BAB IV

PENGUJIAN DAN ANALISIS

4.1 Pengujian Dan Analisis Sistem

Pada bab ini akan diuraikan tentang proses pengujian sistem yang meliputi pengukuran terhadap parameter-parameter dari setiap komponen per blok maupun secara keseluruhan, dan melakukan uji coba terhadap aplikasi alat yang diharapkan dapat berjalan sesuai perancangan pada bab sebelumnya, selanjutnya akan dilakukan analisis terhadap aplikasi hasil pengukuran tersebut. Pengujian dan analisis sistem ini bertujuan untuk mengetahui sistem kerja dari setiap komponen *input*, proses, dan *output* apakah dapat berjalan sesuai target yang diharapkan.

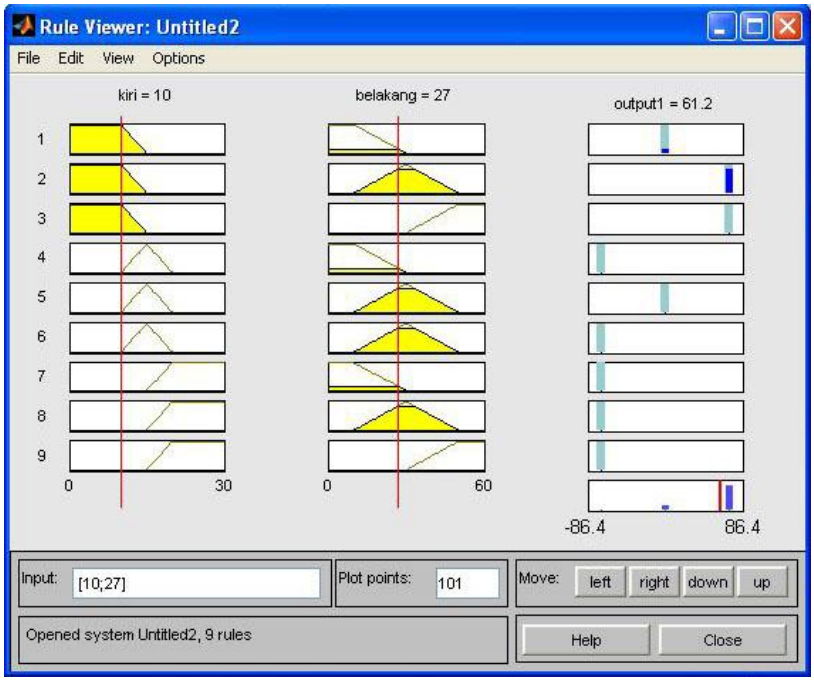
4.1.1 Pengujian Logika *Fuzzy*

Tujuan dari pengujian logika *Fuzzy* ini adalah untuk mengetahui proses logika *Fuzzy* ini sebagai inti dari sistem pemandu kendaraan untuk parkir paralel secara otomatis ini. Logika *Fuzzy* yang digunakan ini diprogram ke dalam mikrokontroler sebagai pusat pengendalian sistem. *Input*-an logika *Fuzzy* pada tugas akhir ini berupa jarak samping kiri yang didapat dari sensor SR-04 kiri-belakang dan jarak belakang dari *prototype* mobil *remote* yang didapat dari sensor SR-04 bagian belakang. Kedua *input*-an ini akan diproses di dalam mikrokontroler menggunakan logika *Fuzzy* yang akan menghasilkan *output* berupa perubahan sudut motor servo sebagai *steering* dari model.

Pengujian dari logika *Fuzzy* ini dilakukan dengan membandingkan hasil *output* yang didapat menggunakan simulasi Matlab, pengujian program yang dibuat menggunakan Codevison AVR dan pengujian perhitungan manual. Seperti yang dijelaskan pada bab sebelumnya, bahwa tahapan membangun sebuah logika *Fuzzy* terdiri dari 3 tahapan yaitu *Fuzzification*, *Fuzzy Logic Inference* dan *Defuzzification.* Setiap *input*-an akan diproses secara bertahap mulai dari menentukan *membership function* hingga menghasilkan *output*.

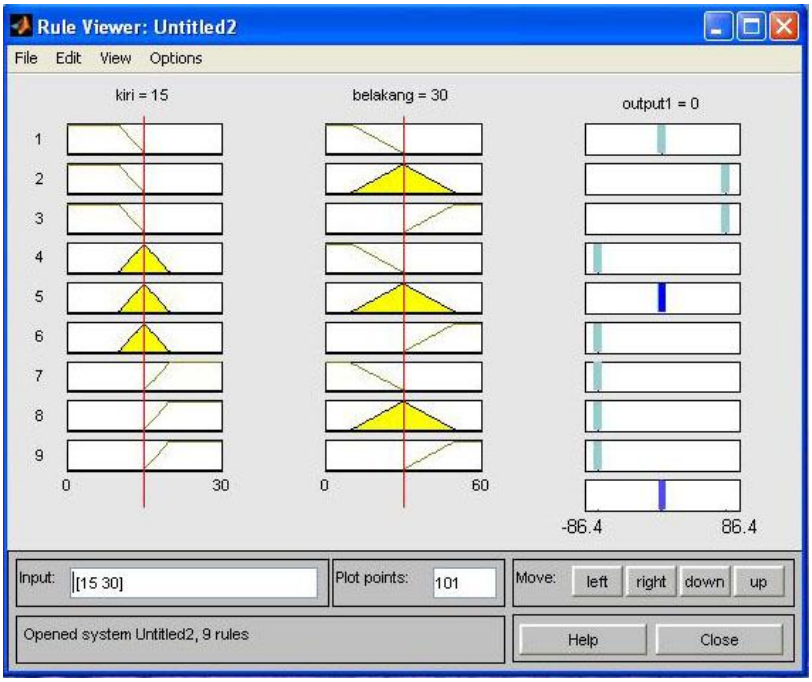
a. Hasil pengujian logika menggunakan simulasi Matlab :

- Jika *input* kiri-depan = 10cm dan belakang = 27cm



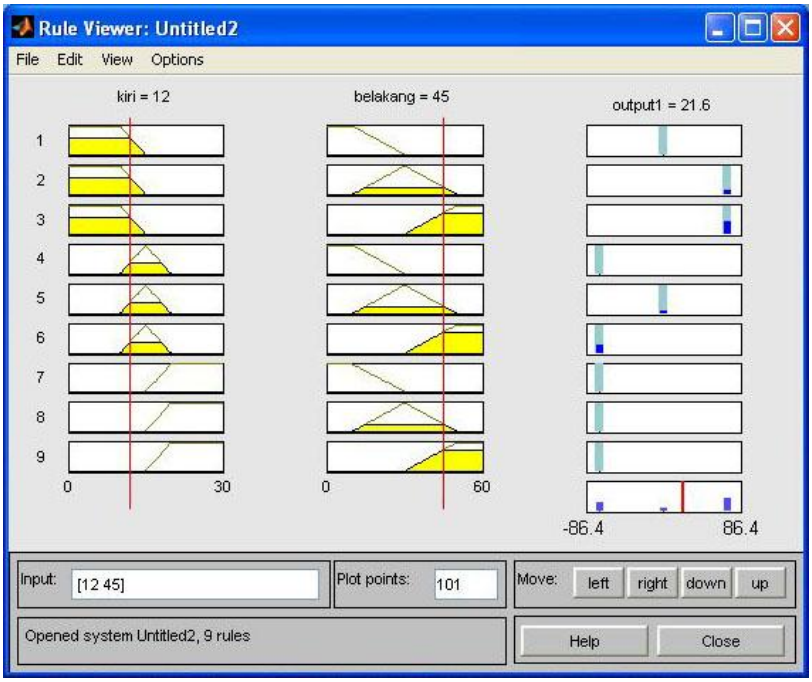
Gambar 4.1 Hasil Simulasi Matlab Pada Pengujian Pertama

- Jika *input* kiri-depan = 15cm dan belakang = 30cm



Gambar 4.2 Hasil Simulasi Matlab Pada Pengujian Kedua

- Jika *input* kiri-depan = 12cm dan belakang = 45cm



Gambar 4.3 Hasil Simulasi Matlab Pada Pengujian Ketiga

b. Hasil pengujian logika *Fuzzy* yang dibuat menggunakan software CodeVision AVR dan ditampilkan ke LCD:

- Jika *input* kiri-depan = 10cm dan belakang = 27cm



- Jika *input* kiri-depan = 15cm dan belakang = 30cm



- Jika *input* kiri-depan = 12cm dan belakang = 45cm



c. Hasil pengujian dengan perhitungan manual:

- Jika *input* kiri-depan = 10cm dan belakang = 27cm

Kiri-depan		
Input	=	10cm
MF dekat	=	1
MF sedang	=	0
MF jauh	=	0

Belakang		
Input	=	27cm
MF dekat	=	$\frac{30-27}{30-10} = \frac{3}{20} = 0{,}15$
MF sedang	=	$\frac{27-10}{30-10} = \frac{17}{20} = 0{,}85$
MF jauh	=	0

Rule Fuzzy

Kiri-depan / Belakang	**Dekat**	**Sedang**	**Jauh**
Dekat	0,15	0	0
Sedang	0,85	0	0
Jauh	0	0	0

Defuzzification

$$out = \frac{(0{,}15\times0)+(0{,}85\times72)+(0\times72)+(0\times-72)+(0\times0)+(0\times-72)+(0\times-72)+(0\times-72)+(0\times-72)}{0{,}15+0{,}85+0+0+0+0+0+0+0}$$

$$= 61.2$$

- Jika *input* kiri-depan = 15cm dan belakang = 30cm

Kiri-depan			Belakang		
Input	=	15cm	*Input*	=	30cm
MF dekat	=	0	MF dekat	=	0
MF sedang	=	1	MF sedang	=	1
MF jauh	=	0	MF jauh	=	0

Rule Fuzzy

Kiri-depan / Belakang	**Dekat**	**Sedang**	**Jauh**
Dekat	0	0	0
Sedang	0	0	0
Jauh	0	0	0

Defuzzification

$$out = \frac{(0\times0)+(0\times72)+(0\times72)+(0\times-72)+(0\times0)+(0\times-72)+(0\times-72)+(0\times-72)+(0\times-72)}{0+0+0+0+0+0+0+0+0}$$

$$= 0$$

- Jika *input* kiri-depan = 12cm dan belakang = 45cm

Kiri-depan			Belakang		
Input	=	12cm	*Input*	=	45cm
MF dekat	=	$\frac{15-12}{15-10} = \frac{3}{5} = 0{,}6$	MF dekat	=	0
MF sedang	=	$\frac{12-10}{15-10} = \frac{2}{5} = 0{,}4$	MF sedang	=	$\frac{50-45}{50-30} = \frac{5}{20} = 0{,}25$
MF jauh	=	0	MF jauh	=	$\frac{45-30}{50-30} = \frac{15}{20} = 0{,}75$

Rule Fuzzy

Kiri-depan / Belakang	**Dekat**	**Sedang**	**Jauh**
Dekat	0	0	0
Sedang	0,25	0,25	0
Jauh	0,6	0,4	0

Defuzzification

$$out = \frac{(0\times0)+(0{,}25\times72)+(0{,}6\times72)+(0\times-72)+(0{,}25\times0)+(0{,}4\times-72)+(0\times-72)+(0\times-72)+(0\times-72)}{0+0{,}25+0{,}6+0+0{,}25+0{,}4+0+0+0}$$

$$= 21{,}6$$

Dari ketiga pengujian tersebut, dapat lihat perbandingan dari setiap *output* yang dihasilkan. Tabel berikut menguraikan perbandingan hasil pengujian dari ketiga cara yang telah dilakukan :

Tabel 4.1 Hasil Pengujian Logika *Fuzzy*

No	*Input*		*Output*		
	Kiri-Depan (cm)	Belakang (cm)	Simulasi Matlab	Program Codevision AVR	Perhitungan Manual
1	10	27	61,2	61,2	61,2
2	15	30	0	0	0
3	12	45	21,6	21,6	21,6

Dari tabel hasil pengujian logika *Fuzzy* terlihat bahwa *output* dari program yang dibuat menggunakan Codevision AVR bernilai sama dengan simulasi Matlab maupun perhitungan manual. Dengan begitu, program yang telah dibuat menggunakan *software* Codevision AVR tersebut, dapat digunakan dalam tugas akhir ini.

4.1.2 Pengujian Sensor SR-04

Tujuan pengujian untuk sensor SR-04 ini agar mengetahui kemampuan sensor ini dalam mendeteksi jarak halangan yang berada di depan sensor. Pengujian dilakukan sebanyak 25 kali dengan jarak halangan yang berbeda. Jarak yang terukur oleh sensor akan ditampilkan ke LCD dibandingkan dengan jarak yang sebenarnya. Berikut hasil pengujian dari sensor jarak SR-04:

Tabel 4.2 Hasil Pengujian Sensor SR-04

Tampilan di LCD (cm)	Jarak sebenarnya (cm)
4,9	5
5,7	6
6,6	7
8,1	8
8,9	9
10,3	10
11,3	11
12,2	12
13	13
13,9	14
15	15
16,2	16
16,9	17
18	18
18,9	19
19,7	20
24,9	25
29,7	30
35,2	35
40,2	40
45	45
50,1	50
55,1	55
60,2	60

Berdasarkan tabel di atas dapat disimpulkan bahwa kemampuan sensor SR-04 dalam mendeteksi halangan hingga sejauh 60 cm menghasilkan nilai yang hampir sama dengan pengukuran sebenarnya, meskipun dalam pengujian sensor ini sering terjadi *error* atau selisih pada jarak-jarak tertentu.

4.1.3 Pengujian *Optocoupler*

Pengujian ini dilakukan bertujuan untuk mengetahui parameter *output* berupa logika *high* dan *low* pada *optocoupler*. Pengujian dilakukan dengan memberikan tegangan sebesar 5V pada rangkaian, kemudian memutar roda cacah yang terdapat pada *optocoupler*. Roda cacah ini berfungsi agar *output* yang

dihasilkan saat roda cacah diputar berupa tegangan yang berlogika *high* dan *low*. Pada tabel berikut diuraikan hasil pengujian *optocoupler*:

Tabel 4.3 Hasil Pengujian *Optocoupler*

Tegangan *input*	Posisi roda cacah	Tegangan *output*	Logika biner
5Volt	Menghalangi cahaya IR LED	4,9Volt	1
	Meloloskan cahaya IR LED	0,85Volt	0

Dari tabel pengujian diatas terlihat bahwa *optocoupler* yang digunakan menghasilkan tegangan *output* sebesar 4,9Volt berlogika *high* dan 0,85Volt berlogika *low*. Logika *high* dan *low* ini akan menjadi *input* mikrokontroler, sehingga dapat digunakan untuk menghitung panjang ruang parkir yang dilewati oleh *prototype* mobil.

4.1.4 Pengujian PWM Sebagai Pengatur Kecepatan Motor DC

Pengujian PWM ini bertujuan untuk mengetahui pengaruh sinyal PWM terhadap perubahan kecepatan motor DC. Seperti yang dijelaskan pada bab sebelumnya, bahwa pengaturan kecepatan motor DC menggunakan PWM dipengaruhi oleh lebar pulsa *high* dan *low* pada satu periode gelombang yang diberikan, sehingga menghasilkan *duty cycle*. *Duty cycle* ini akan mempengaruhi tegangan yang diberikan ke motor DC sehingga mempengaruhi kecepatan motor. Pada pengujian dilakukan dengan memberikan variasi lebar pulsa *high* dan *low* untuk menghasilakan *duty cycle* yang berbeda. Berikut tabel hasil pengujian motor DC:

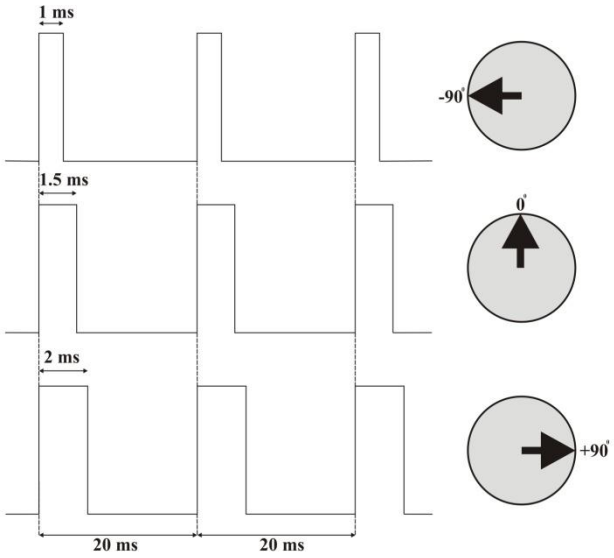
Tabel 4.4 Hasil Pengujian Motor DC

VFull	T*high*	T*low*	Ttotal	DutyCycle	V=DutyCycle×Vfull	V rata-rata
5 V	0	10	10	0	0 V	0,9V
	10	10	20	0,5	2,5 V	3,2V
	20	10	30	0,67	3,35 V	3,7V
	30	10	40	0,75	3,75 V	3,9V
	50	10	60	0,83	4,15 V	4V
	30	15	45	0,67	3,35 V	3,8V

Dari tabel diatas terlihat bahwa semakin besar *duty cycle* yang diberikan, maka tegangan rata-rata motor DC juga semakin besar sehingga kecepatan motor DC juga akan semakin cepat. *Duty cycle* yang diberikan dapat diaatur dengan mengganti nilai T *high* atau T *low* dari lebar pulsa PWM sesuai dengan keinginan.

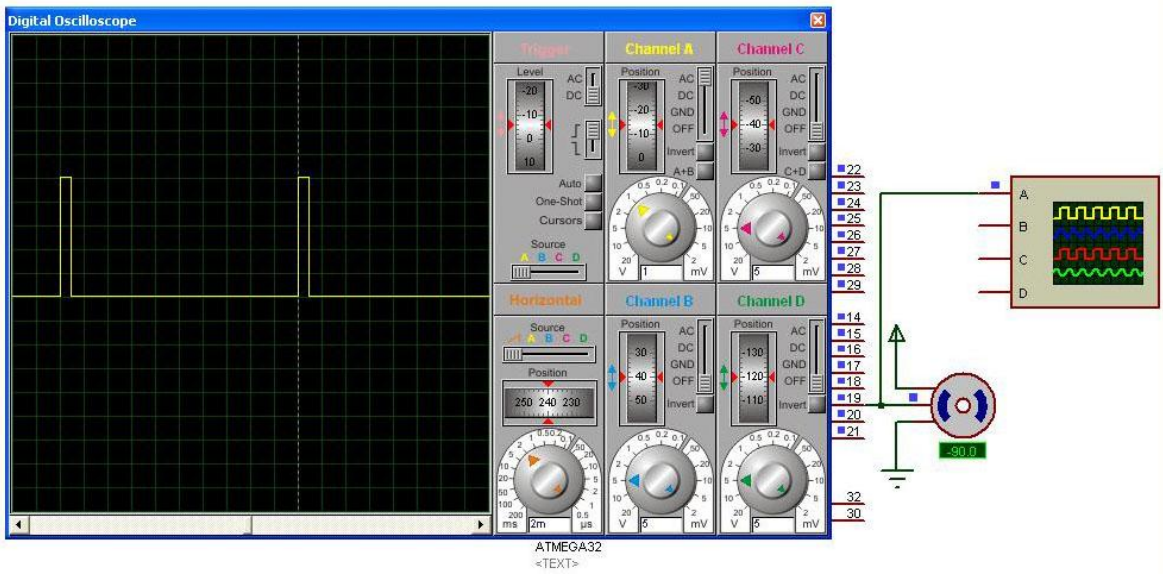
4.1.5 Pengujian PWM Sebagai Pengatur Sudut Motor Servo

Pengujian PWM ini bertujuan untuk mengetahui pengaruh sinyal PWM terhadap perubahan perubahan sudut motor servo. PWM sebagai pengaturan sudut motor servo tidak jauh berbeda dengan pengaturan kecepatan motor DC yaitu mengatur lebar pulsa *high* dan *low* pada satu periode gelombang yang diberikan. Pada pengujian motor servo ini dilakukan dengan membandingkan lebar pulsa referensi dengan hasil simulasi. Pengujian dilakukan sebanyak 3 kali dengan menggunakan software Proteus untuk mensimulasikan hasil dari program yang telah dibuat mengunakan *software* CodeVision AVR untuk mengatur sudut motor servo ini. Berikut hasil dari pengujian motor servo:



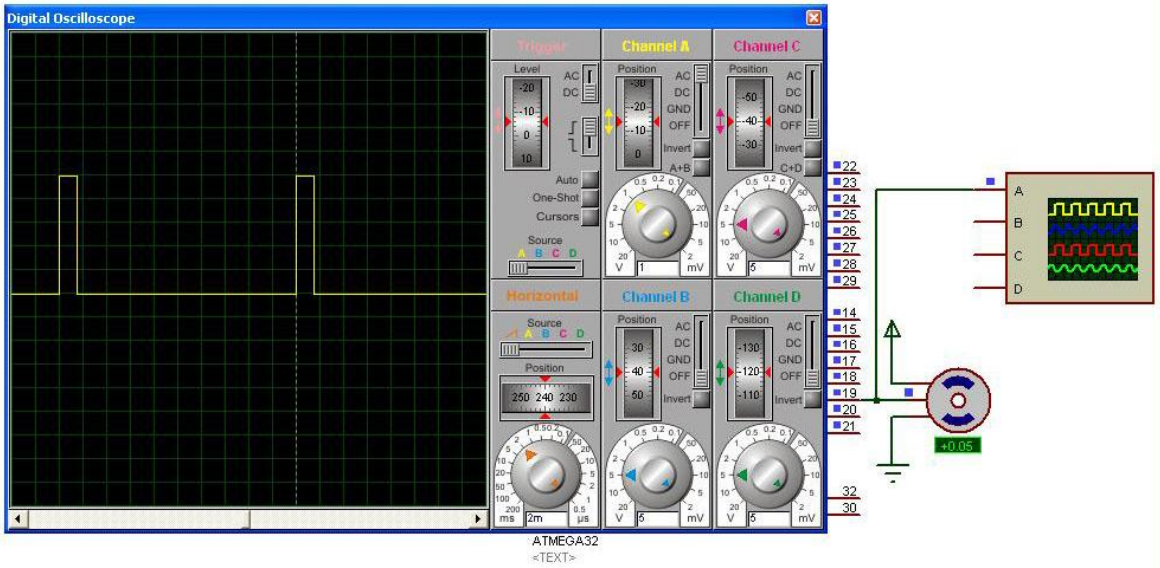
Gambar 4.4 Sinyal Kontrol Motor Servo Sebagai Acuan Dalam Pengujian

Gambar di atas merupakan sinyal kontrol yang digunakan untuk mengubah sudut motor servo. Sinyal di ataslah yang dijadikan acuan untuk menentukan T *high* dan T *low* untuk pengontrolan motor servo ini.



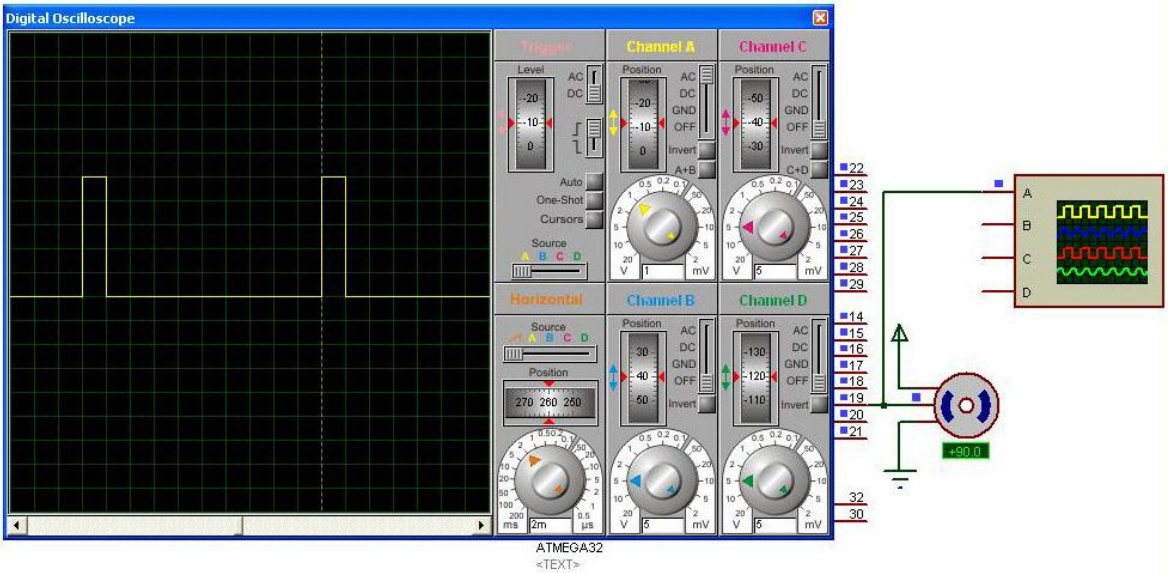
Gambar 4.5 Hasil Pengujian Motor Servo Pertama

Pada gambar terlihat lebar pulsa sebesar 20ms dengan T *high* sebesar 1ms menghasilkan pergerakan motor servo dengan sudut -90.



Gambar 4.6 Hasil Pengujian Motor Servo Kedua

Pada gambar terlihat lebar pulsa sebesar 20ms dengan T *high* sebesar 1,5ms menghasilkan pergerakan motor servo dengan sudut 0.



Gambar 4.7 Hasil Pengujian Motor Servo Ketiga

Pada gambar terlihat lebar pulsa sebesar 20ms dengan T *high* sebesar 2ms menghasilkan pergerakan motor servo dengan sudut +90.

Dari ketiga hasil simulasi diatas terlihat bahwa program yang dibuat untuk pengontrolan motor servo ini dapat digunakan, karena lebar pulsa yang dihasilkan

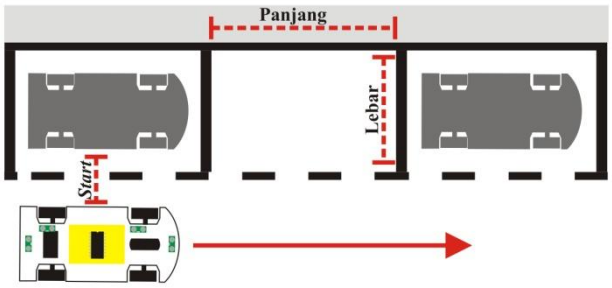
dari program yang dibuat telah sama dengan lebar pulsa yang dijadikan acuan yaitu dengan cara memberikan pulsa *high* selama 1,5 ms dan mengulangnya setiap 20 ms, maka posisi servo akan berada ditengah atau netral (0°). Untuk pulsa 1 ms maka akan bergerak berkebalikan arah jarum jam dengan sudut -90°. Dan pulsa *high* selama 2 ms akan bergerak searah jarum jam sebesar 90° seperti yang dijelaskan pada bab sebelumnya.

4.2 Pengujian Sistem Secara Bertahap

Pengujian sistem secara bertahap ini bertujuan untuk mengetahui kinerja dari sistem yang telah dibuat. Pengujian dibagi menjadi 2 bagian sesuai dengan gambaran sistem yang telah dijelaskan pada bab sebelumnya, yaitu:

4.2.1 Mobil Maju Mencari Ruang (*Space*) Parkir

Pada tahap ini akan diuji sensor SR-04 pada bagian kiri-belakang untuk mendeteksi *space* parkir dan *optocoupler* sebagai penghitung panjang *space* parkir yang akan digunakan. *Optocoupler* akan menghitung jika sensor ultrasonik pada bagian kiri-belakang mendeteksi adanya *space* parkir. *Prototype* mobil akan berhenti jika hasil *counter* dari *optocoupler* bernilai 19 yang artinya panjang *space* parkir tersebut ±70 dan dapat digunakan sebagai *space* parkir.



Gambar 4.8 Gambar Pengujian Mencari *Space* Parkir

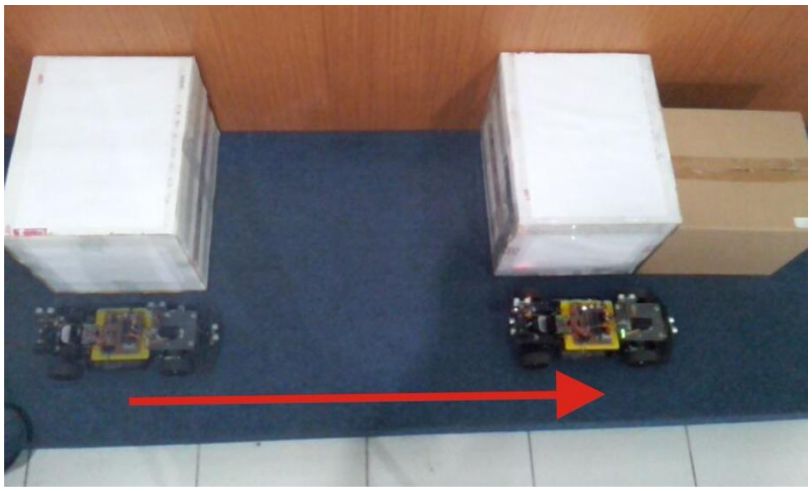
Pada gambar terlihat ”panjang“,“lebar“dan”*start“*. Panjang dan lebar merupakan ukuran *space* parkir, dan *start* merupakan jarak *prototype* mobil dengan mobil yang telah terparkir. Pengujian dilakukan dengan jarak *start* serta panjang dan lebar *space* yang berbeda-beda. Berikut tabel hasil pengujian mobil maju mencari *space* parkir.

Tabel 4.5 Hasil Pengujian Mobil Maju Mencari Ruang (*Space*) Parkir

Jarak Posisi *Start*	**Panjang *Space* Parkir**	**Lebar *Space* Parkir**			
		40 cm		**25 cm**	
		Berhasil	**Gagal**	**Berhasil**	**Gagal**
3 cm	66 cm		✓		✓
	68 cm		✓		✓
	70 cm		✓		✓
	72 cm	✓		✓	
	74 cm	✓		✓	
	76 cm	✓		✓	
	78 cm	✓		✓	
	80 cm	✓		✓	
5 cm	66 cm		✓		✓
	68 cm		✓		✓
	70 cm		✓		✓
	72 cm	✓			✓
	74 cm	✓			✓
	76 cm	✓		✓	
	78 cm	✓		✓	
	80 cm	✓		✓	
7 cm	66 cm		✓		✓
	68 cm		✓		✓
	70 cm	✓		✓	
	72 cm	✓			✓
	74 cm	✓			✓
	76 cm	✓			✓
	78 cm	✓			✓
	80 cm	✓			✓
10 cm	66 cm		✓		✓
	68 cm		✓		✓
	70 cm		✓		✓
	72 cm		✓		✓
	74 cm	✓			✓
	76 cm	✓			✓
	78 cm	✓			✓
	80 cm	✓			✓

15 cm	66 cm		✓		✓
	68 cm		✓		✓
	70 cm		✓		✓
	72 cm	✓			✓
	74 cm	✓			✓
	76 cm	✓			✓
	78 cm	✓			✓
	80 cm	✓			✓

Dari tabel di atas dapat dilihat bahwa proses deteksi *space* parkir dapat bekerja dengan baik apabila panjang *space* yang tersedia besar dari 70 cm dan lebar 40 cm. Baik dalam posisi *start* dekat yaitu 3 cm maupun posisi *start* yang jauh yaitu 15 cm, sensor ultrasonik dan *counter* dapat bekerja menghitung *space* yang tersedia. Namun pada lebar *space* sebesar 25 cm, proses deteksi *space* sering gagal, karena sensor ultrasonik tidak mendeteksi *space* yang cocok untuk digunakan sehingga *counter* tidak menghitung dengan baik. Berikut contoh hasil pengujian dari tahap pertama ini:



Gambar 4.9 Mobil Maju Mencari Ruang (*space*) Parkir

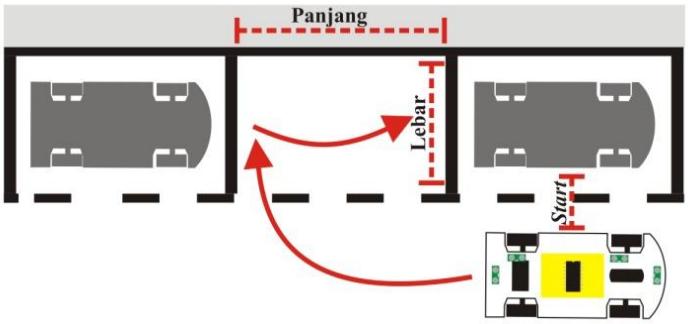


Gambar 4.10 *Sequence* Mobil Maju Mencari Ruang (*space*) Parkir

Pada gambar 4.10 terlihat *prototype* mobil berjalan dan medeteksi *space* parkir. Saat *counter* telah menghitung sebanyak 19 kali maka *prototype* mobil akan berhenti dan bersiap masuk ke tahap selanjutnya yaitu melakukan proses parkir.

4.2.2 Proses Parkir

Pada tahap ini akan diuji kemampuan sistem dalam melakukan proses parkir saja. Proses parkir ini terdiri dari 2 tahapan yaitu mundur melakukan maneuver parkir dan maju di dalam *space*, namun dalam pengujian ini dilakukan secara bersamaan karena kedua tahapan tersebut masuk dalam 1 proses. Pengujian dilakukan dengan mengganti jarak *start* dan bebagai panjang *space* parkir yang akan digunakan. Namun dilihat dari pengujian sebelumnya, lebar *space* sebesar 25 cm tidak dapat dideteksi dengan baik, maka pada pengujian proses parkir ini hanya akan menggunakan lebar *space* sebesar 40 cm.



Gambar 4.11 Gambar Pengujian Proses Parkir

Gambar di atas menunjukkan cara pengujian untuk proses parkir ini yang terdiri dari ukuran *space* dan jarak *start* dengan mobil yang telah terparkir. Berikut tabel hasil pengujian untuk proses parkir.

Tabel 4.6 Hasil Pengujian Proses Parkir

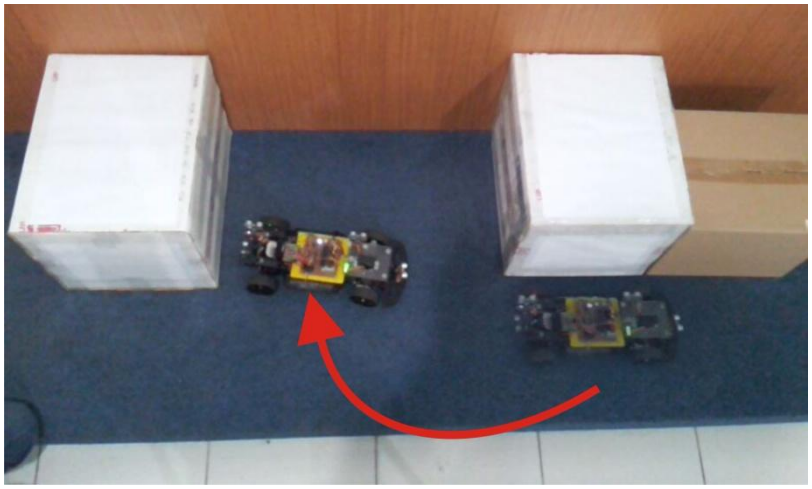
Jarak Posisi *Start*	**Panjang *Space* Parkir**	**Lebar *Space* Parkir**	
		40 cm	
		Berhasil	**Gagal**
3 cm	66 cm		✓
	68 cm		✓
	70 cm		✓
	72 cm	✓	
	74 cm	✓	
	76 cm	✓	
	78 cm	✓	
	80 cm	✓	
5 cm	66 cm		✓
	68 cm	✓	
	70 cm	✓	
	72 cm	✓	
	74 cm		✓
	76 cm	✓	
	78 cm	✓	
	80 cm	✓	
7 cm	66 cm		✓
	68 cm		✓
	70 cm	✓	
	72 cm	✓	
	74 cm	✓	
	76 cm	✓	
	78 cm	✓	
	80 cm	✓	

10 cm	66 cm		✓
	68 cm		✓
	70 cm	✓	
	72 cm	✓	
	74 cm	✓	
	76 cm	✓	
	78 cm	✓	
	80 cm	✓	
15 cm	66 cm		✓
	68 cm		✓
	70 cm		✓
	72 cm		✓
	74 cm		✓
	76 cm		✓
	78 cm		✓
	80 cm		✓

Dari tabel di atas dapat diketahui bahwa proses parkir ini dapat berjalan dengan baik pada jarak *start* atau jarak *prototype* dengan mobil yang telah terparkir sebelumnya kecil dari 10 cm dengan ukuran panjang *space* parkir diantara 72 cm – 80 cm. Berikut penjelasan serta salah satu contoh proses parkir yang berhasil dilakukan.

a. Mobil Mundur Melakukan Manuver Parkir

Pada tahap ini akan diuji kemampuan dari logika *Fuzzy* dalam memroses masukan jarak dari sensor kiri-depan dan belakang. Proses manuver parkir ini akan berhenti saat sensor belakang telah mendeteksi jarak sebesar 5 cm dan bersiap masuk ke tahap ketiga. Berikut gambaran dari hasil pengujian proses maneuver parkir:



Gambar 4.12 Mobil Mundur Melakukan Manuver Masuk Ke Dalam Ruang (*space*) Parkir



Gambar 4.13 *Sequence* Mobil Mundur Melakukan Manuver Masuk Ke Dalam Ruang (*space*) Parkir

b. Mobil Maju Di Dalam Ruang (*Space*) Parkir

Pada tahap ini akan diuji proses *prototype* mobil yang akan maju dalam *space* parkir. Pada tahap ini mobil akan maju dengan *steering* maksimal kearah kiri atau sebesar 72 derajat motor servo hingga sensor depan telah mendeteksi jarak sebesar 8 cm. Berikut gambaran hasil pengujian pada tahap ketiga ini:



Gambar 4.14 Mobil Maju Masuk Ke Dalam Ruang (*space*) Parkir



Gambar 4.15 *Sequence* Mobil Maju Masuk Ke Dalam Ruang (*space*) Parkir

4.3 Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan ini bertujuan untuk mengetahui kemanpuan dari sistem mulai dari mendeteksi *space* parkir hingga proses parkir selesai dilakukan. Pada pengujian kali ini dilakukan sebanyak 20 kali dengan berbagai jarak posisi *start* dengan ukuran *space* yang telah ditentukan. Pada pengujian sebelumnya diketahui bahwa proses deteksi *space* dan proses parkir akan berjalan dengan baik apabila ukuran *space* sebesar panjang 72 cm – 80 cm

dan lebar 40 cm. Maka pada pengujian sistem secara keseluruhan ini akan dilakukan dengan ukuran *space* terkecil yaitu sebesar panjang 72 cm dan lebar 40 cm. Berikut hasil pengujian dari proses secara keseluruan ini.

Tabel 4.7 Hasil Pengujian Proses Secara Keseluruhan

No	**Jarak Posisi *Start***	***Space* Parkir**		**Deteksi *Space***		**Proses Parkir**		**Waktu (second)**
		Panjang	**Lebar**	**Berhasil**	**Gagal**	**Berhasil**	**Gagal**	
1	3 cm	72 cm	40 cm	✓		✓		10.03
2					✓		✓	-
3				✓		✓		10.98
4				✓		✓		11.38
5				✓		✓		10.47
6	5 cm	72 cm	40 cm	✓		✓		10.36
7				✓		✓		10.36
8				✓		✓		10.21
9					✓		✓	-
10				✓			✓	-
11	7 cm	72 cm	40 cm	✓		✓		09.77
12				✓		✓		09.97
13				✓			✓	-
14				✓		✓		10.14
15				✓		✓		11.00
16	10 cm	72 cm	40 cm		✓		✓	-
17				✓		✓		10.68
18				✓		✓		10.26
19				✓		✓		11.26
20				✓		✓		10.53
Total :				17	3	15	5	157.4
Rata-rata :				$\frac{17}{20} \times 100\% = 85\%$		$\frac{15}{20} \times 100\% = 75\%$		$\frac{157.4}{15} = 10.493\ detik$

Pada tabel di atas terlihat bahwa proses parkir secara keseluruhan ini dapat dilakukan dengan baik pada jarak posisi *start* atau jarak *prototype* dengan mobil yang telah terparkir sebelumnya kecil dari 10 cm dan ukuran *space* panjang 72 cm dan lebar 40 cm dengan rata-rata tingkat keberhasilan sebesar 85% untuk deteksi *space* dan 75% untuk proses parkir hingga selesai dilakukan dengan rata-rata waktu selama 10.493 detik. Kegagalan dalam melakukan deteksi maupun proses

parkir ini terjadi karena *error* dari sensor ultrasonik yang digunakan. Meskipun jarak yang terukur dari sensor ultrasonik hampir sama dengan ukuran yang sebenarnya namun pada pengujian sistem secara keseluruhan dimana semua masukan akan diproses secara bersamaan mengakibatkan *error* akibat sensor ini semakin sering terjadi.