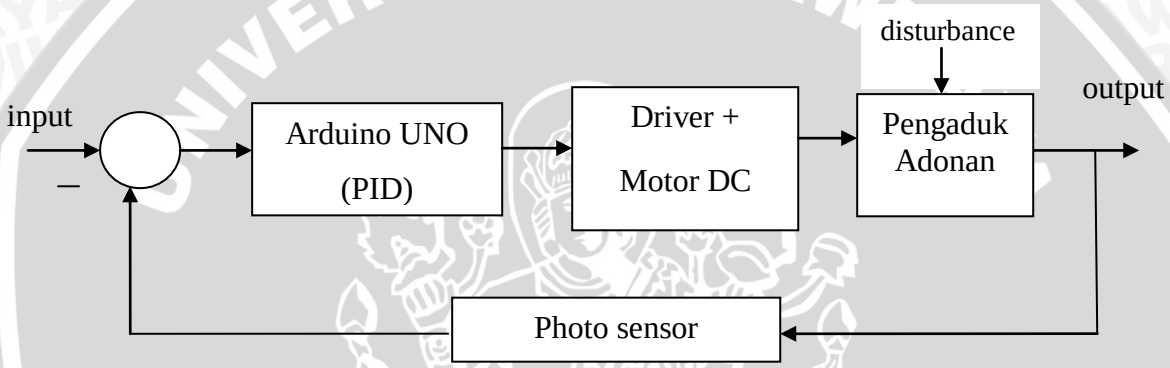
BAB IV
PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan alat pengaduk adonan mulai dari diagram blok sistem, desain mekanik, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga nantinya akan memudahkan dalam analisis sistem.

4.1 Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Blok diagram tersebut dapat dilihat dari Gambar 4.1 berikut.



Gambar 4.1 Blok Diagram Sistem

Keterangan dari blok diagram di atas adalah sebagai berikut :

- Sensor yang digunakan berupa Photo sensor BS5-T2M yang berfungsi mendeteksi kecepatan motor pada saat motor bergerak dan akan memberikan masukan berupa pulsa ke Arduino.
- Pusat pengendalian sistem menggunakan Arduino UNO yang memberikan keluaran berupa *duty cycle* PWM kepada *driver* motor.
- Motor yang digunakan adalah motor DC yang berfungsi sebagai aktuator.
- *Driver* motor menggunakan modul EMS 5A *H-Bridge*.

4.2 Prinsip Kerja Alat

Cara kerja alat adalah sebagai berikut :

- Menggunakan catu daya sebesar 12 volt untuk motor DC dan 5 volt untuk sensor *optocoupler*.
- *Photo sensor* BS5-T2M sebagai sensor kecepatan motor DC. *Photo sensor* akan memberikan keluaran berupa pulsa yang berubah-ubah sesuai dengan perubahan kecepatan.

- Tahap pertama yaitu ketika sistem diaktifkan motor akan berputar, sensor akan mendeteksi kecepatan yang terbaca kemudian memberikan masukan ke arduino berupa pulsa.
- Tahap berikutnya yaitu pulsa dari sensor akan diolah di arduino yang akan memberikan sinyal kontrol berupa PWM ke *driver* motor agar motor berputar sesuai dengan *setpoint* yang diinginkan.
- Ketika motor telah berputar dan mulai mengaduk adonan, maka *timer* akan aktif selama 45-60 menit sampai proses dianggap selesai.

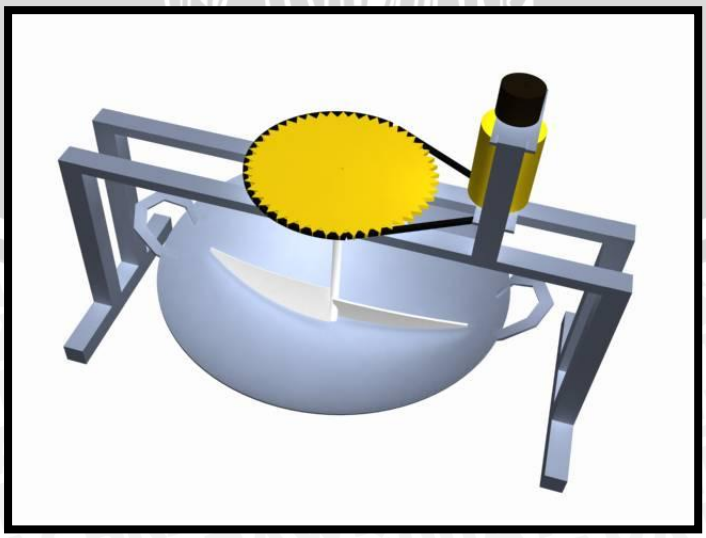
4.3 Perancangan Perangkat Keras (Hardware)

4.3.1 Spesifikasi Alat

Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Alat pengaduk adonan berbentuk silinder berbahan *stainless steel* dengan ukuran:
   Tinggi : 65 cm
   Diameter : 1.7 cm
   Kapasitas Adonan : 4 kg
2. Penggeraknya berupa motor DC dengan catu daya 12 volt dan kecepatan 20 rpm.
3. Sensor yang digunakan untuk mendeteksi kecepatan adalah *Photo sensor* B5S-T2M.
4. Kontroler yang digunakan adalah kontroler PID.
5. Menggunakan satu buah mikrokontroler Arduino Uno Rev3.
6. *Software* yang digunakan sebagai pemrograman yaitu Arduino ERW 1.0.5.

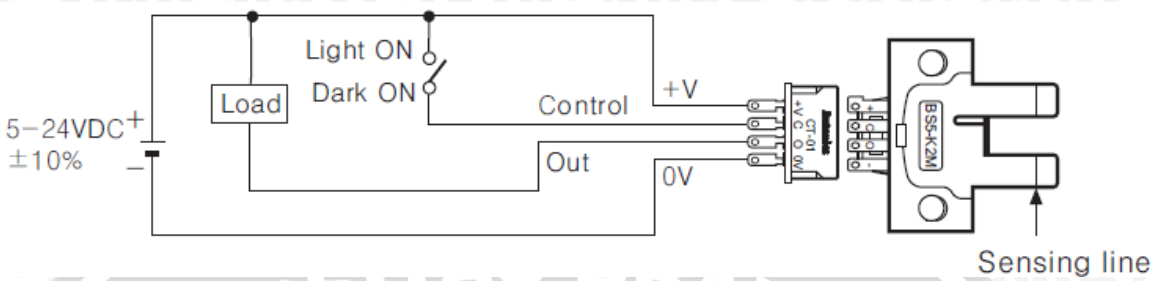
Perancangan perangkat keras alat pengaduk adonan dapat dilihat pada Gambar 4.2.



Gambar 4.2 Rancangan Alat Pengaduk Adonan Dodol

4.4 *Photo Sensor* BS5-T2M

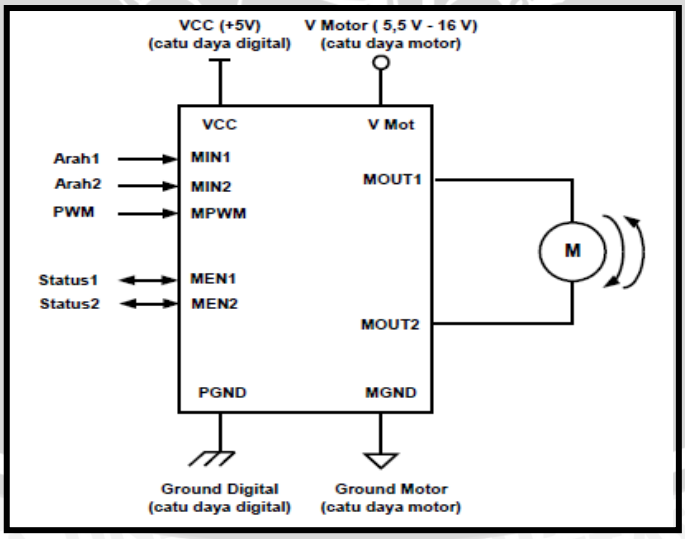
Photo sensor BS5-T2M digunakan sebagai pengukur kecepatan putaran motor DC. Penempatan sensor ini tepat sejajar dengan motor DC, jadi setiap perubahan kecepatan dari motor DC akan ikut mempengaruhi perubahan jumlah pulsa. Sistem perancangan dan hasil perancangan sensor ditunjukkan pada Gambar 4.3.



Gambar 4.3 Photo sensor BS5-T2M

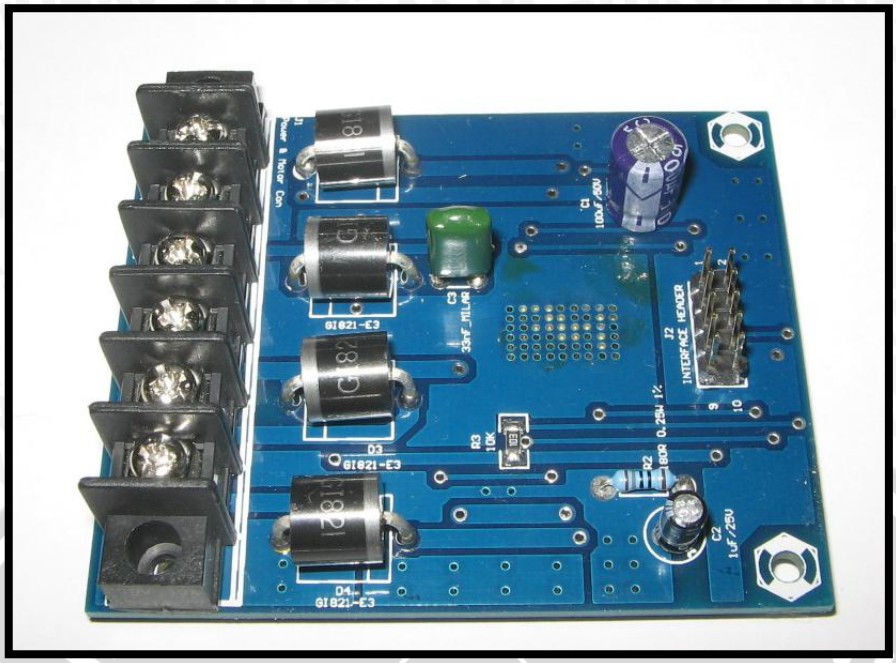
4.5 Perancangan *Driver* Motor DC

Modul pengendali motor DC yang digunakan adalah modul *EMS 5A H-Bridge*. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari Arduino Uno ERW 1.0.5. Gambar koneksi modul pengendali motor DC *EMS 5A H-Bridge* ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Koneksi Modul EMS 5A H-Bridge

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 5 ampere. Gambar modul rangkaian driver *EMS 5A H-Bridge* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Modul Rangkaian *Driver* EMS 5A H-Bridge

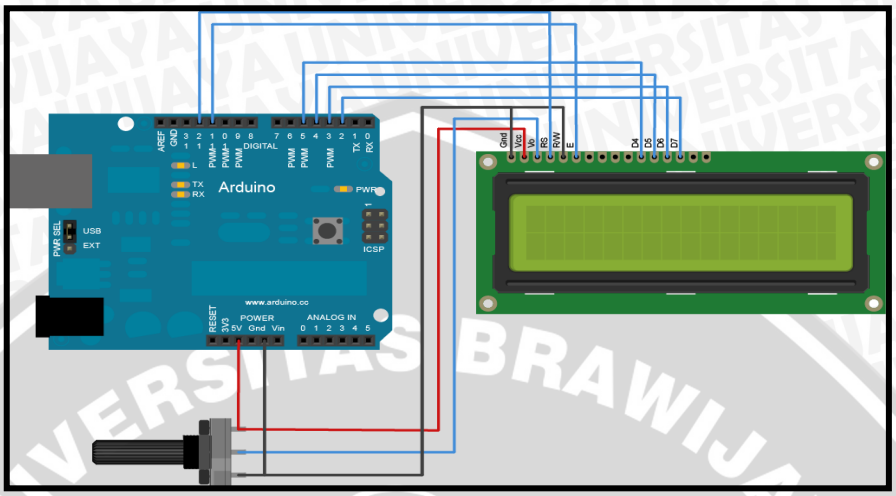
4.6 *Liquid Crystal Display* (LCD)

LCD yang digunakan dalam perancangan kali ini adalah Hitachi HD44780 dan di dalam sistem ini berfungsi sebagai penampil atau penunjuk waktu dan *error* yang ditunjukkan. Digunakannya LCD dalam perancangan ini diperlukan karena apabila hasil atau output waktu dan *error* ditunjukkan pada serial monitor yang ada pada Arduino Uno, untuk pengambilan data sebagai bahan analisis perancangan akan menjadi lebih sulit.

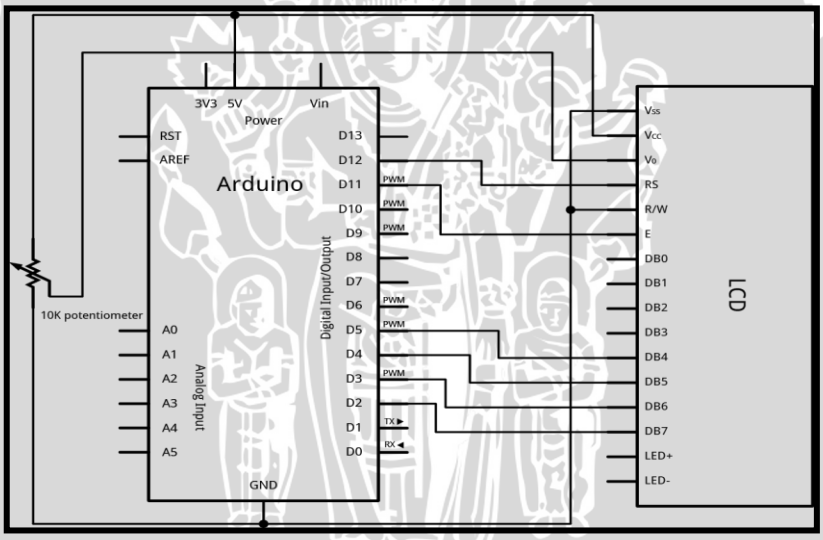
LCD Hitachi dapat dikontrol dalam dua mode : 4-bit atau 8-bit. Mode 4-bit membutuhkan tujuh I/O pin Arduino Uno, sedangkan mode 8-bit membutuhkan sebelas I/O pin Arduino Uno. Untuk rangkaian LCD, diperlukan beberapa hardware sebagai penunjang agar LCD dapat berfungsi dengan baik, yaitu :

- Arduino Uno
- LCD *screen* (yang kompatibel dengan Hitachi HD44780)
- Pin *header* untuk LCD *display*
- 10k potensiometer
- *Project board*
- Kabel *male to female*

Dengan pin *header* disolder terlebih dulu pada LCD, kemudian dengan menggunakan kabel male to female dihubungkan ke Arduino Uno seperti yang ditunjukkan pada Gambar 4.6.



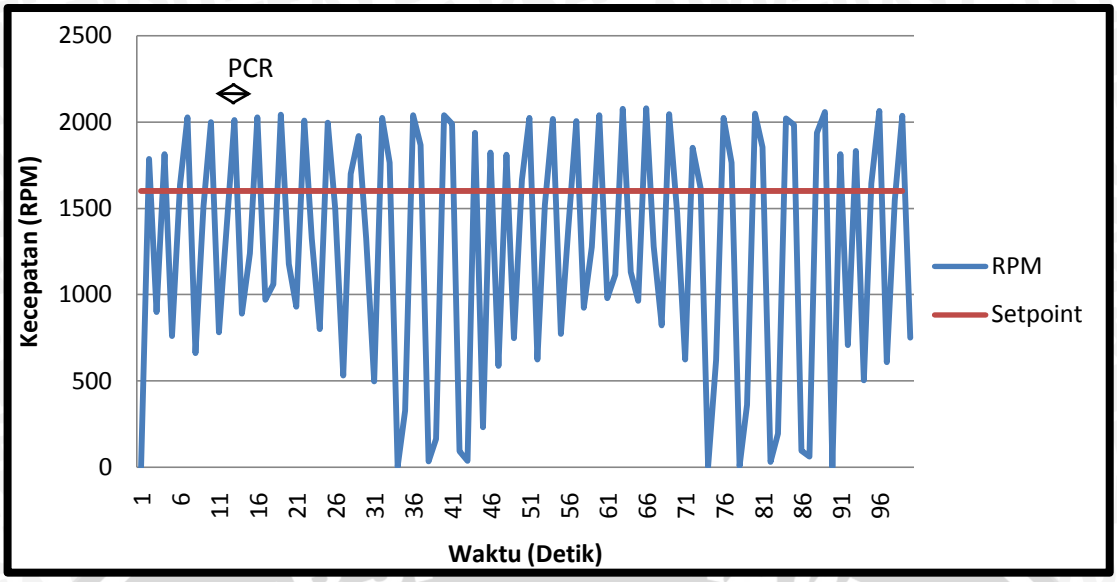
Gambar 4.6 LCD ke Arduino Uno



Gambar 4.7 Rangkain *Wiring* LCD ke Arduino Uno

4.7 Perancangan Kontroler PID

Perancangan kontroler dilakukan dengan menggunakan microsoft excel untuk mengamati respon silinder. Langkah pertama adalah dengan melihat respon motor setelah diberi kenaikan Kcr = 10 seperti dalam Gambar 4.8.



Gambar 4.8 Respon PID Zygler Nichols

Berdasarkan Gambar 4.8, didapatkan nilai Pcr sebesar 3. Nilai parameter kontroler PID ditentukan berdasarkan tabel aturan dasar Ziegler-Nichols dengan *critical gain* Kcr dan *critical period* Pcr yang ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Dasar Ziegler-Nichols Berdasarkan *Critical Gain* Kcr dan *Critical Period* Pcr

(Ogata K., 1997)

TipeKontroler	K_p	T_i	T_d
P	$0.5\ Kcr$	∞	0
PI	$0.45\ Kcr$	$\frac{1}{1.2} Pcr$	0
PID	$0.6\ Kcr$	$0.5\ Pcr$	$0.125\ Pcr$

maka nilai parameter PID yang diperoleh adalah :

Kp = 0.6 x 10 = 6

Ti = 0.5 x 3 = 1.5

Td = 0.125 x 3 = 0.475

Dengan demikian dapat ditentukan nilai Kp, Ki, dan Kd adalah:

Kp = 6

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

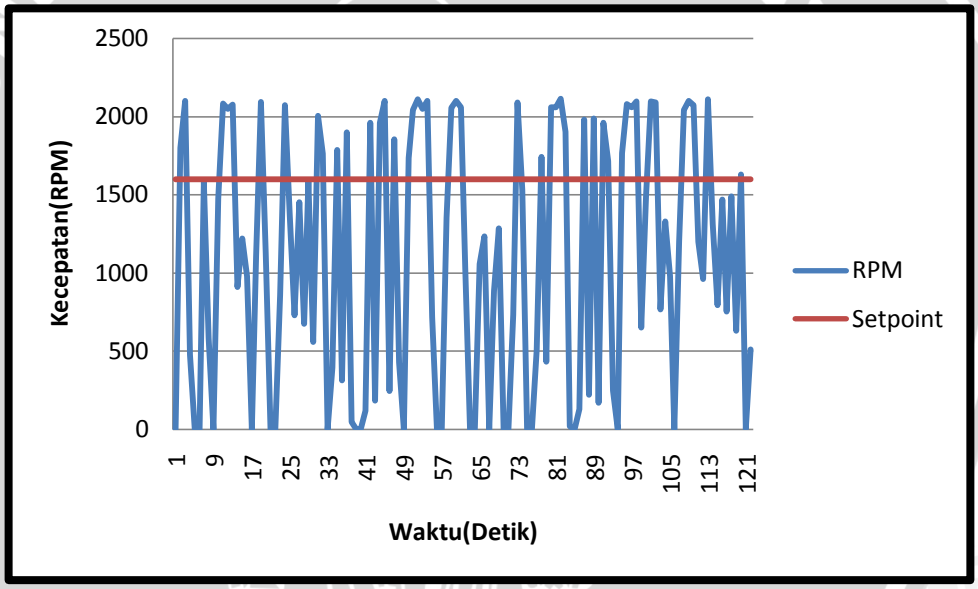
= 6/1.5

= 4

Kd = Kp x Td

= 6 x 1.25 = 2.85

Dari penghitungan penentuan nilai penguatan dari metode kedua Ziegler-Nichols di atas diperoleh Kp = 6, Ki = 4, Kd = 2.85 yang akan digunakan untuk pengendali kecepatan motor DC. Dengan nilai Kp, Ki, dan Kd yang sudah diperoleh, maka grafik sistem respon yang diperoleh ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Respon Sistem dengan Kp, Ki, dan Kd Berdasarkan Ziegler-Nichols

Dari grafik Gambar 4.9 bisa dilihat sistem tidak dapat menggunakan perhitungan Ziegler-Nichols karena respon yang dihasilkan terus mengalami osilasi dan tidak bisa *steady*. Maka dilakukan *hand tuning* (*hand* eksperimen) untuk mendapatkan respon yang lebih baik.

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). Pada

perancangan kontroler PID sistem pengendalian kecepatan ini, menggunakan metode *handtuning* untuk menentukan parameter Kp, Ki, dan Kd.

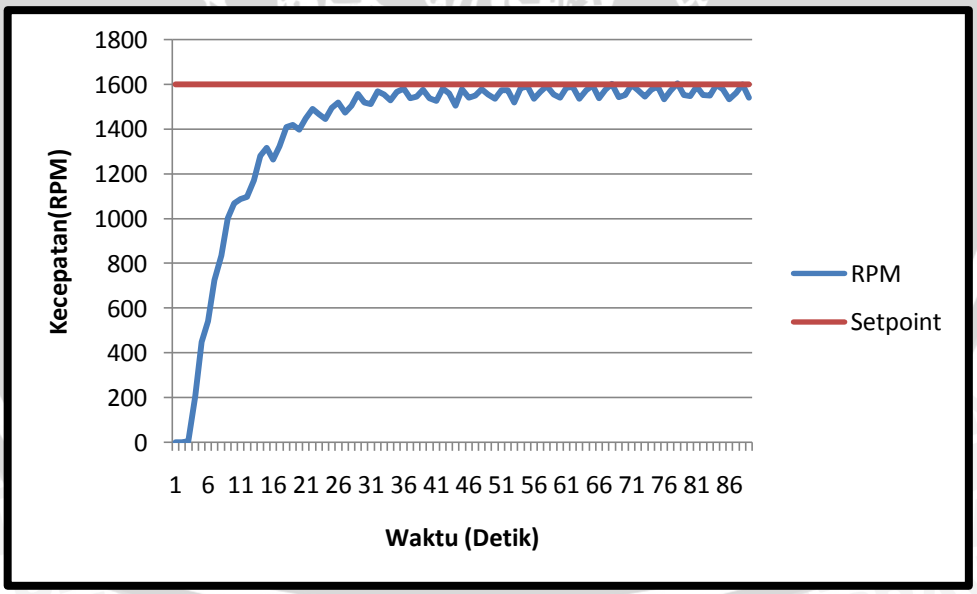
Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut :

1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai Ki = 0 dan Kd = 0.
2. Mengatur nilai Kp hingga didapatkan respon yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai Kp nilai dari Kd dinaikkan untuk mengurangi *overshoot* yang terjadi.
4. Naikkan nilai Ki untuk mengurangi *offset*.

Dengan menggunakan metode *hand tuning* nilai parameter PID perlu diubah-ubah secara *trial* dan *error* agar respon yang diperoleh sesuai dengan harapan.

Proses *tuning* parameter PID ini dilakukan dengan cara mengatur nilai Kp hingga didapatkan respon sistem yang mendekati *setpoint* 20 rpm (1600 pada pembacaan sensor).

Untuk nilai Kp = 0.1, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.10.



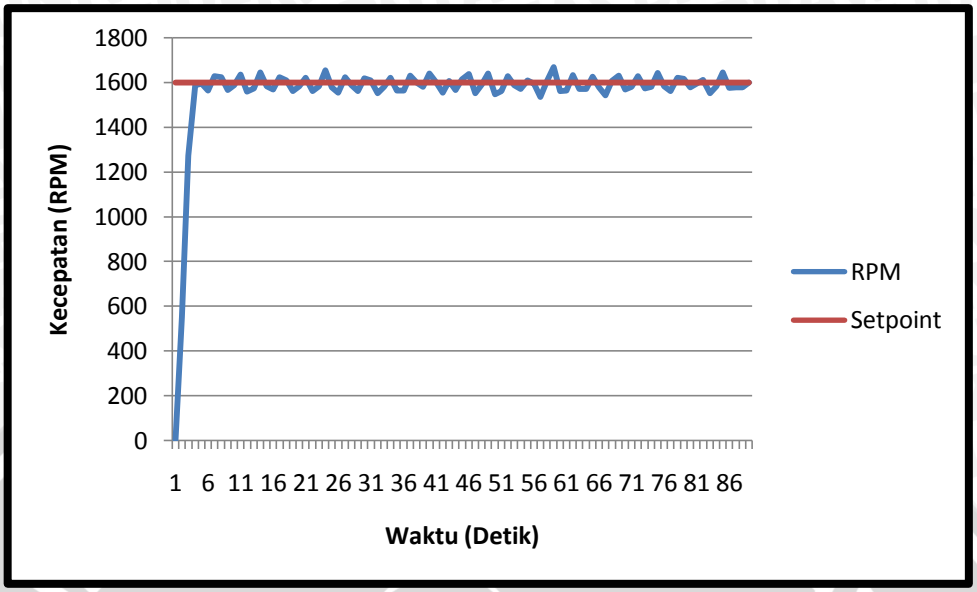
Gambar 4.10 Grafik Respon untuk Kp = 0.1

$$\%\ Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1400.31 - 1600|}{1600} \text{ x } 100\%$$

= 12.48% dan memiliki waktu *steady* 33 detik.

Untuk nilai Kp = 0.5, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.11.



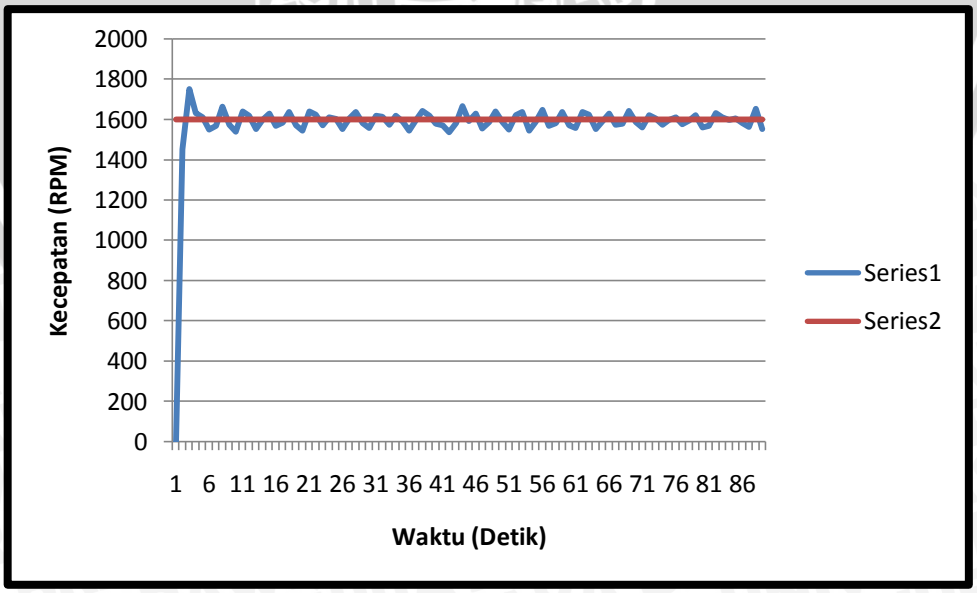
Gambar 4.11 Grafik Respon untuk Kp = 0.5

$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1559.21 - 1600|}{1600} \text{ x } 100\%$$

= 2.54% dan memiliki waktu *steady* 5 detik.

Untuk nilai Kp = 0.95, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada Gambar 4.12.



Gambar 4.12 Grafik Respon untuk Kp = 0.95

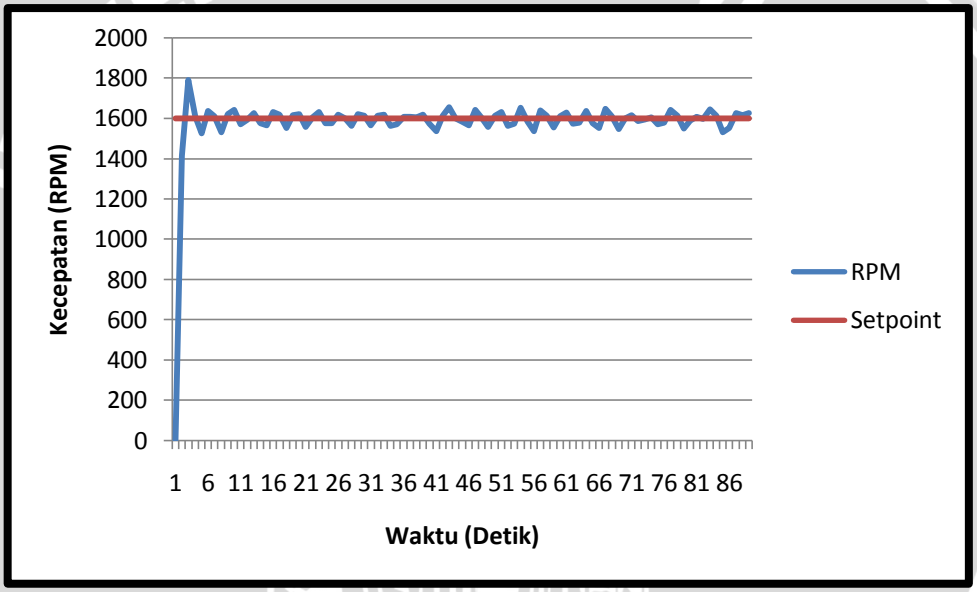
$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1577.85 - 1600|}{1600} \text{ x } 100\%$$

= 1.38% dan memiliki waktu *steady* 4 detik.

Dari hasil perancangan tersebut, didapatkan nilai Kp = 0.95 dapat mencapai stabil dan memiliki *error steady state* paling kecil walaupun masih terjadi *overshoot* dari titik stabil sekarang dan akan terjadi osilasi jika Kp dinaikkan. Setelah diperoleh nilai Kp yang cukup baik, Untuk mengurangi *overshoot* maka digunakanlah kontroler *derivative*.

Untuk nilai Kp = 0.9, Ki = 0, dan Kd = 10 diperoleh grafik respon seperti pada Gambar 4.13.



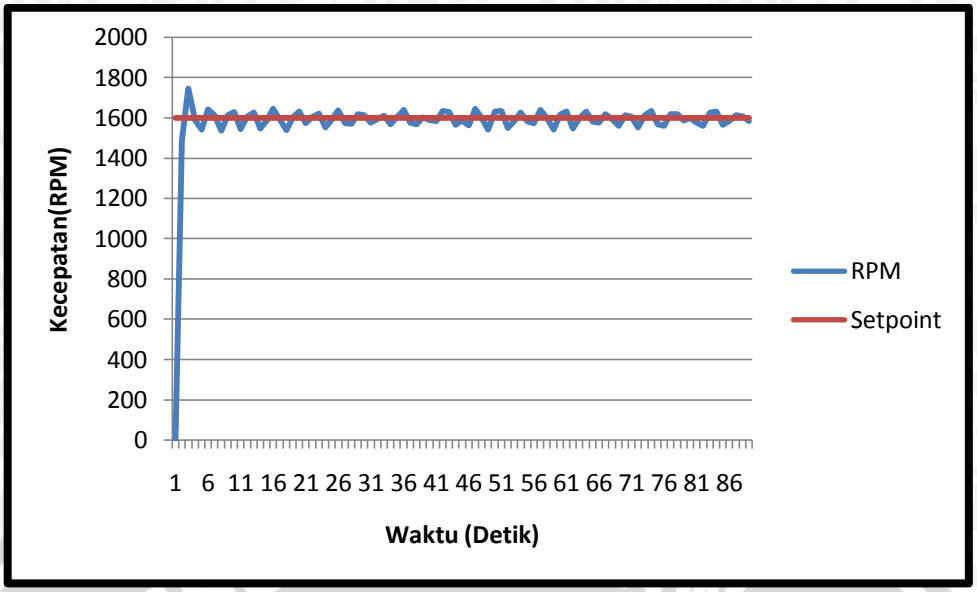
Gambar 4.13 Grafik Respon untuk Kp = 0.95 dan Kd = 10

$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1578.14 - 1600|}{1600} \text{ x } 100\%$$

= 1.36% dan memiliki waktu *steady* 4 detik.

Untuk nilai Kp = 0.9, Ki = 0, dan Kd = 50 diperoleh grafik respon seperti pada Gambar 4.14.



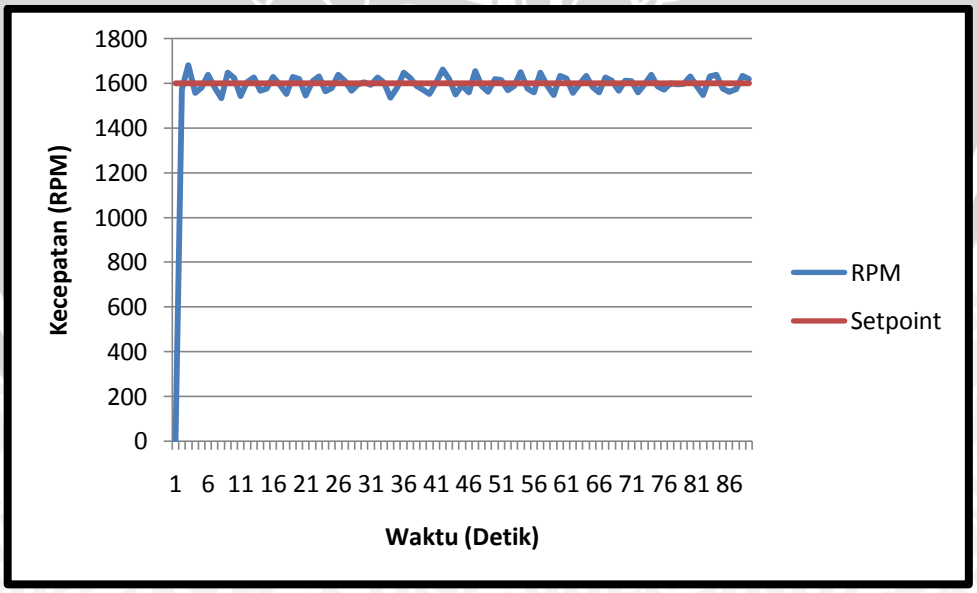
Gambar 4.14 Grafik Respon untuk Kp = 0.95 dan Kd = 50

$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1578.12 - 1600|}{1600} \text{ x } 100\%$$

= 1.37% dan memiliki waktu *steady* 4 detik.

Untuk nilai Kp = 0.9, Ki = 0, dan Kd = 100 diperoleh grafik respon seperti pada Gambar 4.15.



Gambar 4.15 Grafik Respon untuk Kp = 0.95 dan Kd = 100

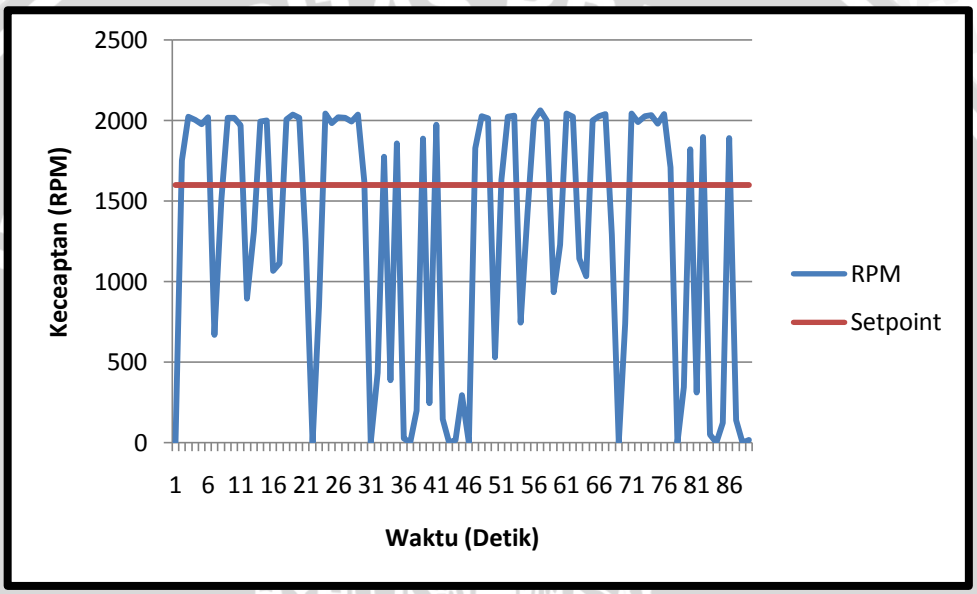
$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1578.79 - 1600|}{1600} \text{ x } 100\%$$

= 1.32% dan memiliki waktu *steady* 2 detik.

Grafik respon dalam Gambar 4.15 menunjukkan penggunaan Kd=100 menghilangkan *overshoot* tetapi masih terdapat *offset.* Untuk menghilangkan *offset* yang terjadi maka digunakanlah kontroler integral.

Untuk nilai Kp = 0.9, Ki = 0.1, dan Kd = 100 diperoleh grafik respon seperti pada Gambar 4.16.



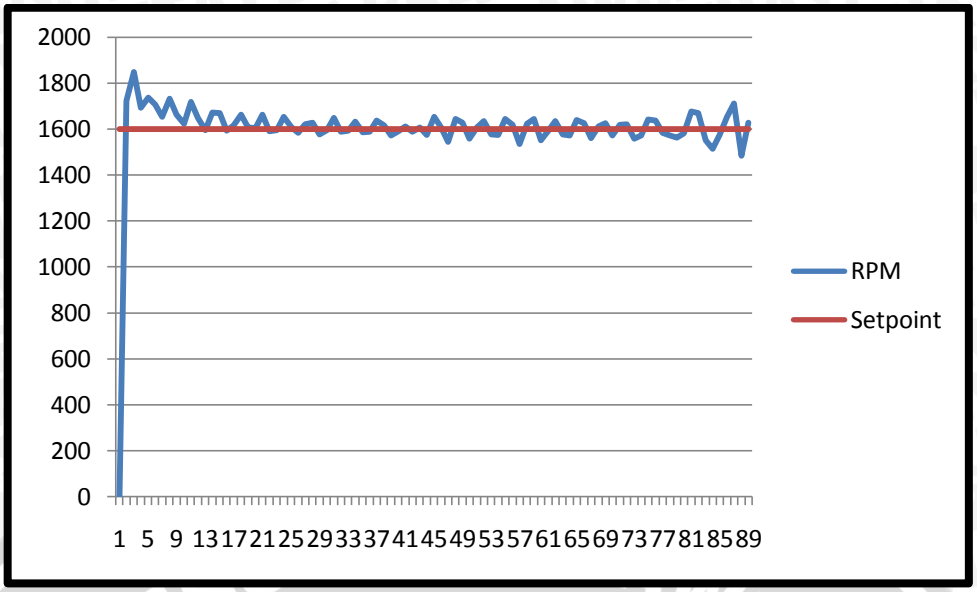
Gambar 4.16 Grafik Respon untuk Kp = 0.95, Ki = 0.1, dan Kd = 100

$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1287.97 - 1600|}{1600} \text{ x } 100\%$$

= 19.5%.

Untuk nilai Kp = 0.9, Ki = 0.0001, dan Kd = 100 diperoleh grafik respon seperti pada Gambar 4.17.



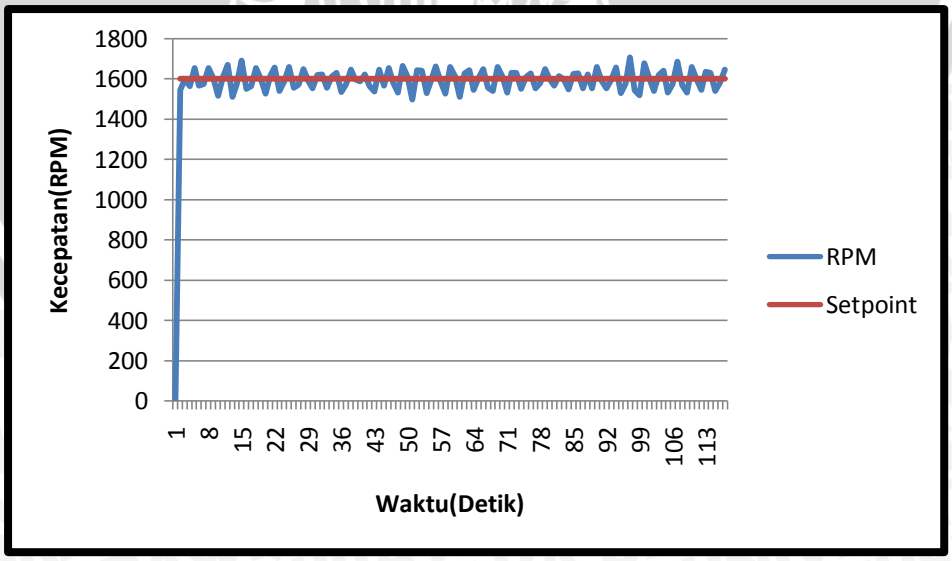
. Gambar 4.17 Grafik Respon untuk Kp = 0.95, Ki = 0.0001, dan Kd = 100

$$\% \, Ess = \frac{|Average\ speed\ steady - \text{setpoint}|}{\text{setpoint}} \text{ x } 100\ \%$$

$$= \frac{|1580.05 - 1600|}{1600} \text{ x } 100\%$$

= 1.24% dan memiliki waktu *steady* 10 detik.

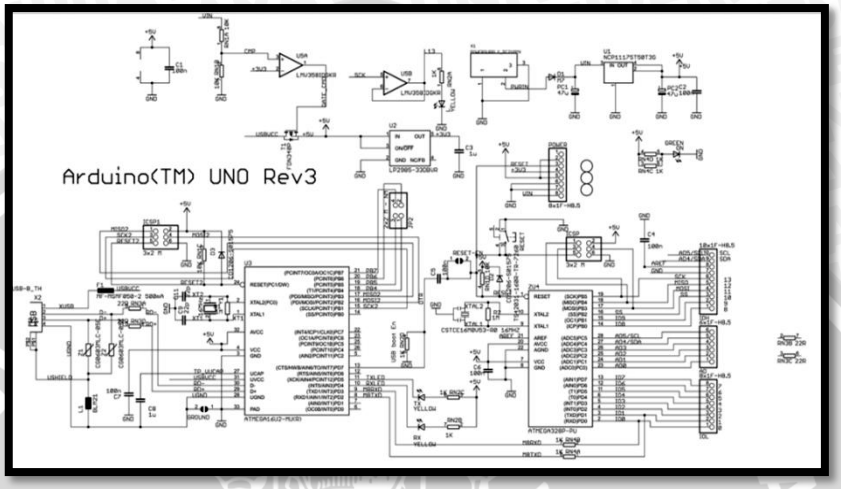
Hasil *tuning* parameter kontroler telah diperoleh nilai Kp = 0.95, Ki = 0.00000001, dan Kd=100 didapatkan repon yang baik karena tidak ada *overshoot*, waktu *steady* 2 detik dan hanya memiliki *error steady state* 0.3 %. Dengan nilai Kp, Ki, Kd yang sudah diperoleh, maka grafik respon akan ditunjukkan pada Gambar 4.18.



Gambar 4.18 Respon Sistem dengan Kp, Ki, dan Kd Berdasarkan *Hand-Tuning*

4.8 Modul Arduino Uno Rev.3

Pada sistem pengendalian kecepatan putaran alat pengaduk adonan ini digunakan Arduino Uno Rev.3 sebagai pengolah data dalam proses pengaturan kecepatan terdapat pada Driver motor *H-Bridge* untuk menggerakan motor DC. Konfigurasi kaki I/O dari Arduino Uno Rev3 ditunjukkan dalam Gambar 4.19.



Gambar 4.19 Desain Sistem Arduino Uno Rev3

Sumber : Datasheet Arduino Rev3

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. 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*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke adaptor DC atau baterai untuk menjalankannya. Modul arduino uno ditunjukkan pada Gambar 4.20.

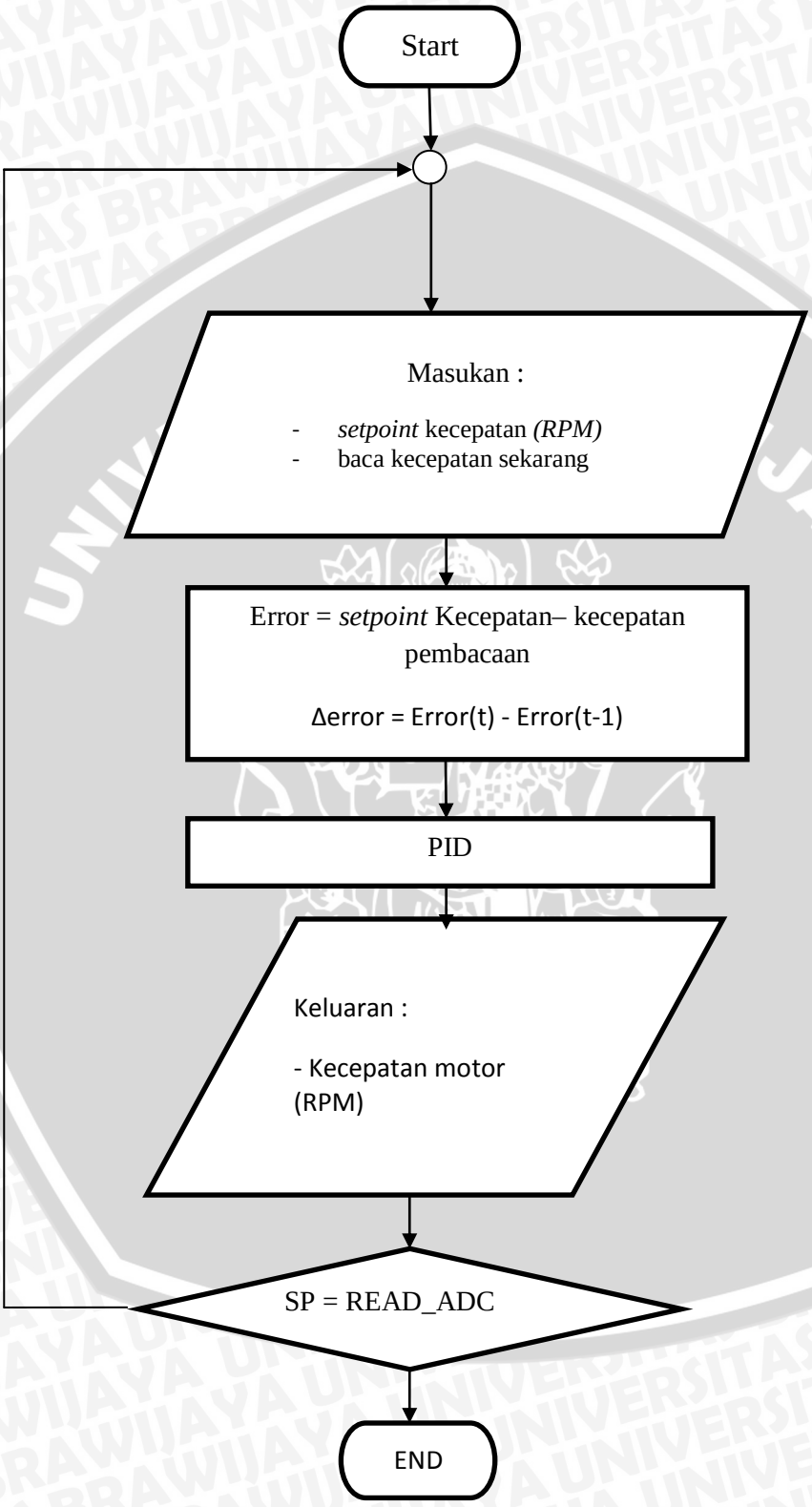
Gambar 4.20 Modul Arduino Uno

No	Pin	Fungsi
`1	2	Jalur masukan data dari sensor
2	3	LCD pin D6
3	4	LCD pin D5
4	5	LCD pin D4
5	6	Jalur masukan PWM
6	7	Pin input untuk MOUT1
7	8	Pin input untuk MOUT2
8	9	Pin enable untuk MOUT1
9	10	LCD pin D7
10	11	LCD pin Enable
11	12	LCD pin RS
12	5V	Jalur keluaran 5V
13	GND	Jalur keluaran ground

Tabel 4.2 Fungsi Pin Arduino Uno Rev3

4.9 Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.21 berikut:



Gambar 4.21 Flowchart Program