

## BAB IV

### PENGUJIAN SISTEM

Pengujian sistem yang dilakukan penulis merupakan pengujian terhadap perangkat keras dan .perangkat lunak dari sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui komponen-komponen sistem apakah berjalan dengan baik sesuai yang diharapkan. Terdapat beberapa pengujian sistem, antara lain :

#### **4.1 Pengujian *Minimum system***

##### **1.1.1 Tujuan**

Pengujian *minimum system* bertujuan untuk mengetahui apakah *minimum system* dapat melakukan proses *signature* dan *download* program ke *microcontroller* dengan baik.

##### **1.1.2 Alat yang Digunakan**

1. Rangkaian *minimum system* dengan komunikasi SPI.
2. Laptop.
3. Kabel *downloader*.
4. Program *Codevision AVR*.
5. Baterai 1200 mA-12V.

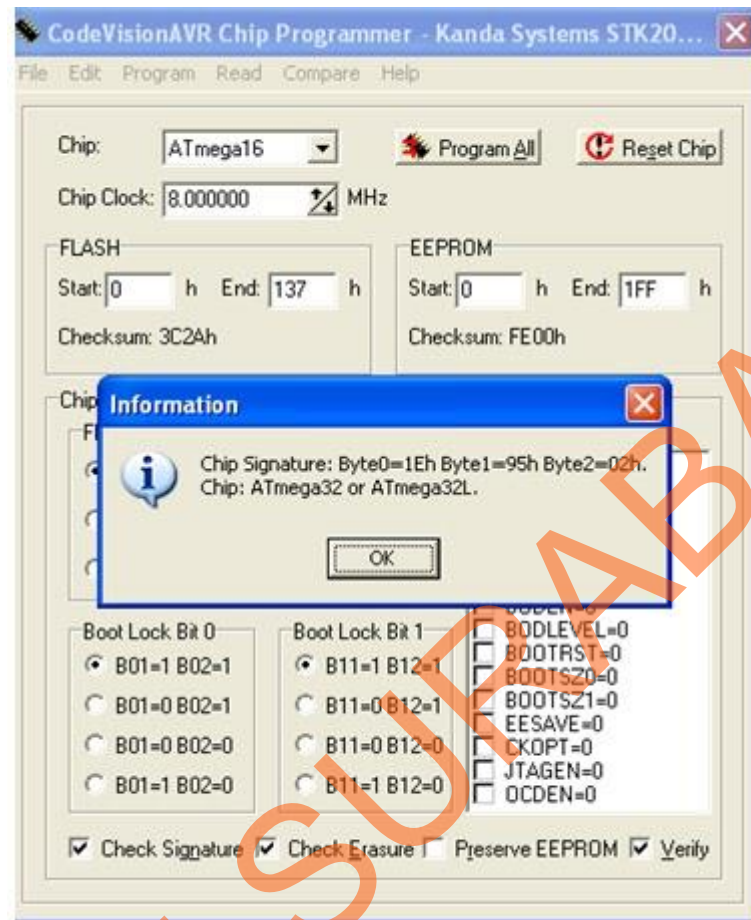
##### **1.1.3 Prosedur pengujian**

1. Aktifkan *power supply* dan hubungkan dengan *minimum system*.
2. Sambungkan *minimum system* dengan kabel *downloader* pada *port parallel*.
3. Selanjutnya aktifkan laptop dan jalankan program *CodeVision AVR*.

4. Untuk *download* program yang telah dibuat ke dalam *minimum system* maka yang harus dilakukan adalah menjalankan menu *chip signature programmer* pada *Codevision AVR*.
5. Setelah proses *signature* selesai maka selanjutnya proses *compile project* dengan menekan F9 pada *keyboard* kemudian proses *download* program ke *microcontroller* masuk ke menu → *make project* pada *Codevision AVR*.

#### 1.1.4 Hasil Pengujian

Dari percobaan diatas apabila menu *chip signature programmer*, *download* program dapat berhasil dikerjakan maka *minimum system* dapat dikatakan bekerja dengan baik. Tampilan dari program *chip signature* pada *Codevision AVR* yang akan digunakan untuk menuliskan program dan melakukan percobaan terhadap *minimum system*. Hasil program *chip signature* dapat dilihat pada Gambar 4.1. berikut :



Gambar 4.1. Tampilan *chip signature*

Pada Gambar 4.1 menunjukkan bahwa *minimum system* telah berhasil *download program* ke *microcontroller* sehingga program telah berhasil dijalankan.

## 1.2 Pengujian Sensor *OptoCoupler*

### 1.2.1 Tujuan

Pengujian terhadap sensor *optocoupler* dilakukan untuk mengetahui apakah sensor tersebut dapat menghasilkan *output* berupa sinyal digital 0 atau 1, sehingga dapat diterapkan sebagai *feedback* dari metode *fuzzy* maupun kontrol PID.

### 1.2.2 Alat yang Digunakan

1. Power *supply fix* 5V.
2. Sensor *optocoupler*.
3. Penghalang.
4. Multimeter.

### 1.2.3 Prosedur Pengujian

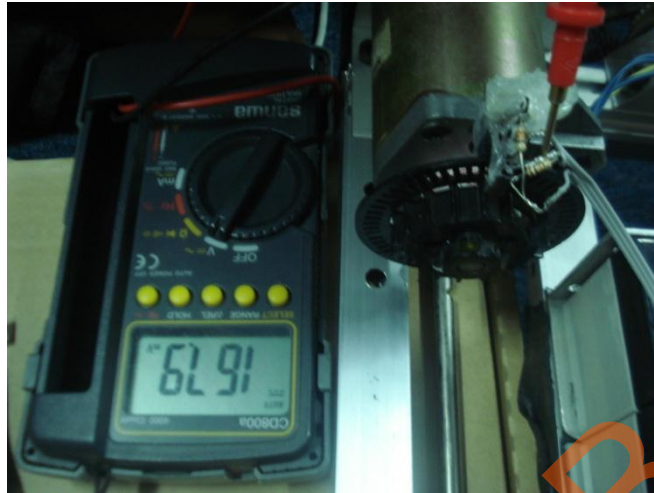
1. Aktifkan *power supply fix* 5V dan hubungkan VCC dengan *pin anode* pada *optocoupler* dan *ground* dengan *pin cathode*.
2. Beri penghalang kertas pada sela-sela sensor.
3. Ukur pada *pin output* dengan multimeter, jika diberi penghalang *output* dapat mengeluarkan logika *high* dan jika tidak diberi penghalang *output* dapat mengeluarkan logika *low*, maka sensor tersebut berjalan normal.

### 1.2.4 Hasil Pengujian

Dari hasil pengujian didapatkan bahwa sensor *optocoupler* dapat berjalan sesuai dengan yang diharapkan. Gambar berikut merupakan hasil pengujian yang telah dilakukan.



Gambar 4.2 Pengujian *rotary encoder* dengan penghalang



Gambar 4.3 pengujian *rotary encoder* dengan tidak menggunakan penghalang

Tabel 4.1 Hasil pengujian

Penghalang	Tegangan
✓	4,89 V/ <i>high</i>
-	0,05V/ <i>low</i>

### 1.3 Pengujian Kontrol PID tanpa Belt

#### 1.3.1 Tujuan

Pengujian kontrol PID dilakukan untuk mengetahui apakah rumusan yang digunakan dalam pembuatan kontrol PID dapat berjalan sesuai dengan yang diharapkan dan dapat menjalankan perintah dari *inputan fuzzy* untuk mengontrol motor DC.

#### 1.3.2 Alat yang Digunakan

1. Rangkaian *minimum system*.
2. Motor driver.
3. Sensor *optocoupler* sebagai *feedback* kecepatan motor DC.
4. Baterai 1200mA-12V DC.

5. Motor DC.
6. *USB to serial* atau kabel serial.
7. Laptop.

### 1.3.3 Prosedur Pengujian

1. Nyalakan *minimum system* yang telah diberikan tegangan.
2. Nyalakan motor driver.
3. Setting parameter PID dengan  $K_p = 0.5$ ,  $K_i = 0.0001$ , dan  $K_d = 0.0001$ .
4. Sambungkan *USB To Serial* dari rangkaian *minimum system* ke laptop.
5. Nyalakan program pengujian pada laptop menggunakan *Visual Basic 6.0* dan kemudian *running* program.
6. Program *Visual Basic 6.0* akan menampilkan rpm (*rotation per minutes*) yang digambarkan dalam dua buah grafik yaitu grafik rpm motor kanan dan grafik rpm motor kiri.

### 1.3.4 Hasil Pengujian

Dari pengujian yang dilakukan di atas, didapatkan hasil sebagai berikut :

1. Pengujian kontrol PID pada motor DC

Tabel 4.2. Hasil pengujian kontrol PID

Pengujian Ke-	Setpoint (Rpm)	Kecepatan Aktual Motor Kanan (Rpm)	Kecepatan Aktual Motor Kiri (Rpm)
1.	1200	1200	1200
2.	1000	960-1020	960-1020
3.	700	660-720	660-720
4.	500	480-530	480-530
5.	300	180-360	180-360



Gambar 4.4 Pengujian PID pada motor DC dengan setpoint 1200 rpm.



Gambar 4.5 Pengujian PID pada motor DC dengan *setpoint* 1000 rpm

Dari pengujian kontrol PID tanpa menggunakan konversi belt terlihat bahwa risetime antara motor kiri dan motor kanan hampir sama dan juga bisa mencapai kestabilan hampir bersamaan pada detik ke-8.

## 1.4 Pengujian Kontrol PID dengan Belt

### 1.4.1 Tujuan

Pengujian kontrol PID dilakukan untuk mengetahui apakah rumusan yang digunakan dalam pembuatan kontrol PID dapat berjalan sesuai dengan yang diharapkan dan dapat menjalankan perintah dari *inputan fuzzy* untuk mengontrol motor DC.

### 1.4.2 Alat yang Digunakan

1. Rangkaian *minimum system*.
2. Motor driver.
3. Sensor *optocoupler* sebagai *feedback* kecepatan motor DC.
4. Baterai 1200mA-12V DC.
5. Motor DC.
6. Belt
7. *USB to serial* atau kabel serial.
8. Laptop.

### 1.4.3 Prosedur Pengujian

1. Nyalakan *minimum system* yang telah diberikan tegangan.
2. Nyalakan motor driver.
3. Setting parameter PID dengan  $K_p = 0.5$ ,  $K_i = 0.0001$ , dan  $K_d = 0.0001$ .
4. Sambungkan *USB To Serial* dari rangkaian *minimum system* ke laptop.



5. Nyalakan program pengujian pada laptop menggunakan *Visual Basic 6.0* dan kemudian *running* program.
6. Program *Visual Basic 6.0* akan menampilkan rpm (*rotation per minutes*) yang digambarkan dalam dua buah grafik yaitu grafik rpm motor kanan dan grafik rpm motor kiri.

#### 1.4.4 Hasil Pengujian

Pengujian kontrol PID pada motor DC dengan menggunakan konversi belt

Tabel 4.3. Hasil pengujian kontrol PID

Pengujian Ke-	Setpoint (Rpm)	Kecepatan Actual Motor Kanan (Rpm)	Kecepatan Actual Motor Kiri (Rpm)
1.	1000	980-1020	980-1020
2.	800	780-840	780-840
3.	700	660-720	660-720
4.	500	420-480	420-480
5.	300	240-360	240-360



Gambar 4.6 Pengujian PID menggunakan Belt dengan Setpoint 1000 Rpm.



Gambar 4.7 Pengujian PID menggunakan belt dengan setpoint 800 rpm.

Dari pengujian kontrol PID dengan menggunakan konversi belt pada motor kanan dan motor kiri secara bersamaan dapat disimpulkan bahwa *output* sistem antara motor kanan dan *output* sistem motor kiri dapat mencapai *set point* yang diharapkan, meskipun dalam pengujiaanya *responstime* kedua motor sama. osilasi yang dihasilkan masih cukup tinggi dikarenakan kurang tepatnya dalam pencarian konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  dan kurang sempurnanya dalam pembuatan konversi belt.

## 1.5 Pengujian Kontrol PID robot berjalan

### 1.5.1 Tujuan

Pengujian kontrol PID dilakukan untuk mengetahui apakah rumusan yang digunakan dalam pembuatan kontrol PID dapat berjalan ketika robot berjalan sesuai dengan yang diharapkan dan dapat menjalankan perintah dari *inputan fuzzy* untuk mengontrol motor DC.

### 1.5.2 Alat yang Digunakan

1. Rangkaian *minimum system*.
2. Motor driver.
3. Sensor *optocoupler* sebagai *feedback* kecepatan motor DC.
4. Baterai 1200mA-12V DC.
5. Motor DC.
6. Belt
7. Robot
8. *USB to serial* atau kabel serial.
9. Laptop.

### 1.5.3 Prosedur Pengujian

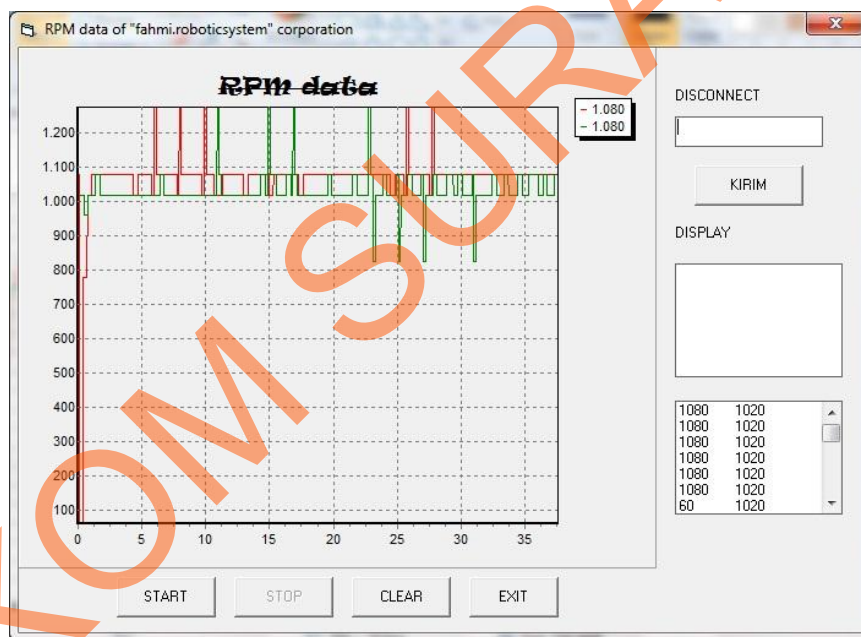
1. Nyalakan *minimum system* yang telah diberikan tegangan.
2. Nyalakan motor driver.
3. Setting parameter PID dengan  $K_p = 0.5$ ,  $K_i = 0.0001$ , dan  $K_d = 0.0001$ .
4. Sambungkan *USB To Serial* dari rangkaian *minimum system* ke laptop.
5. Nyalakan program pengujian pada laptop menggunakan *Visual Basic 6.0* dan kemudian *running* program.
6. Program *Visual Basic 6.0* akan menampilkan rpm (*rotation per minutes*) yang digambarkan dalam dua buah grafik yaitu grafik rpm motor kanan dan grafik rpm motor kiri.

### 1.5.4 Hasil Pengujian

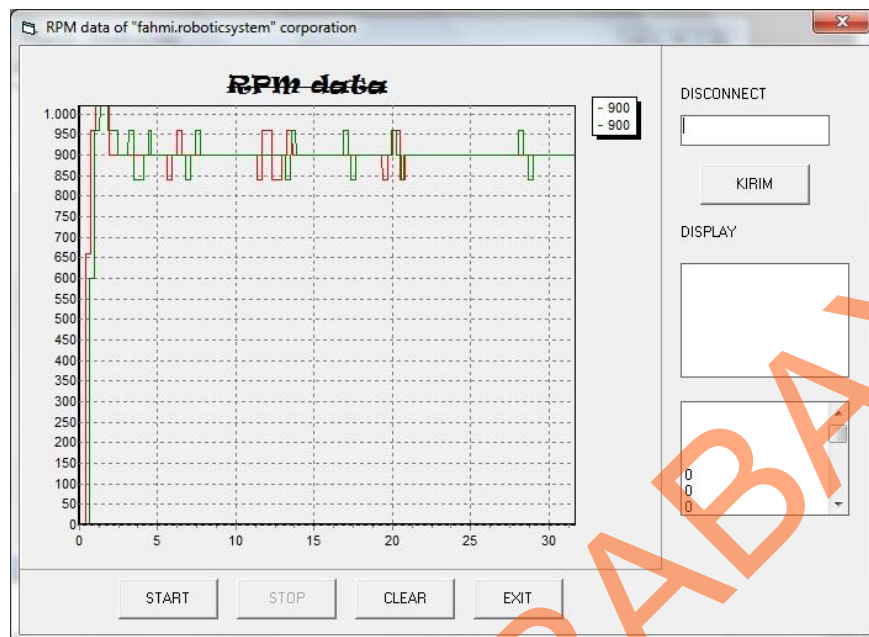
Pengujian kontrol PID pada saat robot berjalan

Tabel 4.4. Hasil pengujian kontrol PID

Pengujian Ke-	Setpoint (Rpm)	Kecepatan Aktual Motor Kanan (Rpm)	Kecepatan Aktual Motor Kiri (Rpm)
1.	1100	1020-1080	1020-1080
2.	900	850-950	850-950
3.	800	780-840	780-840
4.	600	550-660	550-660
5.	300	240-360	240-360



Gambar 4.8 Pengujian PID pada saat robot berjalan *setpoint* 1100 rpm.



Gambar 4.9 Pengujian PID pada saat robot berjalan *setpoint* 900 rpm.

Dari pengujian kontrol PID pada saat robot berjalan dapat disimpulkan bahwa robot berbelok ke kiri. Hal ini dikarenakan *responstime* motor kiri lebih lambat dari pada *responstime* motor kanan.

## 1.6 Pengujian Fuzzy Logic

### 1.6.1 Tujuan

Pengujian *fuzzy logic* dilakukan untuk mengetahui apakah rumusan yang digunakan dalam pembuatannya dapat berjalan sesuai dengan yang diharapkan dan dapat memberikan keluaran berupa kecepatan yang akan dikirimkan ke *microcontroller Master* sebagai *set point* kontrol PID. Dalam pengujiannya juga bertujuan untuk mengetahui apakah sistem dapat merumuskan masukan dari *keypad* berupa jarak dan waktu agar didapatkan *output* yang diinginkan.

### 1.6.2 Alat yang Digunakan

1. Rangkaian *minimum system*.
2. Sensor *optocoupler* sebagai *feedback* dari jarak.
3. Baterai 1200mA-12V DC.
4. *USB to serial* atau kabel serial.
5. Laptop.
6. Lcd 16 x 2.
7. *Keypad*

### 1.6.3 Prosedur Pengujian

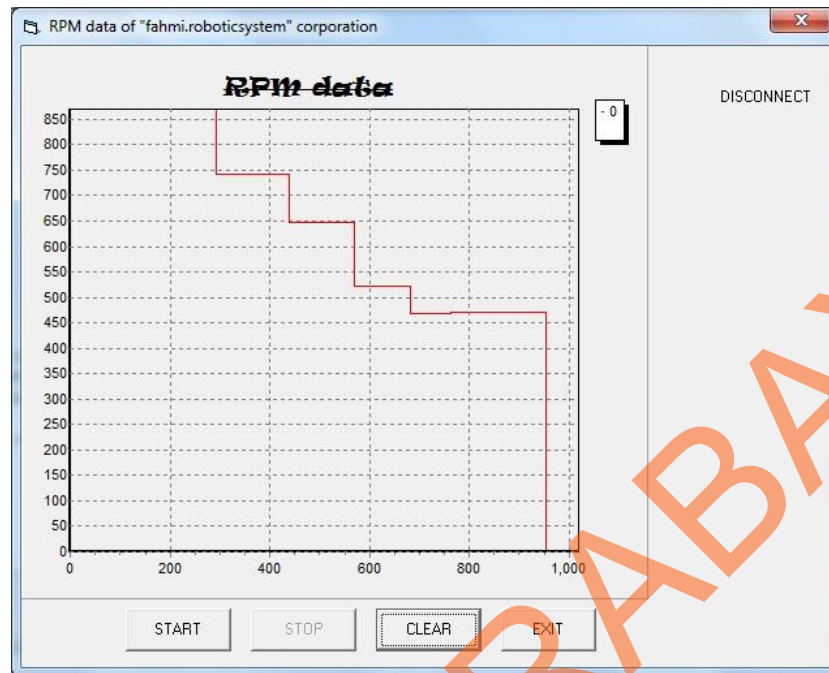
1. Nyalakan *minimum system* yang telah diberi tegangan.
2. Sambungkan *USB to serial* ke laptop.
3. *Running* program *Visual Basic 6.0* sebagai program pengujian.
4. Program akan menampilkan grafik *fuzzy logic* berupa jarak dan waktu. Jika ketika waktu dan jarak habis tepat sesuai yang diharapkan maka pengujian tersebut berhasil.

### 4.6.4 Hasil Pengujian

Dari pengujian yang dilakukan didapatkan hasil sebagai berikut :

Tabel 4.5 Hasil pengujian *fuzzy*

Pengujian Ke-	Input Jarak (meter)	Input Waktu (Second)	Output (Rpm)
1.	10	10	850
2.	12	12	850
3.	14	15	450
4.	16	15	700
5.	18	20	480



Gambar 4.10 Pengujian *fuzzy logic* dengan *input* 10 meter dalam 10 detik



Gambar 4.11 Pengujian *fuzzy logic* dengan *input* 12 meter dalam 12 detik

Dari pengujian *fuzzy logic* yang dilakukan penulis selama lima kali percobaan didapatkan hasil bahwa rpm yang dihasilkan oleh sistem sesuai dengan nilai yang diharapkan.

## **4.7 Pengujian Sistem Keseluruhan**

### **4.7.1 Tujuan**

Dengan adanya pengujian sistem secara keseluruhan dapat mengetahui apakah kedua metode yakni metode *fuzzy logic* dan metode kontrol PID berjalan sesuai dengan yang diharapkan.

### **4.7.2 Alat yang Digunakan**

1. Rangkaian *minimum system*.
2. Robot telah yang dibuat.
3. Lcd 16 x 2.
4. *Keypad*.
5. Sensor *optocoupler* sebagai *feedback* jarak.
6. Sensor *optocoupler* sebagai *feedback* kecepatan.
7. *USB to serial*/kabel serial.
8. Baterai 1200mA-12V DC.
9. Laptop.

### **4.7.3 Prosedur Pengujian**

1. Nyalakan *minimum system* yang telah diberi tegangan.
2. Sambungkan *USB to serial* ke laptop.
3. *Running* program *Visual Basic 6.0* sebagai program pengujian.
4. Program akan menampilkan grafik berupa rpm.

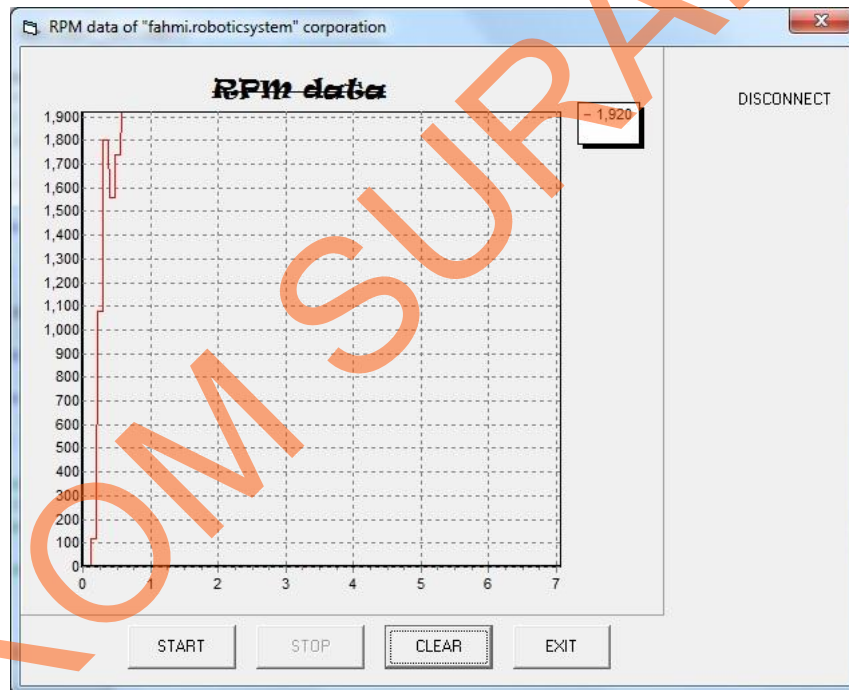


#### 4.7.4 Hasil Pengujian

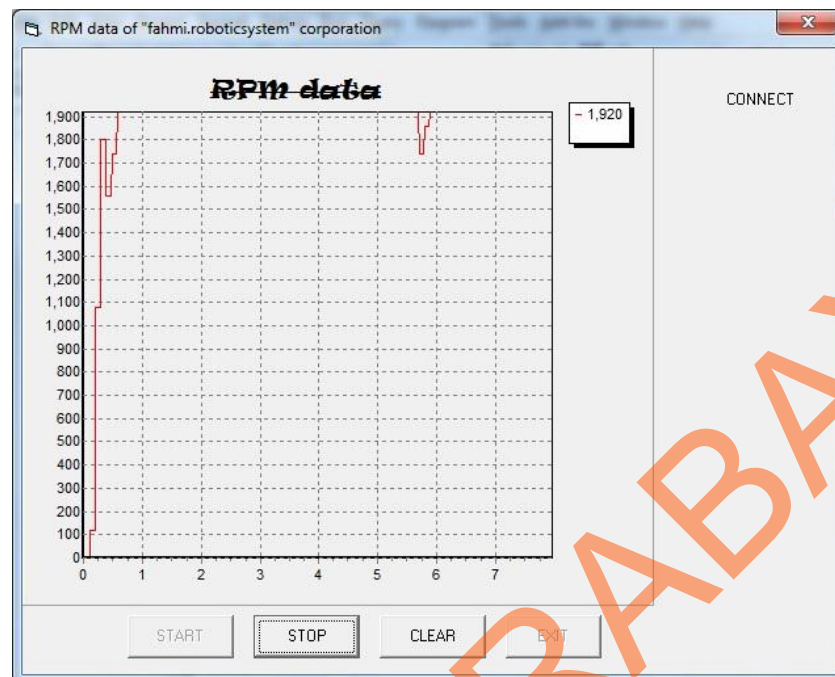
Dari pengujian yang dilakukan didapatkan hasil sebagai berikut :

Tabel 4.6 Hasil pengujian keseluruhan

Pengujian Ke-	Input Jarak (meter)	Input Waktu (Second)	Output (Rpm)
1.	10	10	1920 (gagal)
2.	12	12	1920 (gagal)
3.	14	15	1960 (gagal)
4.	16	15	1960 (gagal)
5.	18	20	1920 (gagal)



Gambar 4.12 Pengujian sistem keseluruhan



Gambar 4.13 Pengujian sistem keseluruhan

Dalam pengujian sistem keseluruhan dilakukan dengan menggabungkan kedua metode yakni metode *fuzzy logic* dan metode kontrol PID. Dalam pengujian penulis mengalami kegagalan dalam mendapatkan *output* yang sesuai yang diharapkan. Hal ini dikarenakan sistem kontrol PID yang dijalankan oleh ATmega8 sebagai mikro *Slave* tidak dapat mengolah data yang diterima dari metode *fuzzy logic* yang dijalankan oleh mikro *Master* ATmega16.