

BAB V

PENGUJIAN DAN ANALISIS SISTEM

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

5.1 Pengujian Sensor Kecepatan

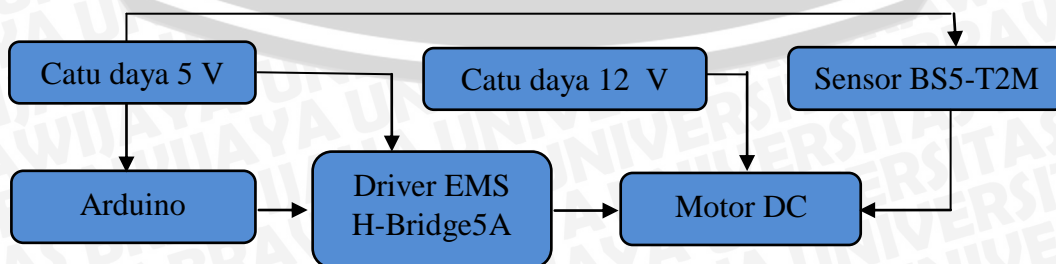
Pengujian sensor kecepatan dilakukan untuk mengetahui RPM yang terbaca pada sensor kecepatan *photo sensor* BS5-T2M telah sesuai dengan yang diinginkan.

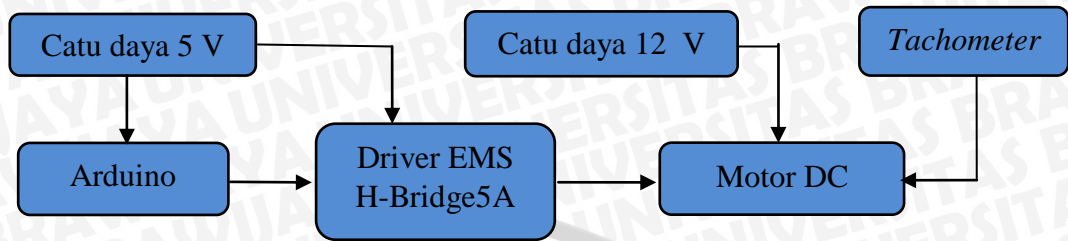
5.1.1 Peralatan yang digunakan

1. Catu daya 5 V dan 12V
2. Sensor kecepatan *photo sensor* BS5-T2M
3. Motor *Direct Current* (DC)
4. Program dan *software* Arduino
5. *Driver* motor
6. *Tachometer*

5.1.2 Prosedur Pengujian

1. Merangkai peralatan seperti pada Gambar 5.1.
2. Program kecepatan berupa *Pulse Width Modulation* (PWM) dimasukkan ke dalam Arduino.
3. Mengaktifkan catu daya 5V dan 12V.
4. Mencatat data kecepatan yang diperoleh dari pembacaan serial monitor dan *tachometer*.
5. Membandingkan data kecepatan dari kedua pembacaan.





Gambar 5.1 Rangkaian Pengujian Sensor *Photo Sensor BS5-T2M* dan *Tachometer*

5.1.3 Hasil Pengujian

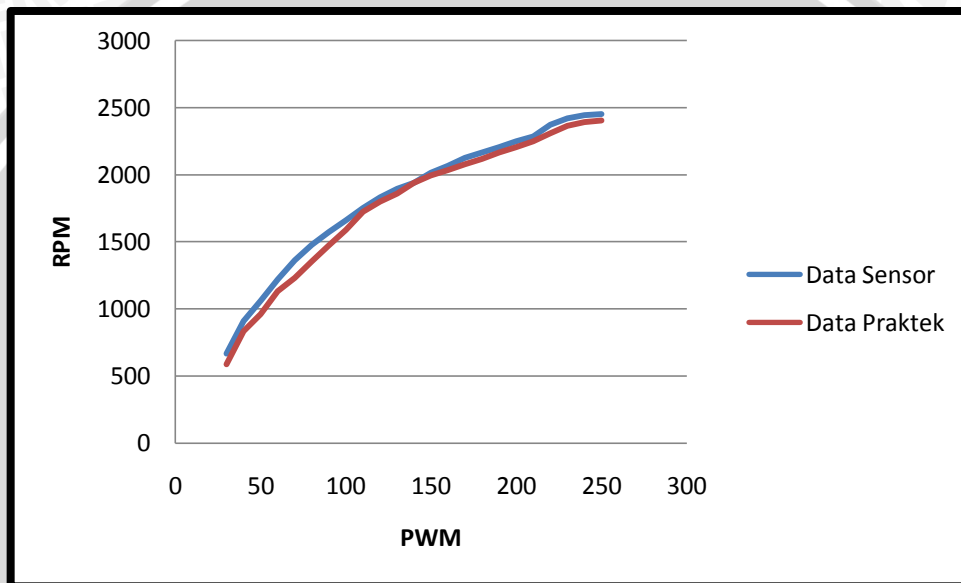
Setelah melakukan prosedur pengujian, didapatkan data hasil pengujian motor *Direct Current* (DC) dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan. Data hasil pengujian ditunjukkan pada Tabel 5.1 berikut.

Tabel 5.1 Hasil Perbandingan Nilai Pembacaan Kecepatan Sensor dan Praktek

| PWM | RPM sensor | RPM Tachometer |
|-----|------------|----------------|
| 0 | 0 | 0 |
| 10 | 0 | 0 |
| 20 | 0 | 0 |
| 30 | 664.44 | 585 |
| 40 | 907.2 | 830 |
| 50 | 1059.24 | 960 |
| 60 | 1218.84 | 1130 |
| 70 | 1359.96 | 1230 |
| 80 | 1475.88 | 1355 |
| 90 | 1573.32 | 1475 |
| 100 | 1659 | 1590 |
| 110 | 1752.24 | 1725 |
| 120 | 1829.52 | 1800 |
| 130 | 1894.2 | 1860 |
| 140 | 1938.72 | 1940 |
| 150 | 2014.32 | 1995 |
| 160 | 2065.56 | 2035 |
| 170 | 2125.2 | 2080 |
| 180 | 2167.2 | 2120 |

| | | |
|-----|---------|------|
| 190 | 2206.68 | 2165 |
| 200 | 2248.68 | 2205 |
| 210 | 2286.48 | 2250 |
| 220 | 2370.48 | 2310 |
| 230 | 2420.88 | 2365 |
| 240 | 2444.4 | 2395 |
| 250 | 2451.96 | 2405 |

Dari tabel 5.1 diperoleh grafik hubungan antara kecepatan dengan *Pulse Width Modulation* (PWM) seperti pada Gambar 5.2.



Gambar 5.2 Grafik Perbandingan Nilai Pembacaan Sensor Kecepatan dengan Data Praktek

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa hasil pengujian sensor *photo sensor* sesuai dengan data pengukuran secara praktek menggunakan *tachometer* dan sensor mampu berjalan dengan baik.

5.2 Pengujian Driver Motor Direct Current (DC)

Pengujian driver motor *Direct Current* (DC) ini bertujuan untuk mengetahui *output driver* motor yang dibandingkan dengan masukannya yang kemudian dapat diketahui juga hubungan keluaran *Pulse Width Modulation* (PWM) dengan tegangan yang dibutuhkan untuk motor *Direct Current* (DC).

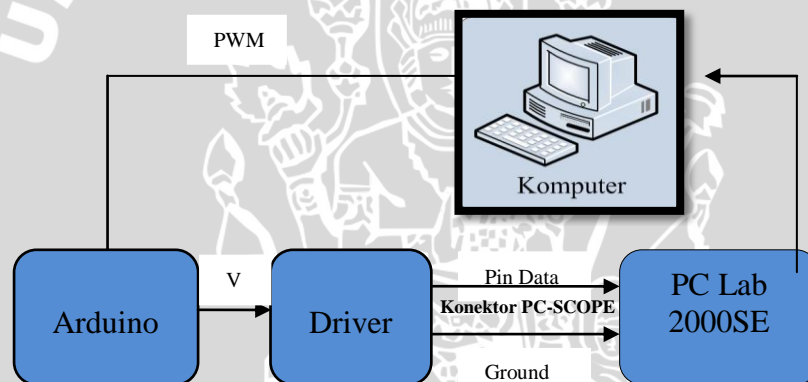
5.2.1 Peralatan yang Digunakan

1. Catu Daya 5V dan 12V

2. *Driver Motor EMS H-Bridge 5 A*
3. Motor *Direct Current (DC)*
4. Hardware dan *software PC Lab 2000SE*.
5. Program dan *software Arduino*.

5.2.2 Prosedur Pengujian

1. Merangkai peralatan seperti Gambar 5.3.
2. Mengunduh program yang mengeluarkan data berupa *Pulse Width Modulation (PWM)* dengan keluaran 10-250 dengan interval setiap 10 *Pulse Width Modulation (PWM)* pada *software arduino ERW 1.0.5*.
3. Jalankan *software PC Lab 2000SE*.
4. Pilih mode osiloskop lalu pilih *Run* untuk menjalankan osiloskop.
5. Mengamati sinyal kontrol berupa *duty cycle* pada osiloskop untuk setiap pengujian mikrokontroler dan *driver*.



Gambar 5.3 Rangkaian Pengujian Sinyal Kontrol *Driver*

5.2.3 Hasil Pengujian

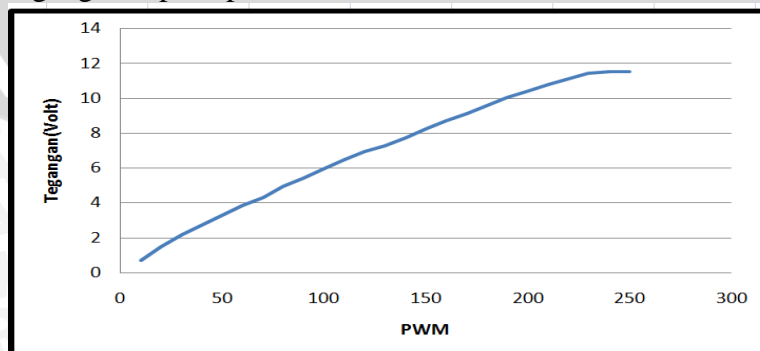
Setelah melakukan prosedur pengujian, didapatkan hasil sinyal dan tampilan parameter. Parameter yang dihasilkan adalah amplitudo, *duty cycle*, dan lebar sinyal.

Setelah mendapatkan data keluaran *duty cycle* hasil pengujian *driver* maka dapat diketahui besarnya tegangan yang dibutuhkan motor untuk setiap interval *Pulse Width Modulation (PWM)* yang diberikan. Data yang dihasilkan ditunjukkan pada Tabel 5.2

Tabel 5.2 Hasil Keluaran Tegangan Driver EMS H-Bridge 5 A

| PWM | DUTY Cycle Driver | Tegangan Driver(Volt) |
|-----|-------------------|-----------------------|
| 10 | 11.6 | 0.704 |
| 20 | 18.1 | 1.47 |
| 30 | 24.7 | 2,191 |
| 40 | 29.4 | 2,767 |
| 50 | 33.0 | 3,292 |
| 60 | 36.2 | 3.84 |
| 70 | 40.1 | 4.33 |
| 80 | 44.3 | 4.96 |
| 90 | 48.1 | 5.42 |
| 100 | 52.1 | 5.98 |
| 110 | 56.8 | 6.51 |
| 120 | 60.3 | 6.94 |
| 130 | 64.0 | 7.27 |
| 140 | 67.7 | 7.73 |
| 150 | 69.7 | 8.24 |
| 160 | 70.3 | 8.70 |
| 170 | 74.0 | 9.13 |
| 180 | 78.2 | 9.62 |
| 190 | 81.9 | 10.06 |
| 200 | 85.1 | 10.43 |
| 210 | 88.4 | 10.82 |
| 220 | ?? | 11.15 |
| 230 | ?? | 11.45 |
| 240 | ?? | 11.53 |
| 250 | ?? | 11.55 |

Dari tabel 5.2 diperoleh grafik hubungan antara *Pulse Width Modulation* (PWM) dengan keluaran tegangan seperti pada Gambar 5.4.



Gambar 5.4 Grafik Hubungan PWM dengan Tegangan Keluaran Driver

5.3 Pengujian Kecepatan Motor DC

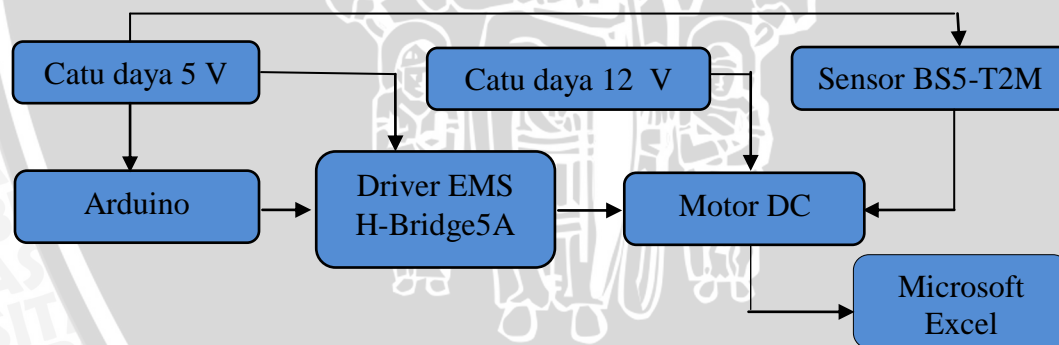
Pengujian kecepatan motor dilakukan untuk mengetahui hubungan PWM dengan kecepatan motor.

5.3.1 Peralatan yang digunakan

1. Catu daya 5 V dan 12V
2. Sensor kecepatan *photo sensor* BS5-T2M
3. Motor *Direct Current*(DC)
4. Program dan *software* Arduino
5. *Driver* motor EMS H-Bridge 5A

5.3.2 Prosedur Pengujian

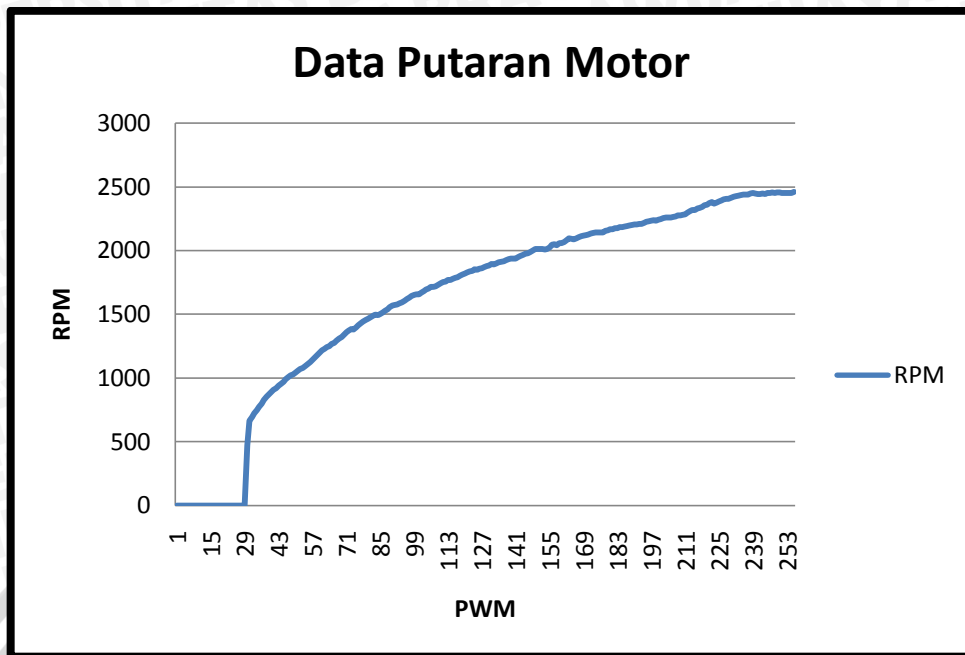
1. Merangkai peralatan seperti pada Gambar 5.5
2. Mengunduh program kecepatan berupa *Pulse Width Modulation* (PWM) dimasukkan ke dalam Arduino.
3. Mengaktifkan catu daya 5V dan 12V.
4. Mencatat data kecepatan yang diperoleh dari pembacaan serial monitor.



Gambar 5.5 Rangkaian Pengujian Kecepatan Motor

5.3.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan data hasil pengujian motor *Direct Current* (DC) dengan nilai *Pulse Width Modulation* (PWM) sebesar 0-255, kemudian diperoleh respon kecepatan motor yang ditunjukkan pada Gambar 5.6 berikut.



Gambar 5.6 Grafik Hubungan PWM dengan Kecepatan Motor

Dari gambar 5.6 dapat dilihat motor mulai berputar di kisaran PWM 29, kecepatan motor *Direct Current* (DC) mengalami perubahan yang besar dalam kisaran PWM 30–120 sedangkan pada PWM 121–255 perubahan kecepatan tidak begitu besar.

5.4 Pengujian Sistem Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk mengetahui kerja dari perangkat keras dan perangkat lunak setelah diintegrasikan dalam sebuah sistem terpadu.

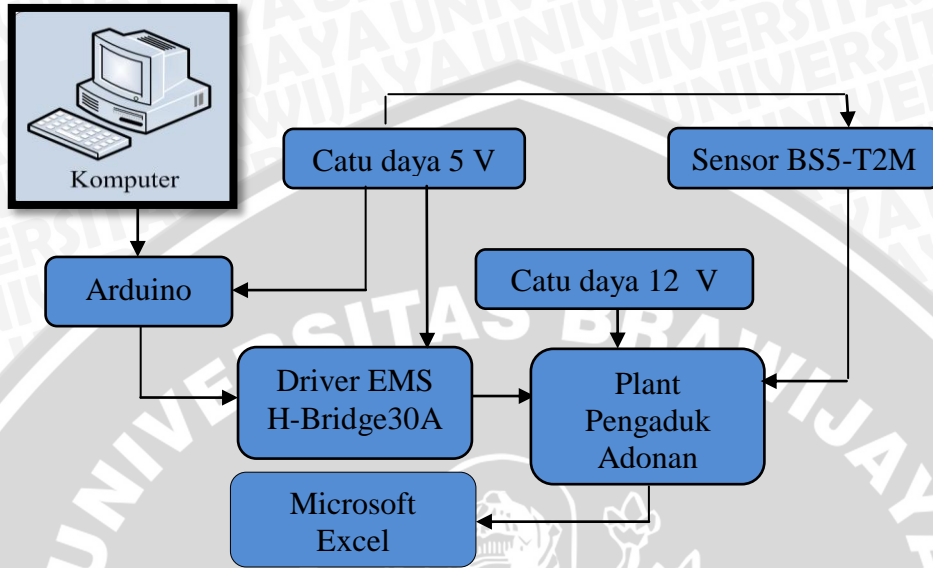
5.4.1 Peralatan yang digunakan

1. Catu daya 5 V dan 12V
2. Alat pengaduk adonan lengkap dengan sensor kecepatan *photo sensor* BS5-T2M dan Motor *Direct Current*(DC)
3. Program dan *software* Arduino
4. *Driver* motor EMS H-Bridge 5
5. Beban adonan

5.4.2 Prosedur Pengujian

Pengujian alat ini dilakukan dengan cara merangkai rangkaian seperti pada gambar kemudian mengunduh program $k_p=0.95$ $k_i= 0.00000001$ dan $k_d =100$ pada *software* arduino uno. Nilai keluaran pembacaan jarak oleh sensor kemudian ditampilkan pada serial monitor *software* Arduino Uno. Setelah itu mengamati hasil keluaran berupa

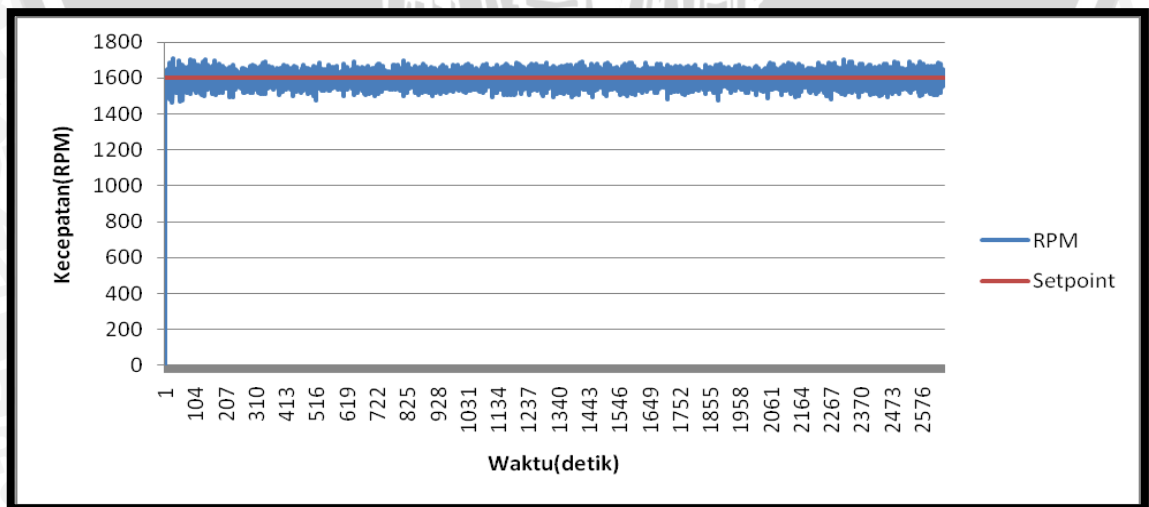
nilai RPM yang terbaca kemudian membuat grafik kecepatan terhadap waktu yang menampilkan data setiap 1 detik. Langkah terakhir mencatat *error* yang menjadi hasil pembacaan sensor dengan *setpoint* yang diinginkan.



Gambar 5.7 Rangkaian Pengujian Sistem Secara Keseluruhan

5.4.3 Hasil Pengujian

Setelah melakukan prosedur pengujian dengan nilai $k_p = 0.95$ $k_i = 0.00000001$ $k_d = 100$ didapatkan hasil keluaran sistem. Berikut hasil pengujian sistem keseluruhan ditunjukkan pada Gambar 5.8.



Gambar 5.8 Grafik Respon Keseluruhan

Dari grafik Gambar 5.8, diketahui bahwa hasil respon memiliki % *error* sebagai berikut:

$$\begin{aligned}\% E_{ss} &= \frac{|Average\ speed\ steady - setpoint|}{setpoint} \times 100\ \% \\ &= \frac{|1599.29 - 1600|}{1600} \times 100\% \\ &= 0.3\ \%\end{aligned}$$

Dari hasil penghitungan didapatkan *error* sebesar 0.3% dan waktu *steady* 3 detik, sistem dikatakan cukup baik karena *error* yang didapatkan masih di bawah dalam toleransi 2%.

