Battery Elimination Circuit*)

4.3.6 Pemilihan *Propeller*

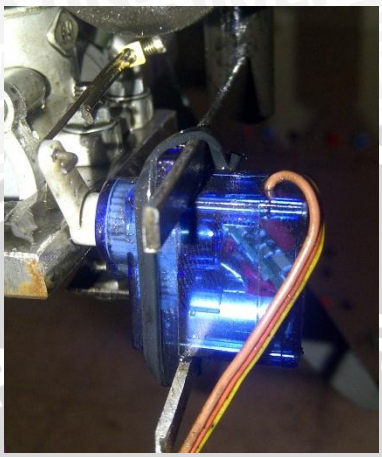
Propeller yang digunakan pada ujung *shaft* motor adalah propeller tipe S2 *series* 11 x 5 inch. Pemilihan ini didasarkan pada rekomendasi pabrik yaitu *propeller* 11 x 5 inch atau 11 x 6 inch jika digunakan pada *Gas Engine displacement* 9cc.

Gambar 4.8 Propeller tipe S2 *series* 11 x 5 inch.

4.3.7 Motor *Servo*

Motor *servo* yang digunakan dalam perancangan kali ini berguna sebagai pengatur buka-tutup *throttle* pada *Gas Engine*. *Throttle* sendiri berguna untuk menaikkan dan merunkan kecepatan putaran pada *Gas Engine*. Motor *servo* ini juga

dapat langsung terhubung ke Arduino Uno tanpa menggunakan driver karena bekerja pada maksimum tegangan masukan 4,8 V dan memiliki torsi sebesar 3,1 kg-cm.



Gambar 4.9 Motor *Servo*

4.3.8 *Windtunnel* (Terowongan Angin)

Windtunnel/ terowongan angin ini memiliki fungsi untuk menguji *Gas Engine* apakah dapat stabil ketika mendapatkan gangguan perubahan aliran angin. Selain itu juga mempunyai fungsi sebagai peyangga *Gas Engine* saat melakukan penyalaan mesin ataupun saat melakukan setting. Spesifikasi *Windtunnel*/ terowongan angin ini memiliki diameter dalam 32 cm dan panjang 70 cm.

Gambar 4.10 *Windtunnel* (Terowongan Angin)

4.3.9 Penentuan Nilai Penguatan Kontroler

Percobaan pemodelan *Gas Engine* menggunakan metode Identifikasi MATLAB dengan membandingkan input dan output tidak memberikan hasil sesuai dengan yang

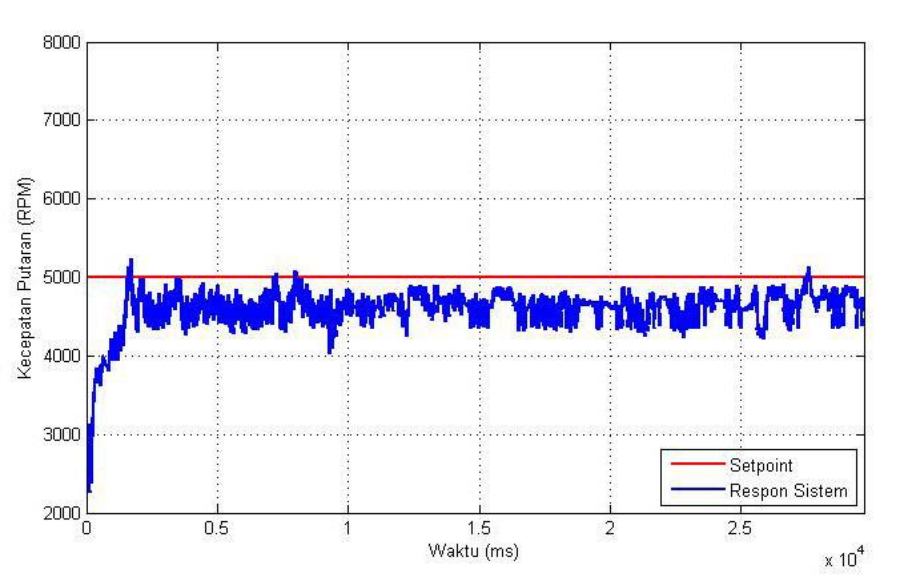
diharapkan, dari beberapa percobaan yang telah dilakukan hanya didapatkan *best-fit* terbaik yaitu sebesar 2,592%.

Kemudian percobaan lainnya dengan penerapan metode Ziegler-Nichols langsung pada *plant* juga tidak memberikan hasil yang baik. Respon tidak dapat memberikan osilasi berkesinambungan pada pemberian nilai Kp dengan nilai berapapun sehingga nilai Kcr *(critical gain*) dan Pcr (*critical period*)-nya tidak dapat ditentukan. Dalam hal ini nilai penguatan kontroler (Kp, Ki, Kd) akhirnya ditentukan dengan menggunakan metode *hand tuning* (*hand eksperimen*) untuk mendapatkan hasil respon sesuai dengan yang diinginkan.

Tuning eksperimen adalah proses yang dilakukan untuk mendapatkan hasil kontroler yang optimal dengan cara suatu percobaan. Inti dari *tuning* eksperimen adalah menentukan nilai dari tiga buah parameter yang terdapat pada kontroler PID yaitu konstanta proporsional (Kp), konstanta integral (Ki) dan konstanta diferensial (Kd).

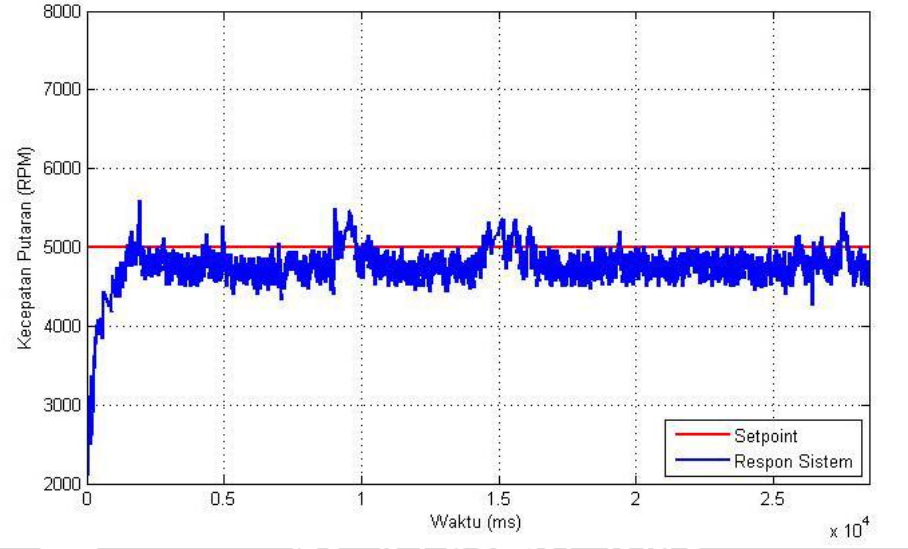
Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai Kp hingga didapatkan respon sistem yang mendekati *setpoint* 5000 RPM. Pengambilan data dilakukan dengan pembacaan sensor yang masuk dalam serial monitor Arduino Uno.

Untuk *tuning* parameter PID dengan nilai Kp = 0.5, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.16.



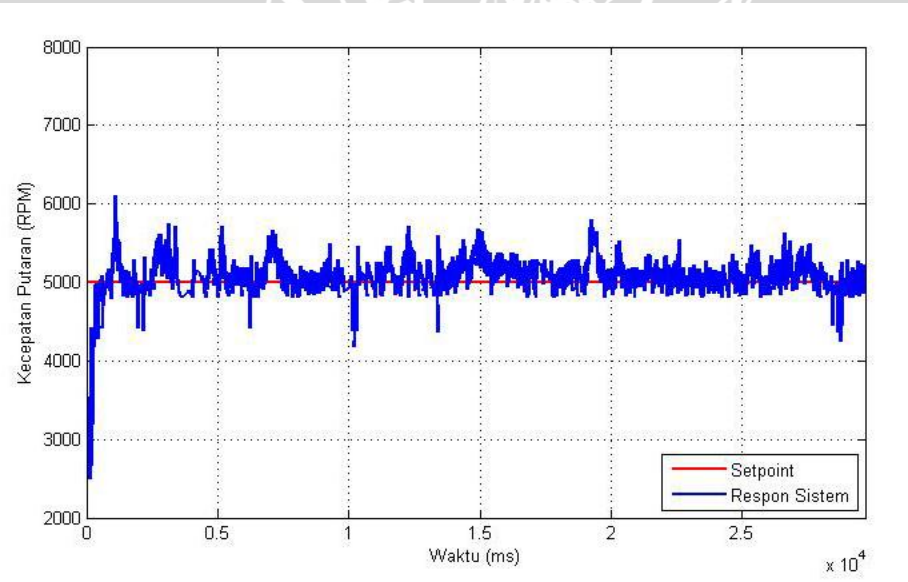
Gambar 4.16 Respon Sistem dengan nilai Kp = 0.5, Ki = 0, dan Kd = 0

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.17.



Gambar 4.17 Respon Sistem dengan nilai Kp = 1, Ki = 0, dan Kd = 0

Untuk *tuning* parameter PID dengan nilai Kp = 1.5, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.18.



Gambar 4.18 Respon Sistem dengan nilai Kp = 1.5, Ki = 0, dan Kd = 0

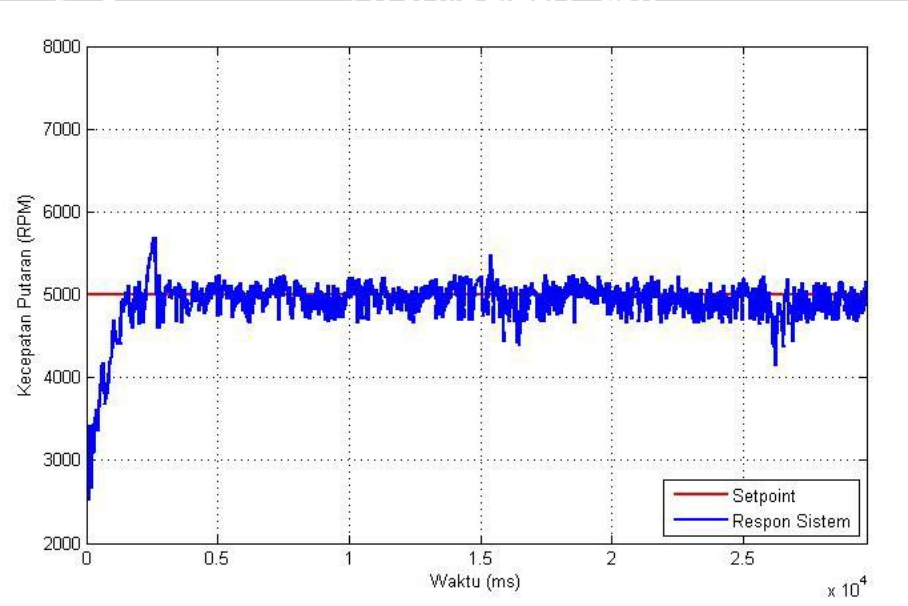
Dari hasil *tuning* degan berbagai parameter Kp yang berbeda maka didapatkan data seperti pada tabel 4.2

Tabel 4.2 Hasil Tuning Parameter Kp

No.	Kp	*ess* (%)
1.	0.5	19,24
2.	1	14,46
3.	1.5	16

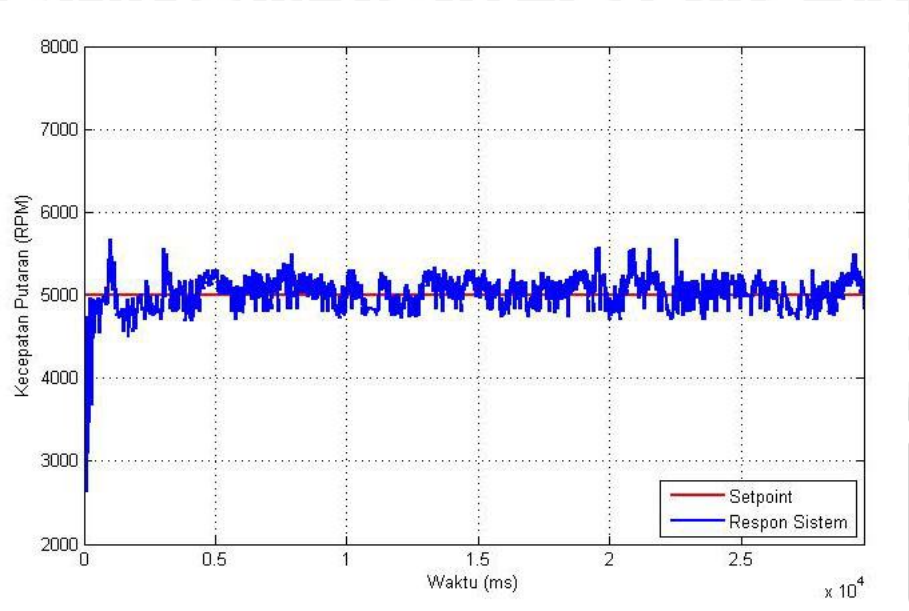
Pemilihan parameter Kp didapatkan dari data tabel diatas, yaitu parameter Kp= 1 karena dari tiga hasil *tuning* berbeda *ess* terkecil adalah pada saat Kp= 1 dan juga secara keseluruhan terletak dibawah *setpoint* sehingga bisa ditambahkan parameter Ki untuk dapat mendekati *setpoint* yang diinginkan. Setelah mendapatkan hasil Kp, maka dilanjutkan dengan mencari parameter nilai Ki untuk dapat memperbaiki respon sistem.

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.005, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.19.



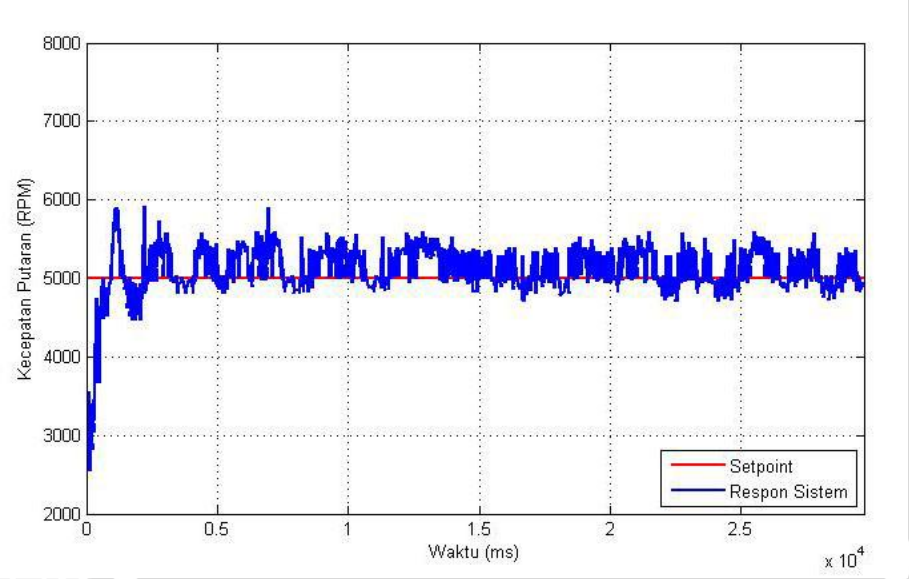
Gambar 4.19 Respon Sistem dengan nilai Kp = 1, Ki = 0.005, dan Kd = 0

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.20.



Gambar 4.20 Respon Sistem dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.015, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.21.



Gambar 4.21 Respon Sistem dengan nilai Kp = 1, Ki = 0.015, dan Kd = 0

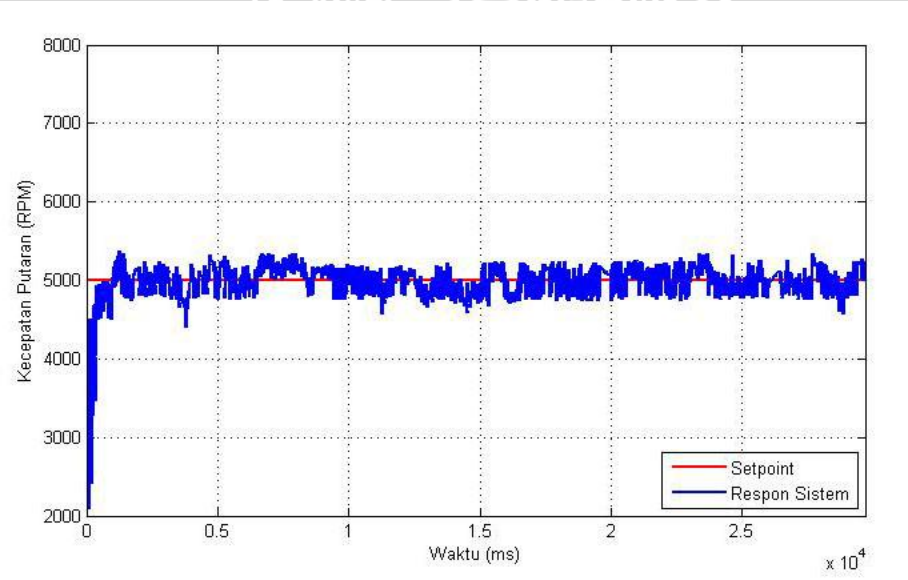
Dari hasil *tuning* degan berbagai parameter Ki yang berbeda maka didapatkan data seperti pada tabel 4.3

Tabel 4.3 Hasil Tuning Parameter Ki

No.	Kp	Ki	Mp (%)	*ess* (%)
1.	1	0.005	13,92	17,2
2.	1	0.01	15,4	13,6
3.	1	0.015	17,28	18,44

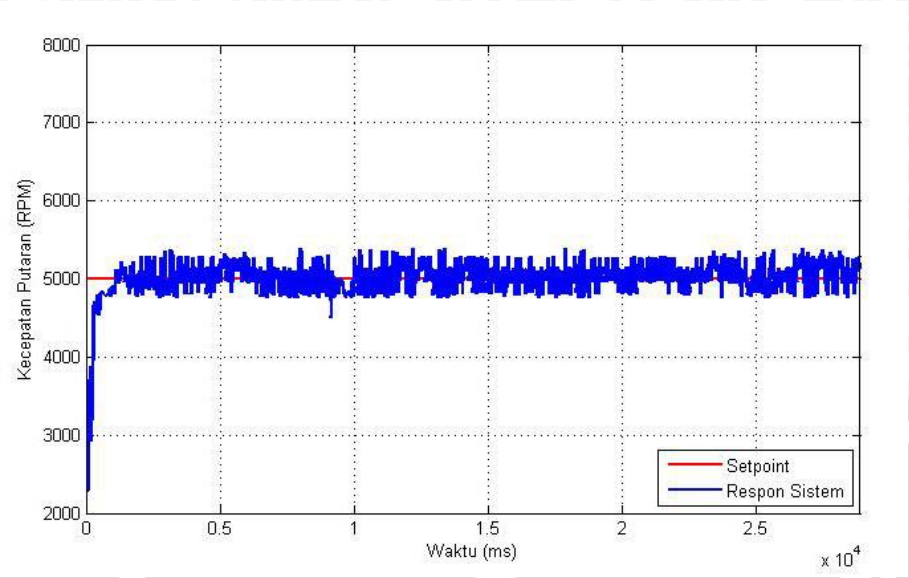
Pemilihan parameter Ki didapatkan dari data tabel diatas, yaitu parameter Ki= 0,01 karena dari beberapa hasil *tuning* berbeda *ess* terkecil adalah pada saat Ki= 0,01. Meskipun nilai *ess* selisihnya tidak terlalu jauh dari sebelum diberikan parameter Ki tetapi saat *steady* nilainya sudah berada di daerah *setpoint*, hal ini berbeda sebelum diberikan parameter Ki yang nilainya masih belum mendekati/ dibawah *setpoint*. Setelah mendapatkan hasil Ki, maka dilanjutkan dengan mencari parameter nilai Kd untuk dapat mengurangi *maximum overshoot* (Mp) pada respon sistem.

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,05 diperoleh grafik respon seperti pada Gambar 4.22.



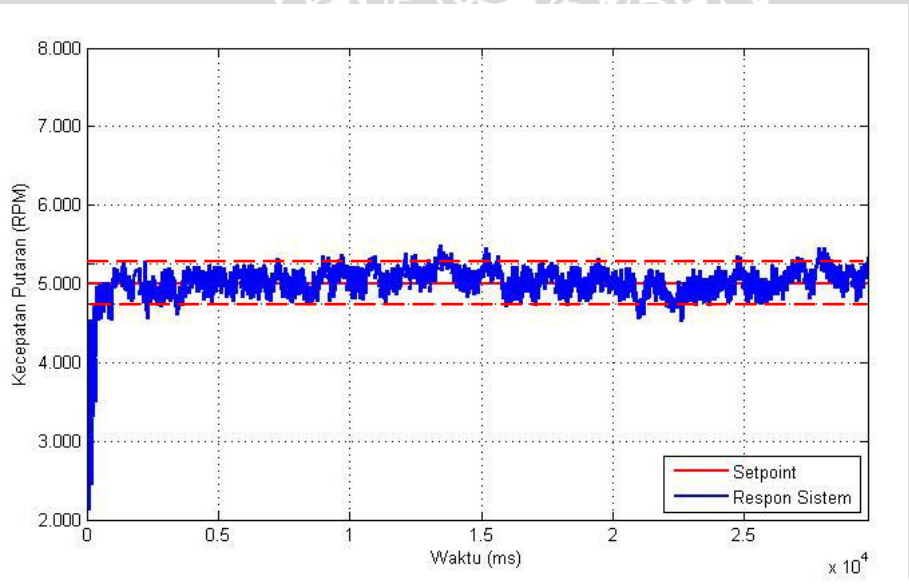
Gambar 4.22 Respon Sistem dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,05

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,1 diperoleh grafik respon seperti pada Gambar 4.23.



Gambar 4.23 Respon Sistem dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,1

Untuk *tuning* parameter PID dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,12 diperoleh grafik respon seperti pada Gambar 4.24.



Gambar 4.24 Respon Sistem dengan nilai Kp = 1, Ki = 0.01, dan Kd = 0,12

Dari hasil *tuning* dengan berbagai parameter Kd yang berbeda maka didapatkan data seperti pada tabel 4.4

Tabel 4.4 Hasil Tuning Parameter Kd

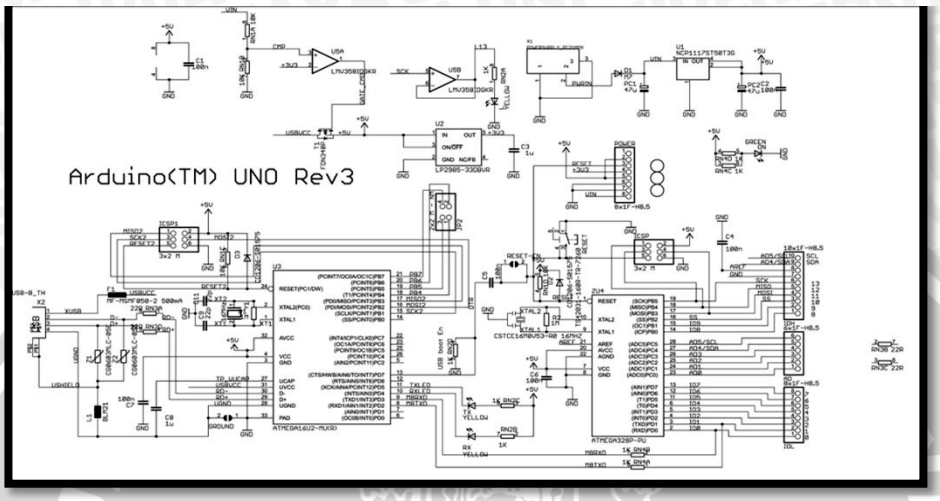
No	Kp	Ki	Kd	Td (ms)	Tr (ms)	Ts (s)	Tp (s)	Mp (%)
1.	1	0,01	0,05	138,2	329,65	1,6	1,25	7,54
2.	1	0,01	0,1	123,6	349,15	2,00	1,32	4,58
3.	1	0,01	0,12	137,9	324,91	1,6	1,05	3,96

Pemilihan parameter Kd didapatkan dari data tabel diatas, yaitu parameter Kd= 0,12 dengan nilai *maximum overshoot* (Mp) sebesar 3,96% yang mana lebih kecil dibandingkan dengan hasil *tuning* lainnya, saat Kd= 0,05 dan 0,1 yaitu sebesar 7,54% dan 4,58%. Terdapat beberapa nilai *ess* lebih dari toleransi 5% dari *setpoint* yaitu sebanyak 8,1%, hal ini dikarenakan adanya gangguan internal seperti bahan bakar dan udara yang kurang lancar. Tetapi secara keseluruhan sistem dapat memberikan respon keluaran yang baik.

Berdasarkan hasil tuning ketiga parameter Kp, Ki, dan Kd dengan menggunakan metode *Hand Tuning* (*Hand* Eksperimen), maka dapat ditentukan parameter penguatan kontroler yang akan digunakan pada sistem yaitu Kp = 1, Ki = 0.01, dan Kd = 0,12.

4.3.10 Modul Arduino Uno

Pada perancangan alat ini Arduino Uno R3 berbasis mikrokontroler ATmega328 digunakan sebagai pusat pengolah utama dalam melakukan proses pengendalian.



Gambar 4.25 Rangkaian Board Arduino Uno
Sumber: Arduino.cc

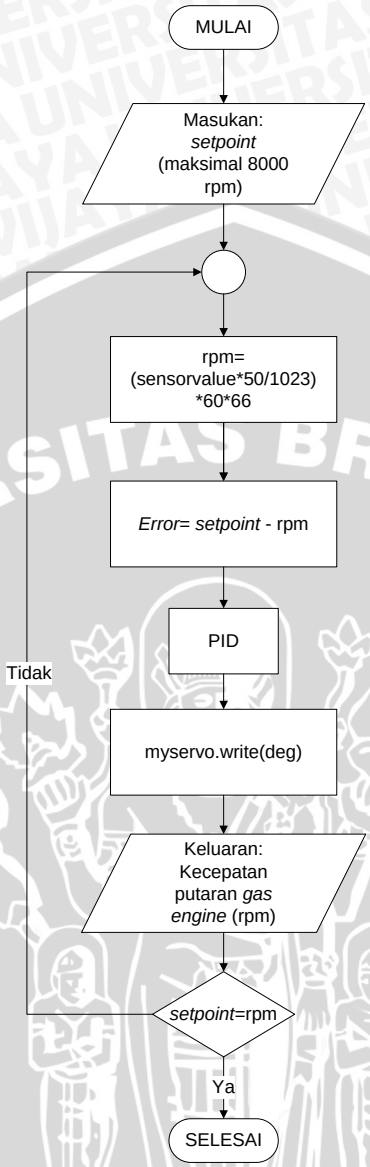
Berikut tabel fungsi masing - masing pin Arduino Uno R3 dapat dilihat dalam Tabel 4.5.

Tabel 4.5 Fungsi Pin Arduino Uno

No.	Pin	Fungsi
1	A1	Digunakan sebagai jalur masukan dari rangkaian *frequency to Voltage*
2	D9	Digunakan sebagai keluaran *motor servo*
3	GND	Jalur masukan GND seluruh sistem
4	5V	Jalur masukan 5V motor *servo*

4.4 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 menggunakan metode *Hand Tuning* (*Hand* Eksperimen) yang telah dimasukkan pada Arduino Uno yang digunakan. *Flowchart* perancangan perangkat lunak dapat dilihat pada Gambar 4.22.



Gambar 4.26 *Flowchart* Perangkat Lunak