

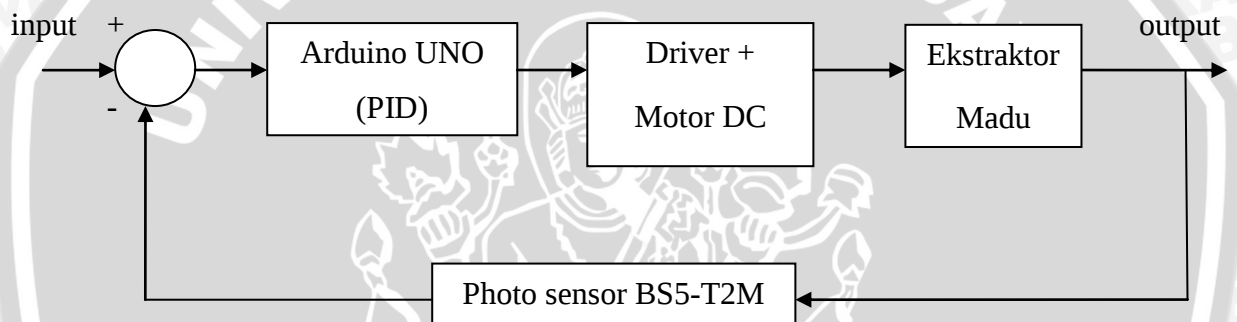
BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan ekstraktor madu 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

Sumber: Perancangan

Keterangan dari blok diagram diatas 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 perintah kepada *driver* motor untuk menggerakkan motor.
- Motor yang digunakan adalah motor DC yang berfungsi sebagai aktuator.
- *Driver* motor menggunakan modul EMS 30A 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 *Pulse Width Modulation* (PWM) ke *driver* motor agar motor berputar sesuai dengan setpoint yang diinginkan.

4.3 Perancangan Perangkat Keras (Hardware)

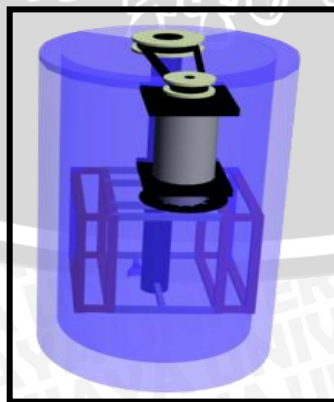
4.3.1 Spesifikasi Alat

Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Alat ekstraktor berbentuk silinder berbahan stainless steel dengan ukuran:

Tinggi	: 70 cm
Diameter	: 40 cm
Kapasitas	: 2 sarang madu
2. Penggeraknya berupa motor DC dengan daya 12 volt dan kecepatan 1800 rpm.
3. Sensor yang digunakan untuk mendeteksi kecepatan adalah *Photo sensor* BS5-T2M.
4. Kontroler yang digunakan adalah kontroler PID.
5. Menggunakan satu buah mikrokontroler Arduino Uno Rev3.
6. Software yang digunakan sebagai pemrograman yaitu Arduino IDE 1.0.5.

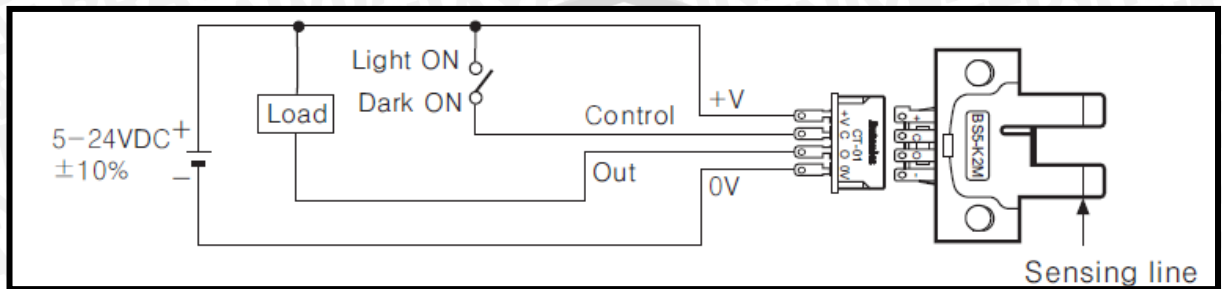
Perancangan perangkat keras ekstraktor madu dapat dilihat pada Gambar 4.2



Gambar 4.2 Rancangan ekstraktor madu

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

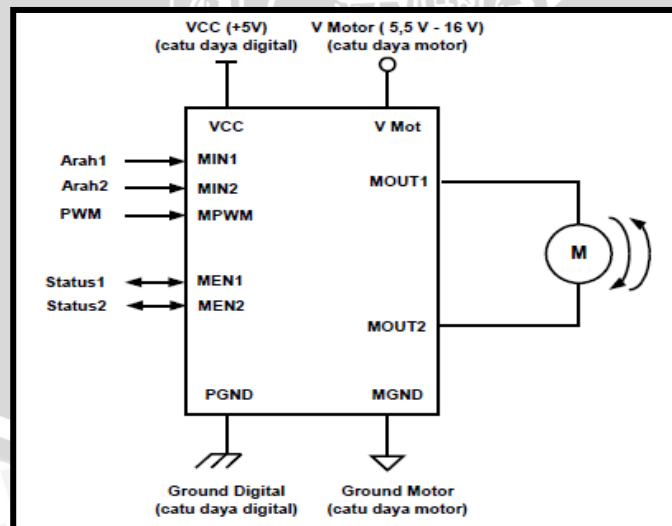


Gambar 4.3 Photo sensor BS5-T2M

Sumber: Photo sensor BS5-T2M datasheet

4.5 Perancangan Driver Motor DC

Modul pengendali motor DC yang digunakan adalah modul EMS 30A 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 30A H-Bridge ditunjukkan dalam Gambar 4.4



Gambar 4.4 Koneksi Modul EMS 30A H-Bridge

Sumber: EMS 30A H-Bridge datasheet

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 30 ampere. Gambar modul rangkaian *driver* EMS 30A H-Bridge dapat dilihat pada Gambar 4.5



Gambar 4.5 Modul Rangkaian *Driver* EMS 30A H-Bridge

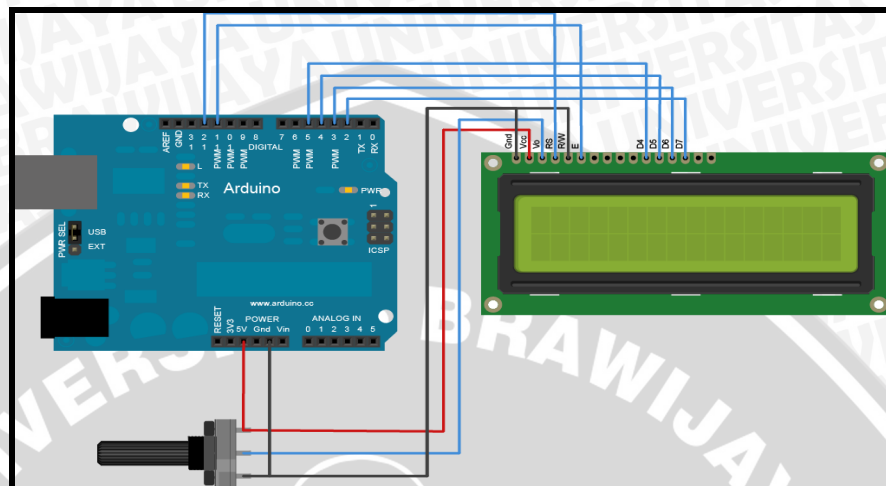
4.6 Liquid Crystal Display (LCD)

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 :

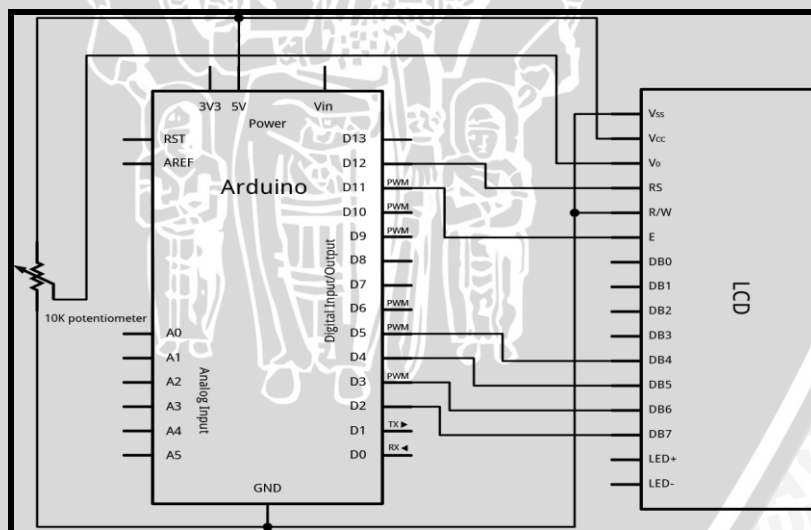
- Arduino Uno
- LCD screen (yang kompatibel dengan Hitachi HD44780)
- Pin header untuk LCD *display*
- 10k potensiometer
- *Project board*
- Kabel *male to female*

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 di bawah ini.



Gambar 4.6 LCD ke Arduino Uno

Sumber: www.arduino.cc/en/Tutorial/LiquidCrystal

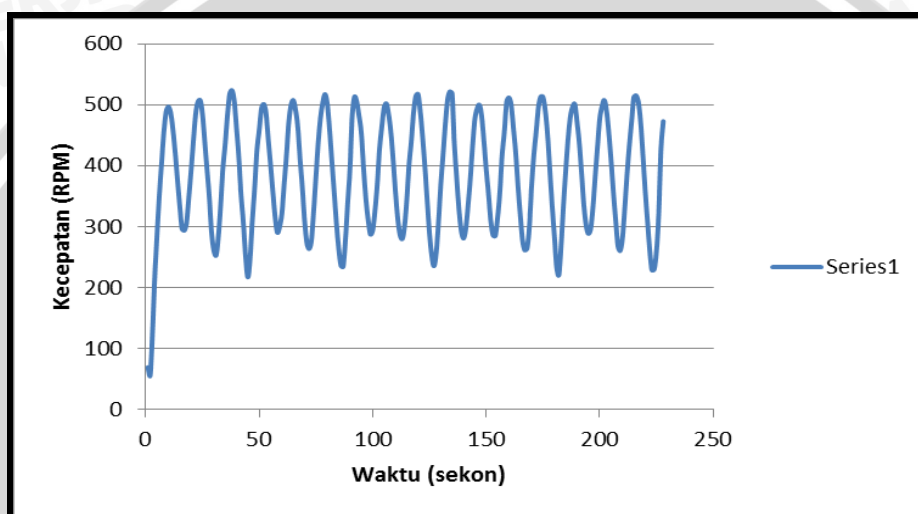


Gambar 4.7 Rangkaian Wiring LCD ke Arduino Uno

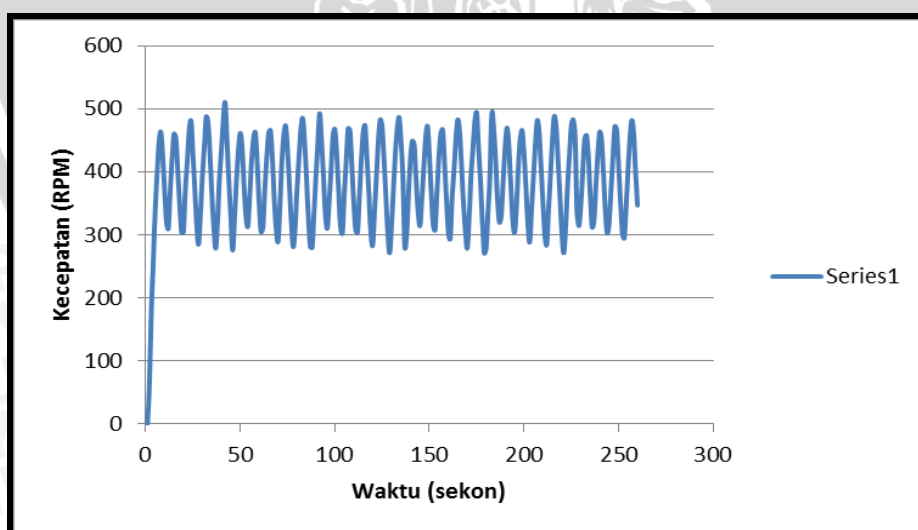
Sumber: www.arduino.cc/en/Tutorial/LiquidCrystal

4.7 Perancangan Kontroler PID

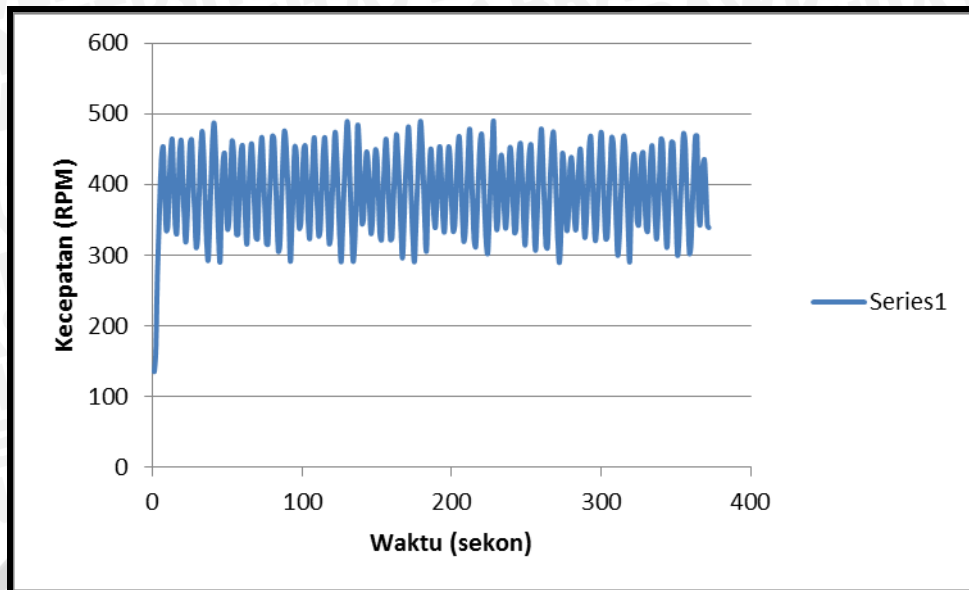
Kontroler PID dapat di *tuning* dalam beberapa cara, antara lain Ziegler-Nichols *tuning*, *loop tuning*, metode analitis, optimisasi, *pole placement*, *auto tuning*, dan *hand tuning* (Smith, 1979; Astrom & Hagglund, 1995). Dalam penelitian ini *tuning* parameter PID menggunakan zyglar-nichols tipe 2 yang merupakan sistem *closed loop*. *Tuning* dilakukan dengan cara memperbesar nilai Kcr sampai diperoleh respon osilasi yang divergen. Keluaran sistem setelah diberi kenaikan Kcr ditunjukkan pada Gambar 4.8 sampai dengan 4.17.



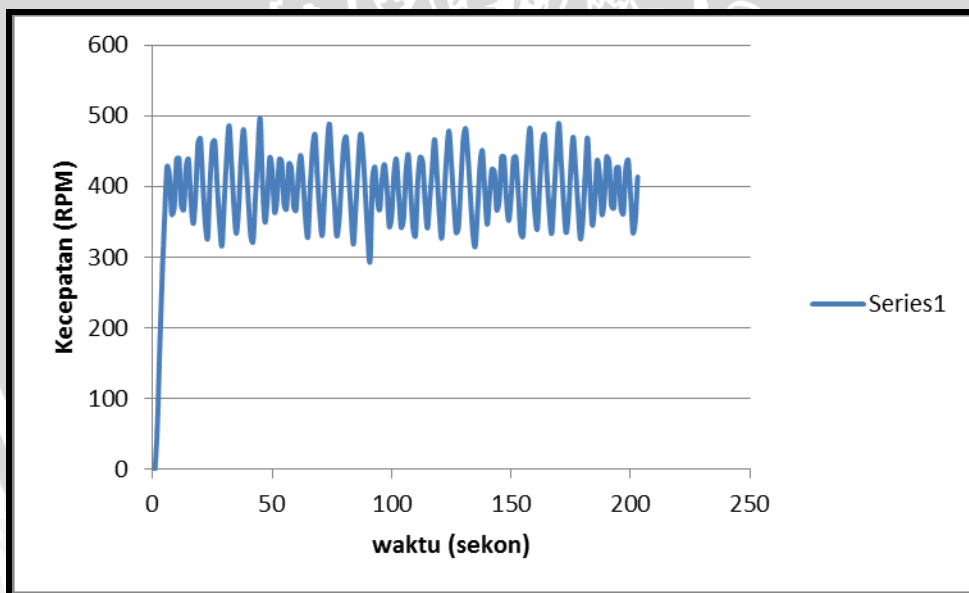
Gambar 4.8 Respon PID Zyglar Nichols dengan Kcr=1



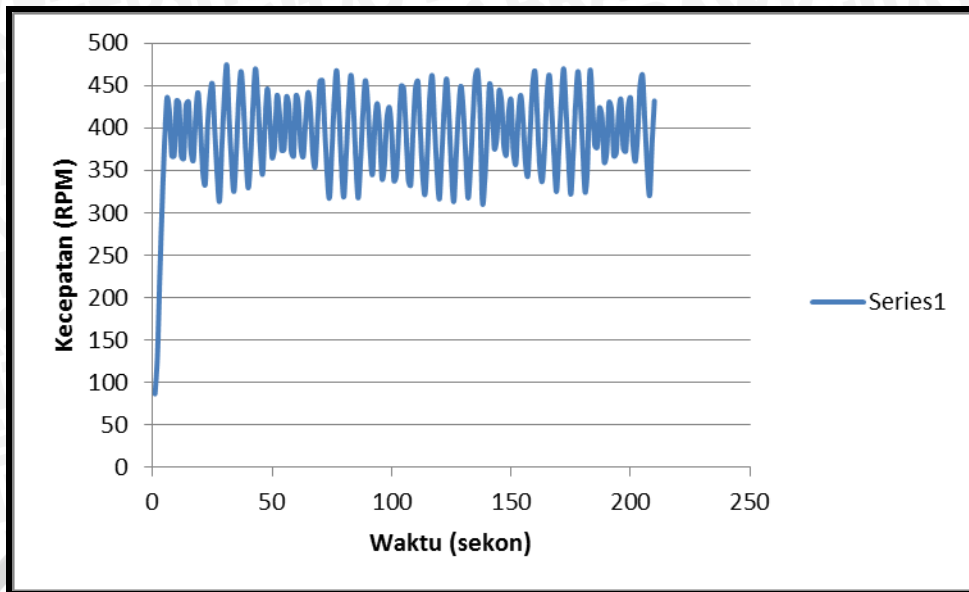
Gambar 4.9 Respon PID Zyglar Nichols dengan Kcr=5



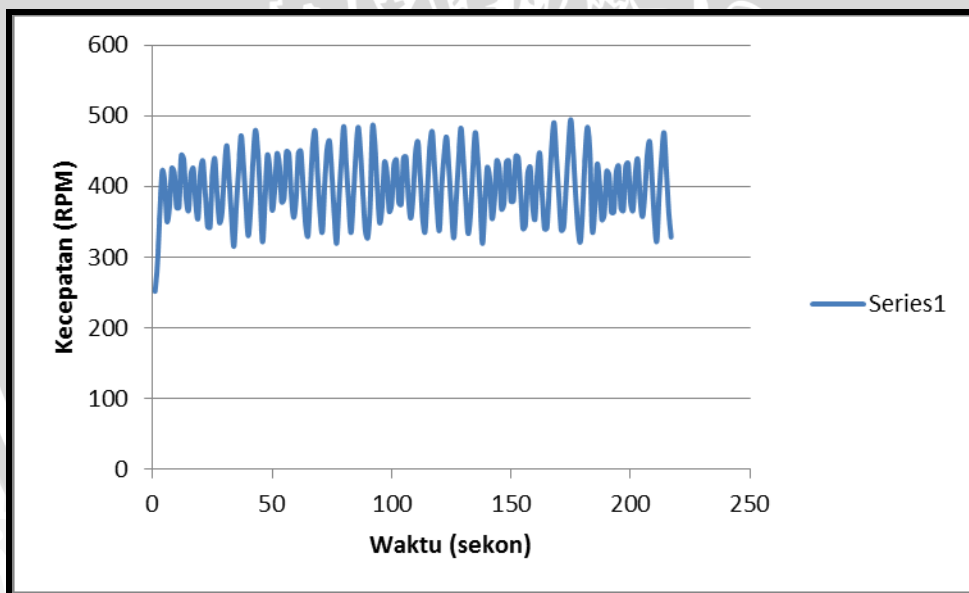
Gambar 4.10 Respon PID Zygler Nichols dengan Kcr=10



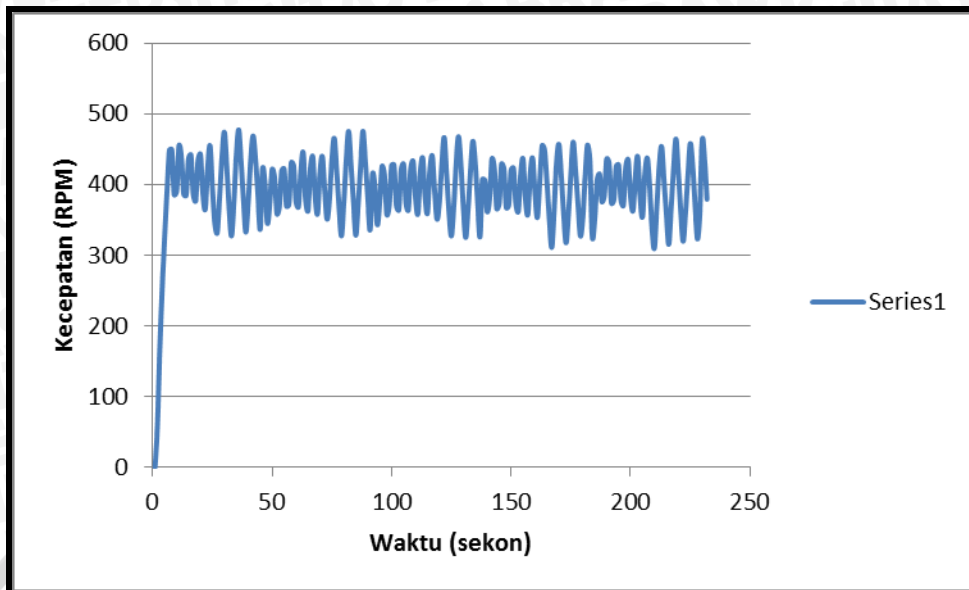
Gambar 4.11 Respon PID Zygler Nichols dengan Kcr=20



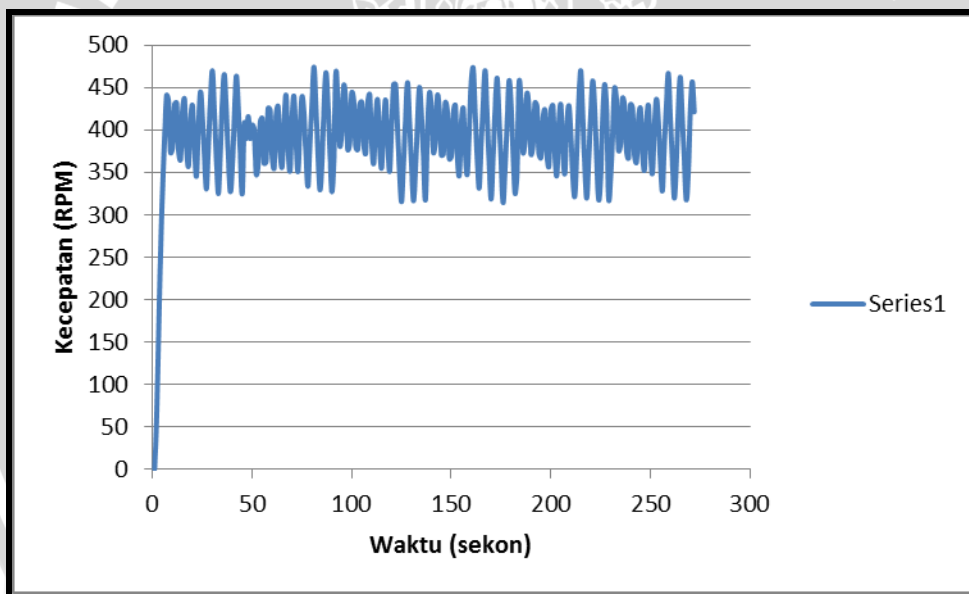
Gambar 4.12 Respon PID Zygler Nichols dengan Kcr=25



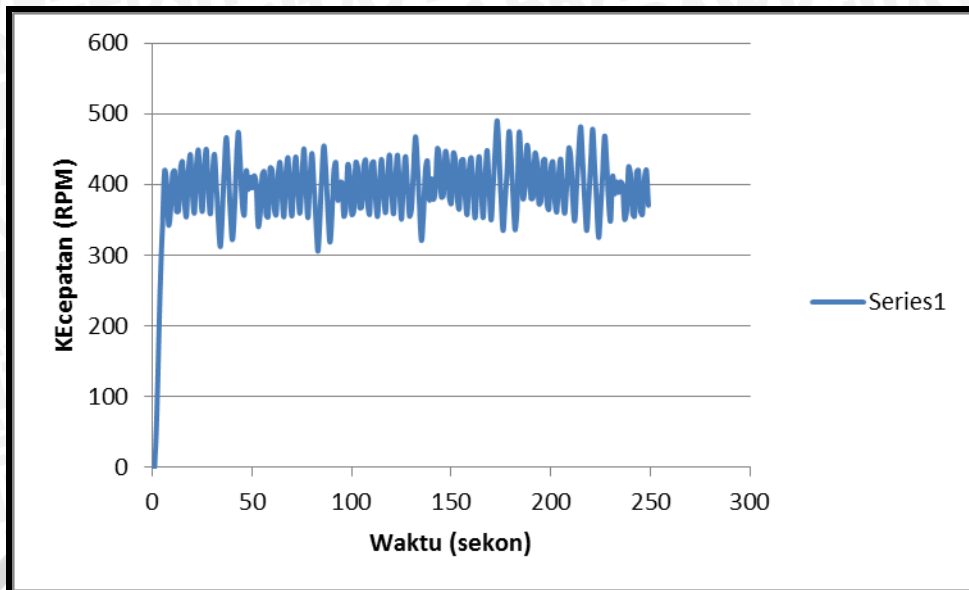
Gambar 4.13 Respon PID Zygler Nichols dengan Kcr=50



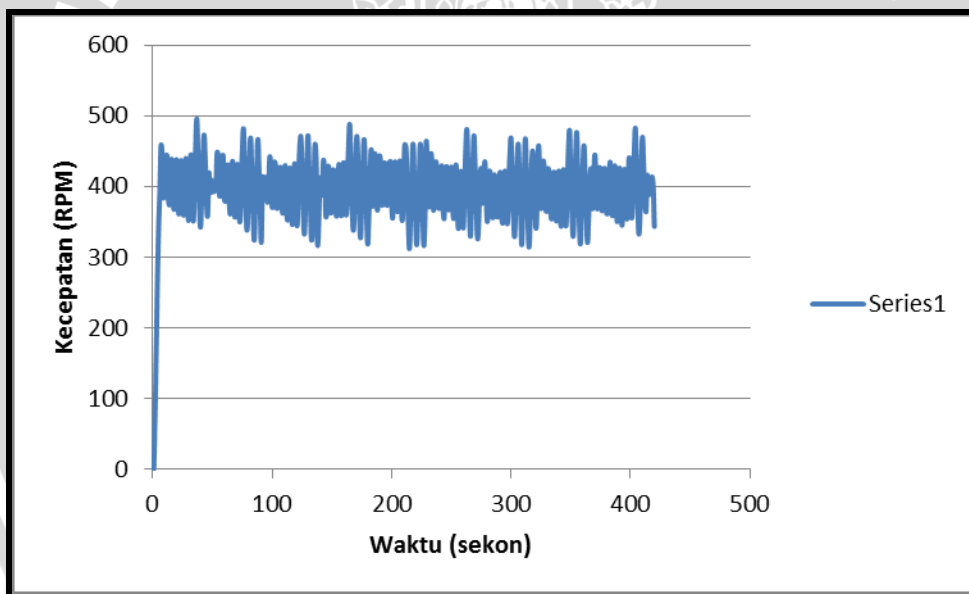
Gambar 4.14 Respon PID Zyglar Nichols dengan Kcr=75



Gambar 4.15 Respon PID Zyglar Nichols dengan Kcr=100



Gambar 4.16 Respon PID Zygler Nichols dengan Kcr=125



Gambar 4.17 Respon PID Zygler Nichols dengan Kcr=150

Dari gambar 4.8 sampai 4.17 di atas, dengan memperbesar nilai Kcr dari 1 sampai dengan 150 tidak diperoleh respon osilasi yang divergen sebagaimana yang di kriteriakan oleh zyglar -nichols tipe 2. Hal ini menunjukkan bahwa *pole* dan *zero* sistem berada di sisi kiri sumbu imajiner yang menyebabkan sistem stabil. Berdasarkan hal tersebut 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 (K_p), konstanta integral (K_i) dan konstanta diferensial (K_d). Pada perancangan kontroler PID sistem pengendalian kecepatan ini, menggunakan metode *handtuning* untuk menentukan parameter K_p , K_i , dan K_d .

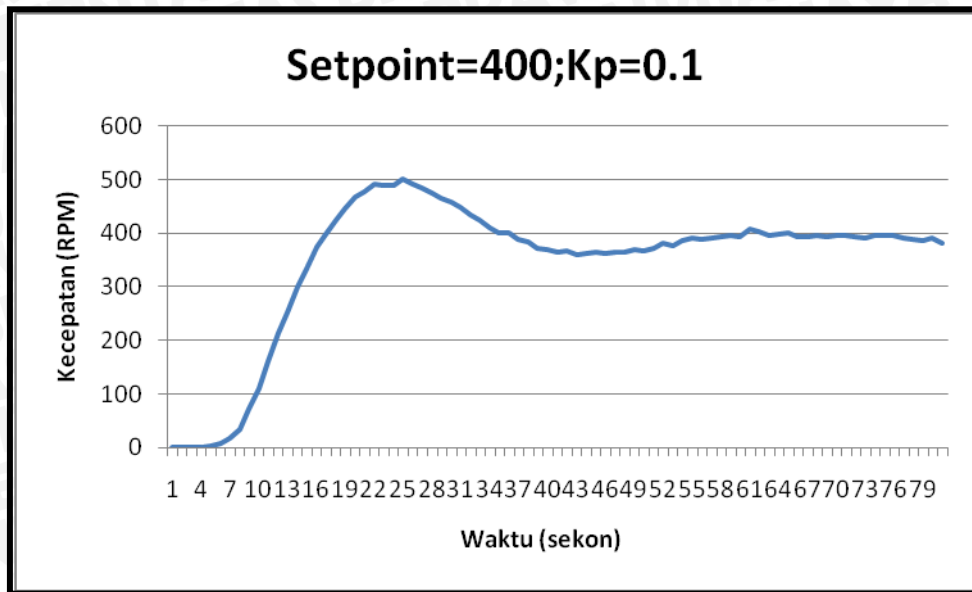
Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut :

1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai $K_i=0$ dan $K_d=0$.
2. Mengatur nilai K_p hingga didapatkan respon yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai K_p nilai dari K_d dinaikkan untuk mengurangi *overshoot* yang terjadi.
4. Naikkan nilai K_i 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 K_p hingga didapatkan respon sistem yang mendekati *setpoint* 400 rpm.

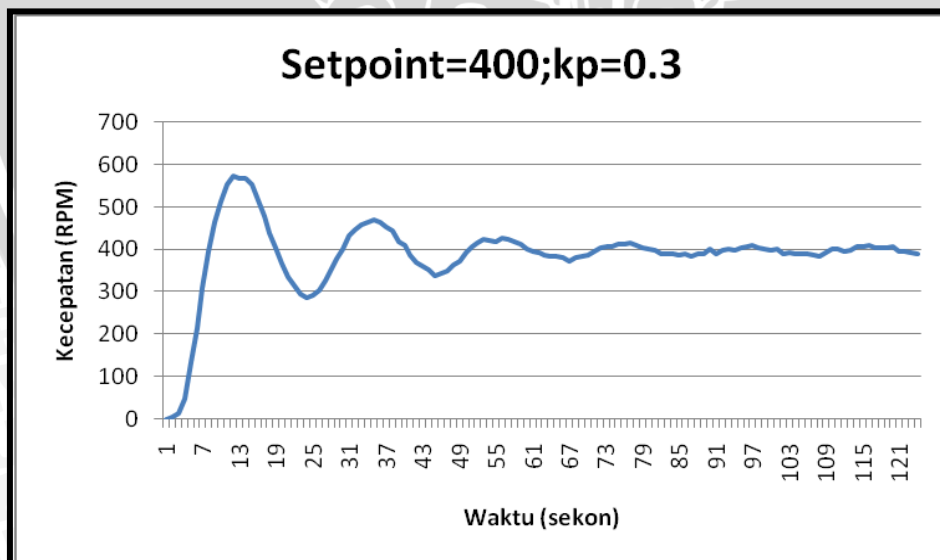
Untuk nilai $K_p = 0.1$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada gambar 4.18.



Gambar 4.18 Grafik respon dengan kp=0.1

$$\begin{aligned} \% Mp &= \frac{[RPM_{puncak} - RPM_{steady}]}{RPM_{steady}} \times 100 \% \\ &= \frac{[500.01 - 393.04]}{393.04} \times 100 \% \\ &= 27.12\% \end{aligned}$$

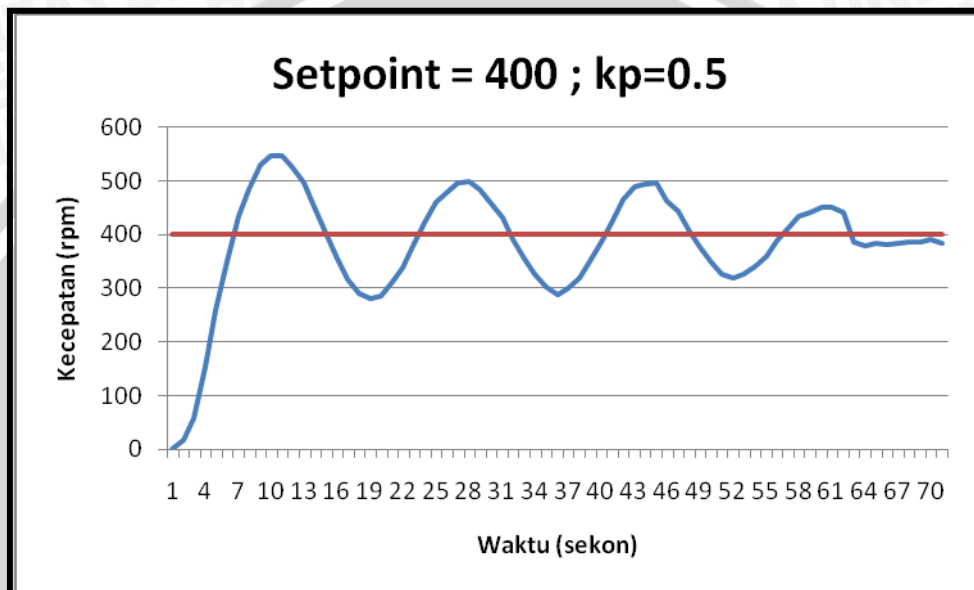
Untuk nilai Kp = 0.3, Ki = 0, dan Kd = 0 diperoleh grafik respon seperti pada gambar 4.19.



Gambar 4.19 Grafik respon dengan kp=0.3

$$\begin{aligned}\% Mp &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|568.08 - 396.28|}{396.28} \times 100 \% \\ &= 43.33\%\end{aligned}$$

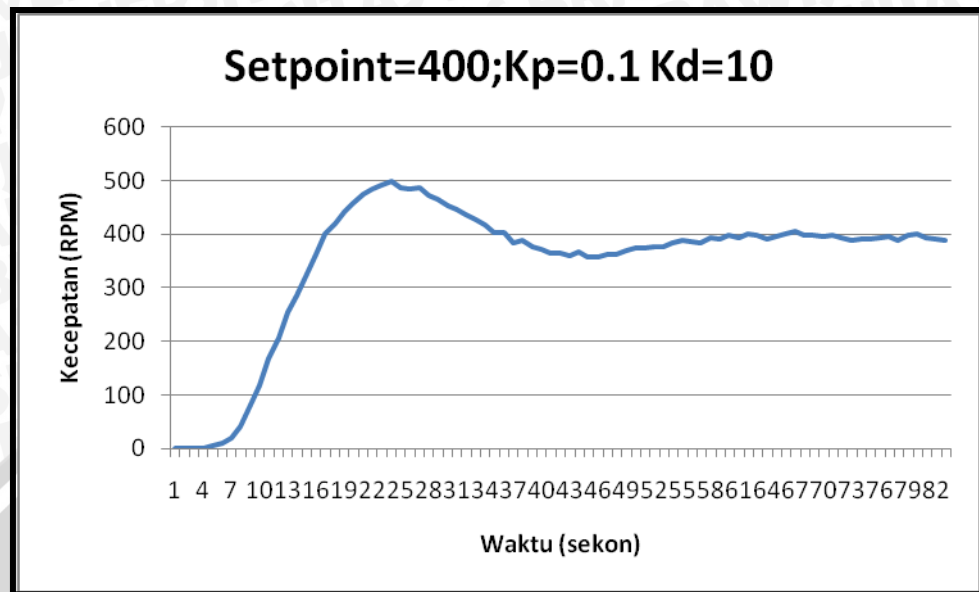
Untuk nilai $K_p = 0.5$, $K_i = 0$, dan $K_d = 0$ diperoleh grafik respon seperti pada gambar 4.20.



Gambar 4.20 Grafik respon dengan $k_p=0.5$

$$\begin{aligned}\% Mp &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|547.56 - 384.88|}{384.88} \times 100 \% \\ &= 42.26\%\end{aligned}$$

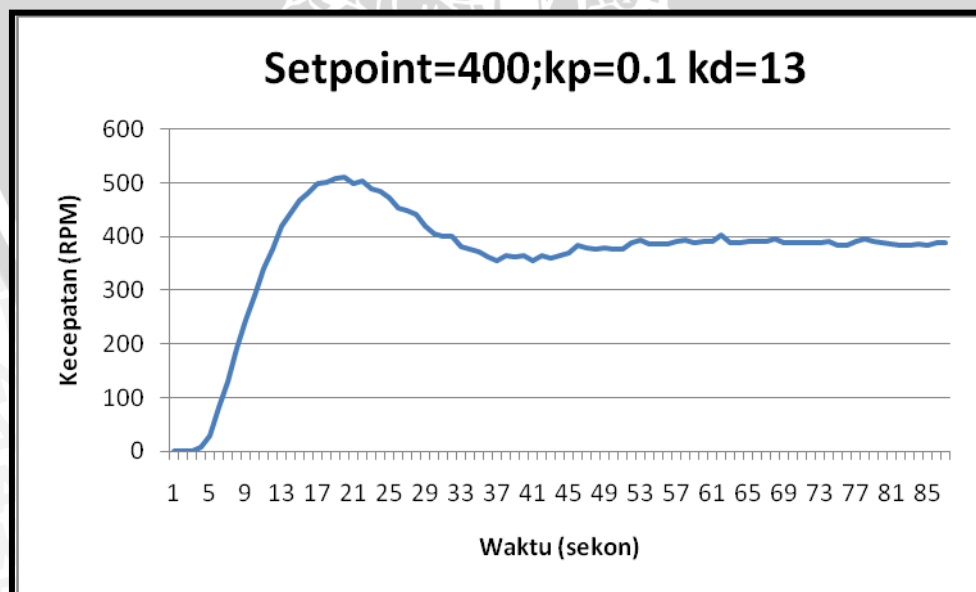
Dari hasil perancangan tersebut, didapatkan nilai $K_p = 0.1$ dapat mencapai stabil walaupun masih terjadi *overshoot* dari titik stabil sekarang dan akan terjadi osilasi jika K_p dinaikkan. Setelah diperoleh nilai K_p yang cukup baik, Untuk mengurangi *overshoot* maka digunakanlah kontroler *derivative*. Grafik respon kecepatan putaran dengan kontroler *derivative* ditunjukkan dalam gambar 4.21.



Gambar 4.21 Grafik respon dengan kp=0.1,kd=10

Untuk nilai Kp = 1, Ki = 0 dan Kd = 10 terdapat overshoot yaitu:

$$\begin{aligned}
 \% Mp &= \frac{RPM_{puncak} - RPM_{steady}}{RPM_{steady}} \times 100 \% \\
 &= \frac{497.88 - 387.07}{387.07} \times 100 \% \\
 &= 28.62\%
 \end{aligned}$$

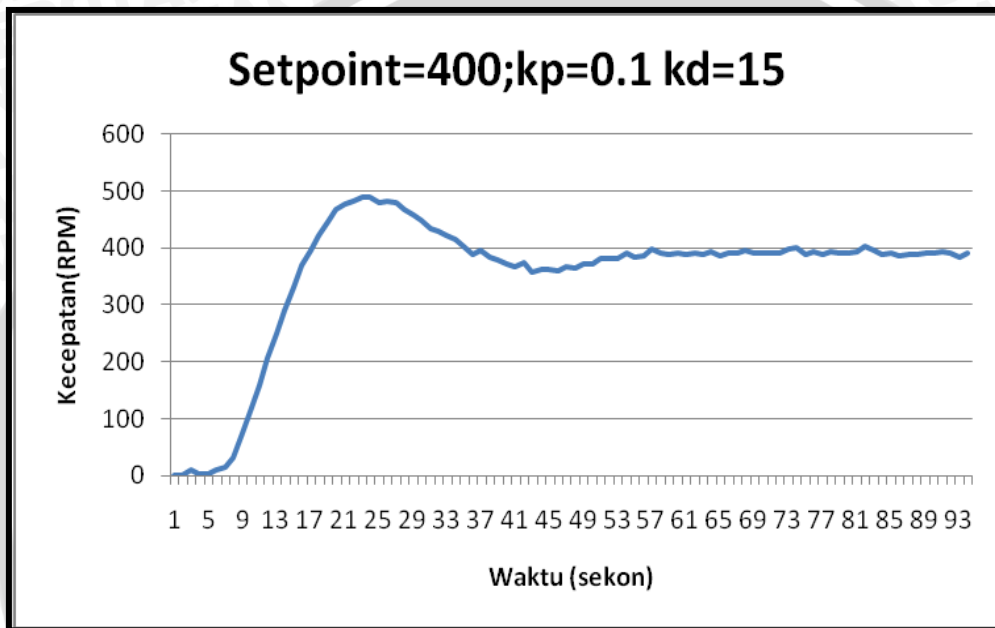


Gambar 4.22 Grafik respon dengan kp=0.1,kd=13

Untuk nilai $K_p = 0.1$, dan $K_d = 13$ terdapat *overshoot* yaitu:

$$\begin{aligned}\% M_p &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|509.76 - 388.55|}{388.55} \times 100 \% \\ &= 31.11\%\end{aligned}$$

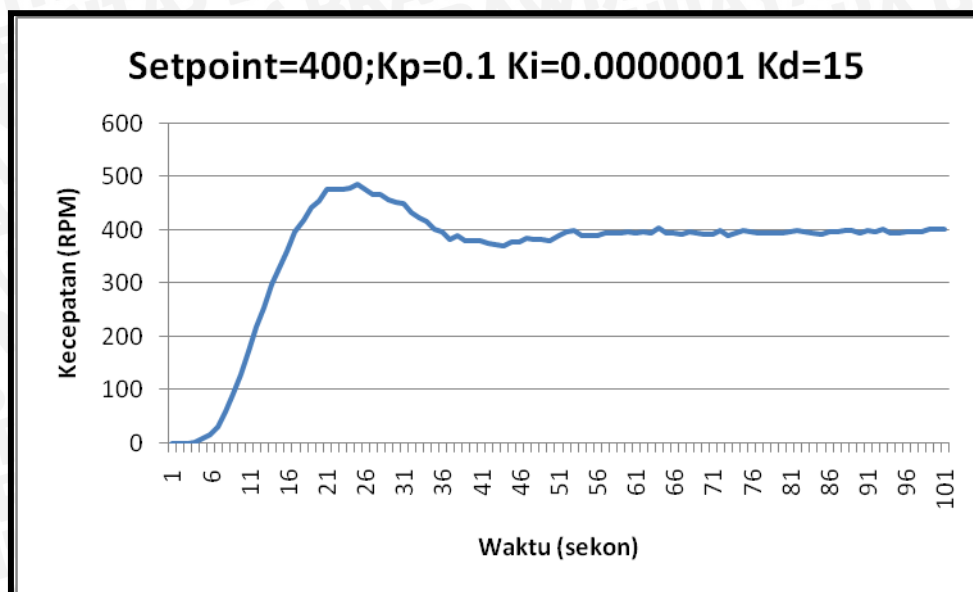
Hasil respon dari nilai $K_p=0.1$, $K_i=0.0000005$ dan $K_d = 0$ dapat dilihat pada gambar 4.23.



Gambar 4.23 Grafik respon dengan $k_p=0.1$ $K_d=15$

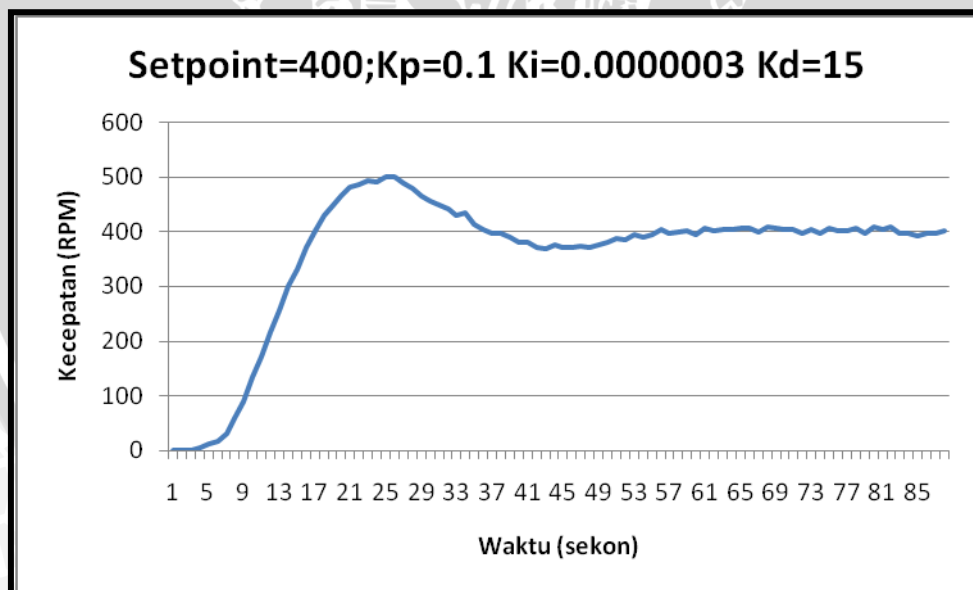
$$\begin{aligned}\% M_p &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|489.24 - 391.44|}{391.44} \times 100 \% \\ &= 24.98\%\end{aligned}$$

Grafik respon dalam gambar menunjukkan penggunaan $K_d=15$ mengurangi *overshoot* tetapi masih terdapat *offset*. Untuk mengurangi *offset* yang terjadi maka digunakanlah kontroler integral.



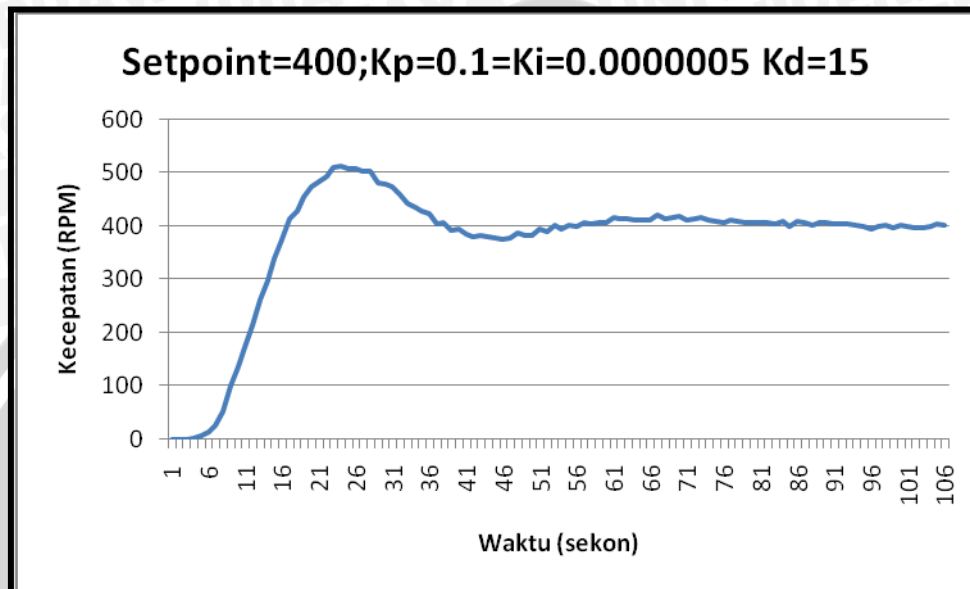
Gambar 4.24 Grafik respon dengan kp=0.1 ki=0.0000001 kd=15

$$\begin{aligned} \% Mp &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|484.92 - 394.98|}{394.98} \times 100 \% \\ &= 22.77\% \end{aligned}$$



Gambar 4.25 Grafik respon dengan kp=0.1 ki=0.0000003 kd=15

$$\begin{aligned}\% Mp &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|501.12 - 401.14|}{401.14} \times 100 \% \\ &= 24.92\%\end{aligned}$$



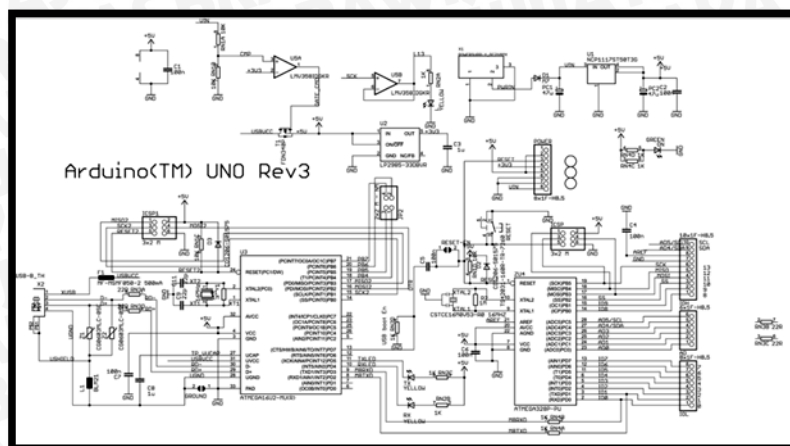
Gambar 4.26 Grafik respon dengan $k_p=0.1$ $k_i=0.0000005$ $k_d=15$

$$\begin{aligned}\% Mp &= \frac{|RPM_{puncak} - RPM_{steady}|}{RPM_{steady}} \times 100 \% \\ &= \frac{|511.92 - 405.38|}{405.38} \times 100 \% \\ &= 26.28\%\end{aligned}$$

Dari hasil *tuning* parameter kontroler diperoleh nilai $K_p = 0.1$ dan $K_i = 0,0000001$ dan $k_d=15$. Dengan nilai $K_p = 0.1$, $K_i = 0,0000001$ dan $k_d=15$ didapatkan respon yang baik dengan *maximum overshoot* (M_p) yaitu 21.23% dan *error steady state* (Ess) dibawah 2%.

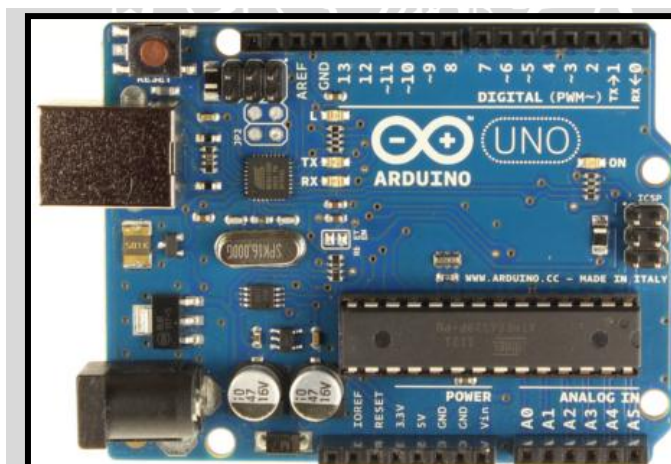
4.8 Modul Arduino Uno Rev.3

Pada sistem pengendalian kecepatan putaran ekstraktor madu ini digunakan Arduino Uno Rev.3 sebagai pengolah data dalam proses pengaturan kecepatan terdapat pada *Driver* motor *H-Bridge* untuk menggerakkan motor DC. Konfigurasi kaki I/O dari Arduino Uno Rev3 ditunjukkan dalam Gambar 4.27.



Gambar 4.27 Desain Sistem Arduino Uno 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 yang ke adaptor-DC atau baterai untuk menjalankannya. Modul arduino uno ditunjukkan pada gambar 4.28.



Gambar 4.28 Modul Arduino Uno

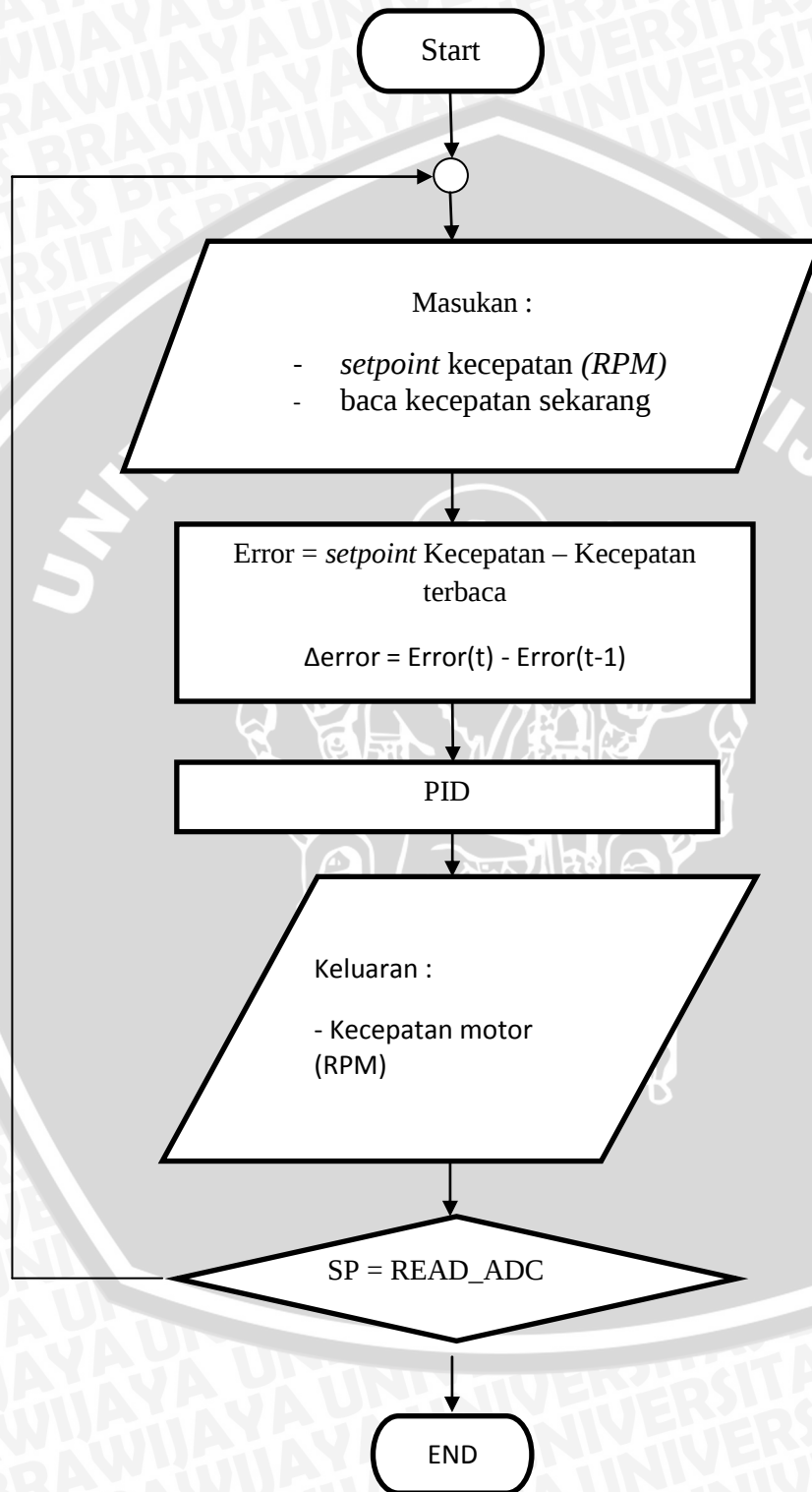
Sumber: www.Arduino.cc

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 4.1 Fungsi Pin Arduino Uno Rev3

4.9 Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.29 berikut:



Gambar 4.29 Flowchart Program