

BAB IV

PENGUJIAN DAN ANALISIS SISTEM

Pada bab ini akan dibahas hasil analisis pengujian telah dilakukan, pengujian dilakukan dalam beberapa bagian yang disusun dalam urutan dari yang sederhana menuju sistem yang lengkap.

Dari penggabungan perangkat keras dan perangkat lunak diharapkan didapat suatu sistem yang dapat mengendalikan *mobile robot* dengan pengendali PID yang bekerja dengan baik dan optimal.

4.1 Pengujian PORT *Input* dan *Output* Modul FPAA AN221K04-V4

4.1.1 Tujuan Pengujian

Menguji Modul FPAA *Anadigm* AN221K04-V4 Agar dapat bekerja sesuai dengan yang diharapkan

4.1.2 Alat yang dibutuhkan

1. Modul FPAA *Anadigm* AN221K04-V4
2. Komputer dan kabel USB *downloader*
3. *Power supply*
4. Multimeter
5. Software *Anadigm designer*

4.1.3 Prosedur Pengujian

1. Nyalakan modul FPAA AN221K04-V4.
2. Hubungkan PIN *input* dengan tegangan sumber yang bisa diatur.
3. *Download* program untuk pengujian pin I/O.

4. Ukur PIN *output* dengan multimeter.

4.1.4 Hasil Pengujian

Berikut pada Tabel 4.1 adalah tabel hasil pengujian pengukuran pin *output* FPAA AN221K04-V4

Tabel 4.1 Hasil Pengujian Port I/O

No	Tegangan <i>Input</i>							Tegangan <i>Output</i>	
	I1P I1N	I2P I2N	I3P I3N	I4PA I4NA	I4PB I4NB	I4PC I4NC	I4PD I4ND	O1P O1N	O2P O2N
1	1.2							-2.29	
		1.2						1.23	
			1.2					1.23	
				1.2				1.067	
					1.2			1.231	
						1.2		1.231	
							1.2	1.229	
2	1.2								-2.29
		1.2							1.24
			1.2						1.232
				1.2					1.075
					1.2				1.232
						1.2			1.233
							1.2		1.23

Dari Tabel 4.1 Pengujian *board* FPAA AN221K04-V4 PIN *input* I1P I1N tidak berfungsi dengan baik. Hal ini dikarenakan PIN negatif dari FPAA disambungkan *ground*. Hal ini tidak mengganggu penelitian, karena pada board FPAA AN221K04-V4 masih terdapat 5 pin *input*.

4.2 Pengujian *Driver Motor*

4.2.1 Tujuan Pengujian

Menguji *driver motor* apakah telah berfungsi dengan baik.

4.2.2 Alat yang Dibutuhkan

1. *Driver motor* LM339
2. *Power supply*

4.2.3 Prosedur Pengujian

1. Nyalakan *driver motor* LM339
2. Beri tegangan pada *dir1*, *dir2*, dan *enable driver motor*

4.2.4 Hasil Pengujian

Hasil pengujian *driver motor* terlihat pada Tabel 4.2.

Tabel 4.2 Pegujian *Driver motor*

Enable	Dir1	Dir2	Keterangan
1	1	0	Searah jarum jam
1	0	1	Berlawanan arah jarum jam
1	1	1	Motor berhenti
0	X	X	Motor berhenti

Dari hasil pengujian di atas dapat disimpulkan bahwa *driver motor* dapat berfungsi dengan baik sesuai dengan *datasheet*.

4.3 Pengujian *Microcontroller*

4.3.1 Tujuan Pengujian

Pengujian *Minimum system* bertujuan untuk mengetahui apakah *Microcontroller* dapat melakukan proses *signature* dan *download* program dengan baik.

4.3.2 Alat yang dibutuhkan

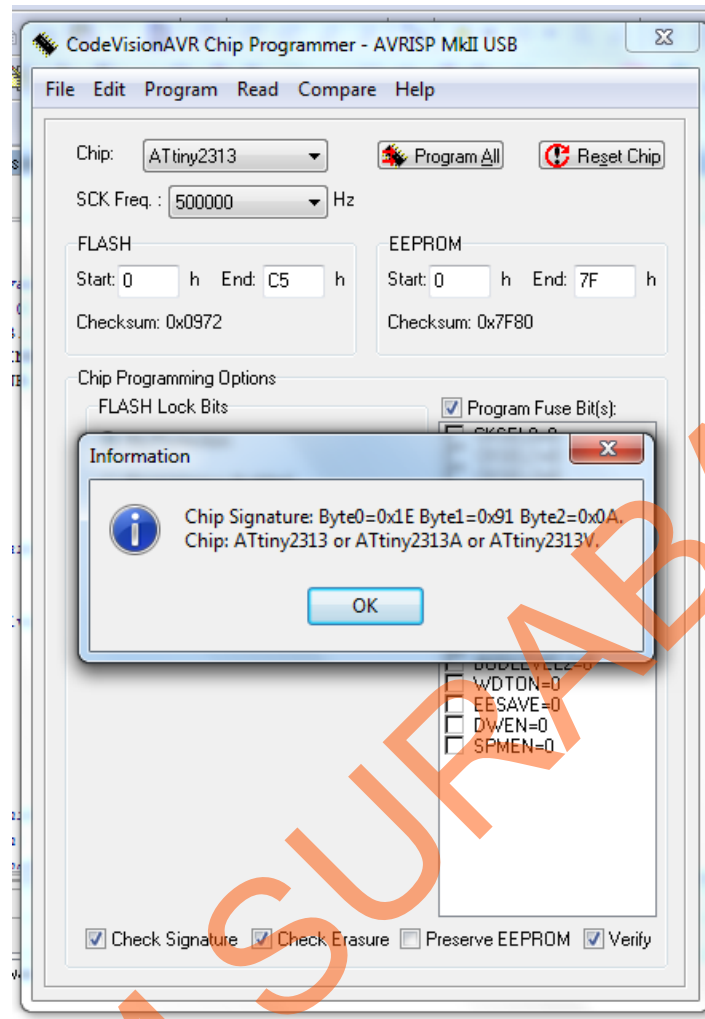
1. Rangkaian *minimum system*
2. Komputer.
3. Kabel *downloader*.
4. Program *CodeVisionAVR*.
5. *Power supply*

4.3.3 Prosedur Pengujian

1. Aktifkan *power supply* dan hubungkan dengan *minimum system*.
2. Sambungkan *minimum system* dengan kabel *downloader* pada *port parallel*.
3. Selanjutnya aktifkan komputer dan jalankan program *CodeVisionAVR*.
4. Untuk *download* program yang telah dibuat ke dalam *minimum system* maka yang harus dilakukan adalah menjalankan menu *Chip Signature programmer* pada *CodeVisionAVR*.
5. Setelah proses *signature* selesai maka selanjutnya proses *compile project* dengan menekan F9 pada *keyboard* kemudian proses *download* program ke *microcontroller* masuk ke menu kemudian *make project* pada *CodeVisionAVR*.

4.3.4 Hasil Pengujian

Dari percobaan di atas apabila menekan *Chip Signature programmer*, *download* program dapat berhasil dikerjakan maka *minimum system* dapat dikatakan bekerja dengan baik. Tampilan dari program *Chip Signature* pada *CodeVisionAVR* yang akan digunakan untuk menuliskan program dan melakukan percobaan terhadap *minimum system*. Hasil program *Chip Signature* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan *Chip Signature*

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

4.4 Pengujian Sensor Sharp GP2D120

4.4.1 Tujuan Pengujian

Tujuan dari pengujian sensor ini adalah untuk mengetahui apakah sensor dapat mengukur jarak dengan baik.

4.4.2 Alat yang dibutuhkan

1. Sensor jarak SHARP GP2D120
2. *Power supply*
3. Multimeter

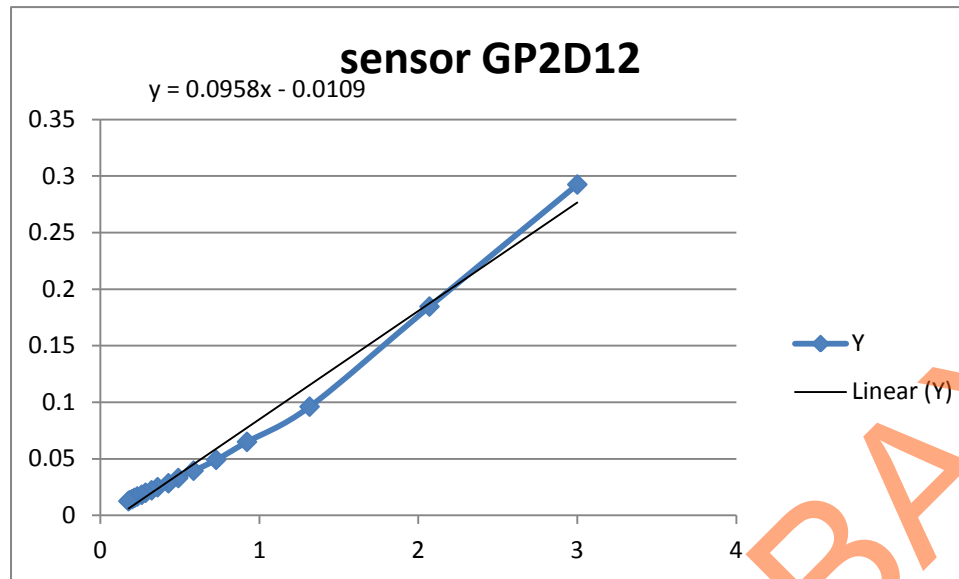
4.4.3 Prosedur Pengujian

1. Hubungkan sensor dengan *power supply* dengan tegangan 5V
2. Mengukur hasil pengukuran *output sensor* dengan menggunakan multimeter
3. Catat hasil pengukuran

4.4.4 Hasil Pengujian

Dari hasil pengujian di atas didapatkan hasil pengukuran. Hasil pengukuran dapat dilihat dari Gambar 4.2 dan Tabel 4.1. Data dari hasil pengujian ini akan digunakan FPAA untuk diolah dengan pengendali PID. Untuk catu daya sensor direkomendasikan menggunakan kapasitor 10 μ F (atau lebih) diletakkan diantara VCC dan *ground*. Kapasitor berfungsi untuk menstabilkan tegangan, karena sensor GP2D120 membutuhkan tegangan yang stabil.

Menurut [application note sensor GP2D120](#) mempunyai hubungan yang cukup linear antara tegangan keluaran sensor (V_{out}) dengan $1/(distance+K)$, K bernilai 0,42 untuk tipe GP2D120. Gambar 4.2 adalah grafik hubungannya. Sumbu X merupakan output sensor sedangkan sumbu Y adalah $1/(distance+K)$. Berdasarkan grafik diatas kita sudah dapat menggunakan persamaan metode Trendline yang ada pada MS. Exel.



Gambar 4.2 Grafik pengukuran sensor

Tabel 4.1 Tabel pengukuran *output* sensor

Jarak	Vout (V) X
3	3
5	2.07
10	1.316
15	0.922
20	0.728
25	0.586
30	0.49
35	0.428
40	0.36
45	0.322
50	0.284
55	0.26
60	0.232
65	0.214
70	0.2
75	0.184
80	0.176

4.5 Pengujian Kendali PID

4.5.1 Tujuan Pengujian

Pengujian kendali PID dilakukan untuk mengetahui apakah rumusan yang digunakan dalam pembuatan kendali PID dapat berjalan sesuai dengan yang diharapkan

4.5.2 Alat yang dibutuhkan

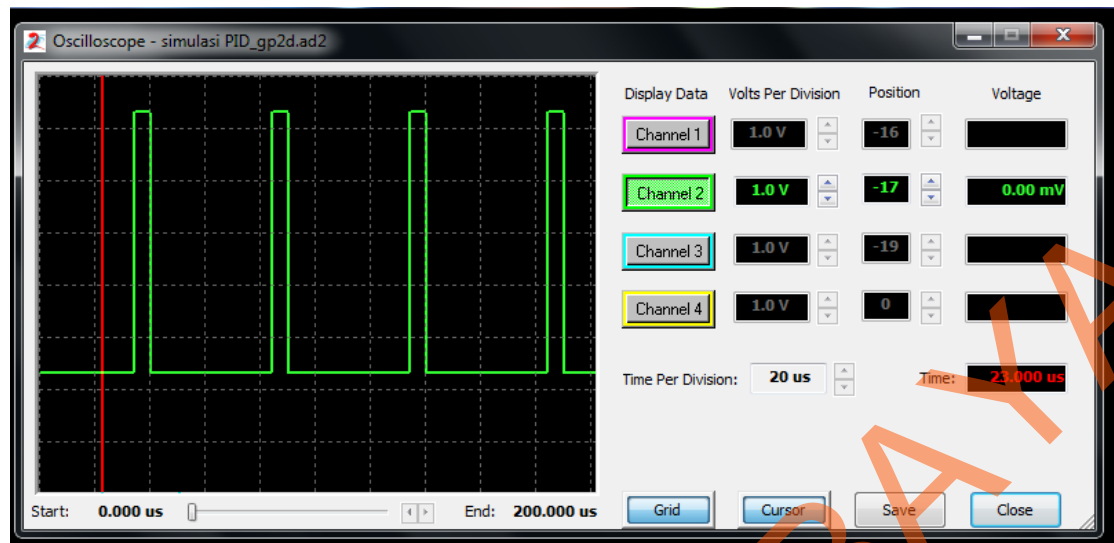
1. Modul FPAA
2. *Minimum system microcontroller ATTiny 2313*
3. *Driver motor*
4. Rangkaian sensor GP2D120

4.5.3 Prosedur Pengujian

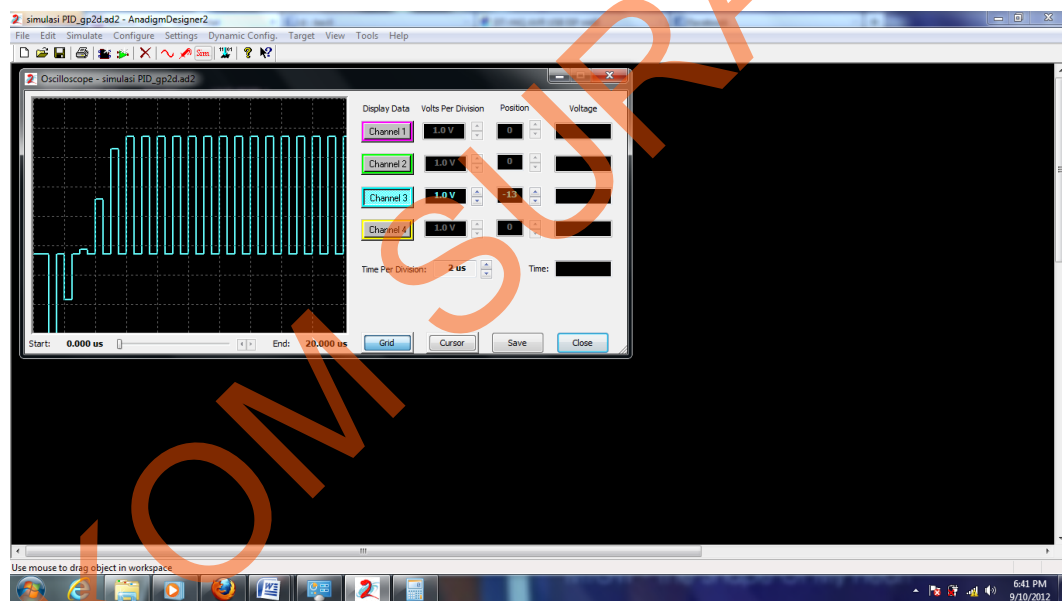
1. Nyalakan FPAA
2. Hubungkan sensor, pin PWM *driver motor*, dan *set point* ke FPAA
3. *Download program kendali PID*

4.5.4 Hasil Pengujian

Dari hasil penelitian ini dapat dilihat pada Gambar 4.3 dan Gambar 4.4. Ketika jarak sensor mendekati *set point* maka PWM akan seperti Gambar 4.3. Hasil dari penelitian kendali PID ini dapat memperbaiki respon kecepatan dengan nilai $K_p=4$, $K_I=3.6$, $K_D=3$. Gambar grafik dari respon kecepatan dapat dilihat pada Gambar 4.4.



Gambar 4.3 Pengujian PWM



Gambar 4.4 PID pada saat jarak 45 cm

4.6 Pengujian Pergerakan Kemudi

4.6.1 Tujuan Pengujian

Pengujian pergerakan kemudi dilakukan untuk mengetahui apakah pergerakan kemudi sesuai dengan sensor kiri, kanan, dan depan. Jika sensor kiri dan depan mendeteksi adanya halangan dan sensor kanan tidak mendeteksi maka

kemudi akan belok kanan. Jika sensor kiri tidak mendeteksi adanya halangan, sensor kanan dan sensor depan mendeteksi halangan maka kemudi belok kiri. Jika ketiga sensor tidak mendeteksi adanya halangan maka kemudi akan tetap lurus. Jika ketiga sensor mendeteksi adanya halangan maka kemudi akan belok kanan.

4.6.2 Alat yang dibutuhkan

1. Sensor GP2D120.
2. Rangkaian *comparator*.
3. *Minimum system*.
4. *Power supply*.

4.6.3 Prosedur Pengujian

1. Nyalakan *power supply*.
2. Hubungkan *power supply* dengan *microcontroller* dan rangkaian *comparator*.
3. Sambungkan *output* sensor sharp GP2D120 ke *input comparator*.
4. *Output* dari *comparator* dihubungkan ke *port input* dari *microcontroller*.
5. *Download* program untuk mengendalikan kemudi.

4.6.4 Hasil Pengujian

Tabel 4. 2 Pengujian kemudi

Sensor Depan	Sensor Kiri	Sensor Kanan	Kemudi
0	0	0	Lurus
1	0	1	Belok Kiri
1	1	0	Belok Kanan
1	1	1	Belok Kanan

Dari hasil Tabel 4.2 dapat disimpulkan bahwa kemudi dapat berjalan sesuai dengan yang diharapkan

4.7 Evaluasi Sistem Keseluruhan

Pengujian terakhir adalah pengujian system secara keseluruhan dari awal hingga akhir, dimana pengujian ini dilakukan dengan menjalankan aplikasi secara keseluruhan. Robot diletakkan di tempat yang lapang dengan disertai halangan. Dan kita lihat apakah robot dapat berhenti secara perlahan ketika mendekati jarak yang diinginkan, dan kemudian akan bergeser untuk menghindari halangan.

4.7.1 Tujuan Pengujian

Tujuan dari aplikasi ini adalah untuk mengetahui system pada aplikasi apakah dapat berjalan sesuai dengan yang diharapkan. Dimulai dari mendeteksi halangan, berhenti perlahan, dan kemudian berbelok.

4.7.2 Alat yang dibutuhkan

1. Modul FPAA
2. *Anadigm designer*
3. Komputer
4. *Obstacle*
5. Rangkaian *minimum system*
6. Rangkaian *set point*
7. Rangkaian *driver motor*
8. Rangkaian *comparator*
9. Rangkaian sensor
10. *Motor servo kemudi*

4.7.3 Prosedur Pengujian

1. Menyalakan FPAA
2. Hubungkan sensor, *set point*, dan pin PWM *driver motor* ke FPAA

3. Nyalakan *minimum system*
4. Hubungkan hasil *comparator* sensor kiri dan kanan ke *microcontroller*
5. Hubungkan motor *servo* kemudi ke *microcontroller*
6. *Download* program kemudi ke *microcontroller* dan program PID ke FPAA
7. Meletakkan *obstacle*

4.7.4 Hasil Pengujian

Setelah melalui seluruh prosedur pengujian di atas didapatkan hasil robot sudah dapat berhenti pada set point yang diinginkan oleh *user* dan kemudi dapat berbelok sesuai dengan yang diinginkan.