

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pengujian Koneksi Serial UART

Pengujian koneksi ini membuktikan bahwa PC dapat dihubungkan dengan *microcontroller* menggunakan komunikasi serial.

##### 4.1.1 Tujuan

Pengujian koneksi ini bertujuan untuk mengetahui perintah yang dikirim di PC dapat diolah oleh *microcontroller*.

##### 4.1.2 Alat Yang Digunakan

1. Processor (PC)
2. Software CodeVision AVR 2.03.4
3. *Minimum System* ATMEGA8535

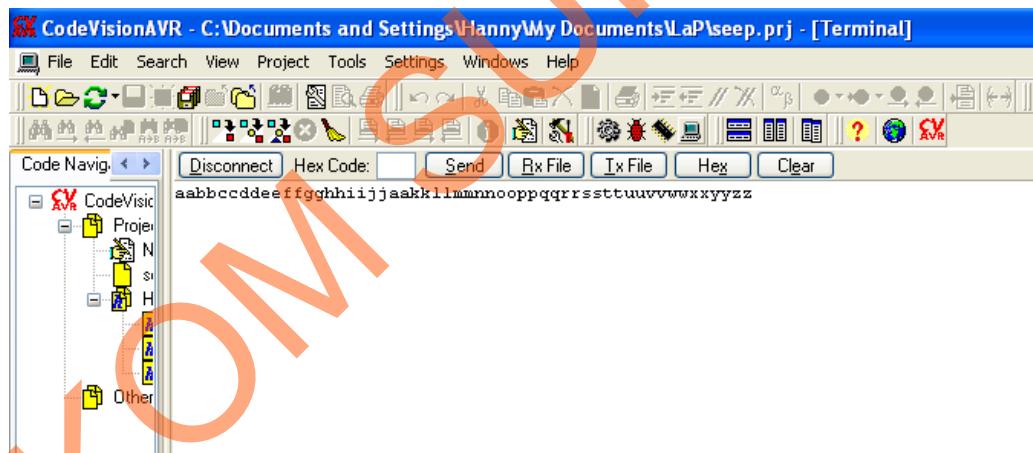
##### 4.1.3 Prosedur Pengujian

1. Menyalakan *Minimum System* ATMEGA8535
2. Siapkan PC dengan program test serial dengan compiler CodeVision AVR.
3. *Download* program tersebut ke ATMEGA8535.

4. Hubungkan melalui kabel serial antara PC dengan *Minimum System ATMEGA8535*.
5. Jalankan Program CodeVision AVR dan pilih menu “Run the terminal”.
6. Ketik perintah yang dapat diolah microcontroller, sesuai dengan program yang sebelumnya telah di-*download*.
7. Perhatikan respon dari terminal tersebut.

#### 4.1.4 Hasil Pengujian Koneksi Serial UART

Koneksi antara PC dan *microcontroller* dapat terkoneksi secara serial dengan baik, hal ini tampak pada gambar 4.1.



Gambar 4.1 Pengujian Komunikasi Serial Sukses

Terlihat pada gambar 4.1 bahwa satu karakter muncul dua kali, hal tersebut dikarenakan satu karakter input dan satu lagi karakter output. Karena program yang di-*download* adalah program untuk mengembalikan karakter yang dikirim (karakter input).

## 4.2 Pengujian Pergerakan *Mobile Robot*

Pengujian ini menggunakan objek dua buah motor DC 12 V dengan driver relay.

### 4.2.1 Tujuan

Tujuan dari pergerakan *mobile robot* adalah untuk mengetahui pergerakan dari *mobile robot*.

### 4.2.2 Alat Yang Digunakan

1. Dua motor DC 12V
2. Driver relay
3. *Minimum System* ATMEGA8353
4. Software CodeVision AVR 2.03.4

### 4.2.3 Prosedur Pengujian

1. Menyalakan *Minimum System* ATMEGA8535
2. Siapkan PC dengan program tes motor dengan CodeVision AVR.
3. *Download* program tersebut ke ATMEGA8535.
4. Beri tegangan 5V dan GND pada input *relay driver*.
5. Perhatikan respon dari motor tersebut

#### 4.2.4 Hasil Pengujian

Hasil dari pengujian ini adalah pergerakan dari motor DC 12V yang dapat dikendalikan sesuai dengan input. Pergerakan dari motor tersebut adalah pergerakan maju dan mundur dari motor tersebut.

Tabel 4.1 Percobaan Dan Hasil Output Motor Dengan Input Serial.

No .	PWM	PD.6 (kiri)	PD.7 (kanan)	M.Kiri	M.kanan
		Input		Output	
1.	0-4	1	1	Berhenti	Berhenti
2.	0-4	0	0	Berhenti	Berhenti
3.	5-37	1	1	Maju pelan sekali	Maju pelan sekali
4.	5-37	0	0	Mundur pelan sekali	Mundur pelan sekali
5.	38-83	1	1	Maju pelan	Maju pelan
6.	38-83	0	0	Mundur pelan	Mundur pelan
7.	84-153	1	1	Maju sedang	Maju sedang
8.	84-153	0	0	Mundur sedang	Mundur sedang
9.	154-255	1	1	Maju cepat	Maju cepat
10.	154-255	0	0	Mundur cepat	Mundur cepat

### 4.3 Pengujian Penentuan nilai *thresholding*

#### 4.3.1 Tujuan

Tujuan dari Pengujian ini adalah mencari nilai thresholding yang sesuai untuk pengambilan data kamera.

#### 4.3.2 Alat yang digunakan

1. Luks meter (pengukur cahaya)
2. Web Cam
3. PC
4. Software Visual C++ 2008 yang telah dilengkapi library OpenCV.

### 4.3.3 Prosedur Pengujian

1. Ukur cahaya ruangan dengan luks meter
2. Lihat hasilnya pada PC
3. Perhatikan dan sesuaikan hasil pada PC dengan hasil pada Luks meter
4. Tentukan nilai thresholding yang sesuai.

### 4.3.4 Hasil Pengujian

Hasil dari pengujian ini berupa satuan cahaya yang sesuai dan dapat digunakan untuk pemrosesan data. Selain itu hasil dari pengujian ini adalah nilai *thresholding* yang digunakan untuk pemrosesan pengolahan citra.

Tabel 4.2 Pengukuran Cahaya dengan hasil greyscale pada PC

No.	Intensitas Cahaya (fluks)	Hasil pada PC (grey)
1.	1,3	Terlalu gelap
2.	2,5	Terlalu gelap
3.	3,0	Terlalu gelap
4.	3,8	Jelas
5.	4,5	Jelas
6.	5,5	Jelas
7.	6,0	Terlalu terang
8.	7,2	Terlalu terang
9.	8,5	Terlalu terang
10.	9,0	Terlalu terang

Didapatkan range yang terbaik untuk melakukan pemrosesan pengolahan data pada citra adalah 3,8 hingga 5,5. Sehingga pada saat pengujian mencari nilai *thresholding* maka *range* intensitas cahaya yang digunakan adalah 3,8 hingga 5,5. Pada pengujian ini penulis menggunakan cahaya pada 4,0 fluks. Sehingga memperoleh hasil sebagai berikut.

Tabel 4.3 Hasil Pengujian Thresholding

No.	Nilai Thresholding	Hasil Pada PC (biner)
1.	2	Terlalu banyak noise
2.	5	Terlalu banyak noise
3.	7	Terlalu banyak noise
4.	10	Terlalu banyak noise
5.	12	Terlalu banyak noise
6.	15	Terlalu banyak noise
7.	17	Terlalu banyak noise
8.	20	Terlalu banyak noise
9.	22	Terlalu banyak noise
10.	25	Terlalu banyak noise
11.	27	Terlalu banyak noise
12.	30	Terlalu banyak noise
14.	32	Terlalu banyak noise
15.	35	Terlalu banyak noise
16.	37	Jelas
17.	40	Jelas
18.	42	Jelas
19.	45	Jelas
20.	47	Jelas
21.	50	Jelas
22.	52	Jelas
23.	55	Jelas
24.	57	Jelas
25.	60	Jelas
26.	62	Jelas
27.	65	Terlalu terang
28.	67	Terlalu terang
29.	70	Terlalu terang
30.	72	Terlalu terang
31.	75	Terlalu terang
32.	77	Terlalu terang
33.	80	Terlalu terang
34.	82	Terlalu terang
35.	85	Terlalu terang
36.	87	Terlalu terang
37.	90	Terlalu terang
38.	92	Terlalu terang

39.	95	Terlalu terang
40.	97	Terlalu terang
41.	100	Terlalu terang
42.	102	Terlalu terang
43.	105	Terlalu terang
44.	107	Terlalu terang
45.	110	Terlalu terang
46.	112	Terlalu terang
47.	115	Terlalu terang
48.	117	Terlalu terang
49.	120	Terlalu terang
50.	122	Terlalu terang
51.	125	Terlalu terang
52.	127	Terlalu terang
53.	130	Terlalu terang
54.	132	Terlalu terang
55.	135	Terlalu terang
56.	137	Terlalu terang
57.	140	Terlalu terang
58.	142	Terlalu terang
59.	145	Terlalu terang
60.	147	Terlalu terang
61.	150	Terlalu terang
62.	152	Terlalu terang
63.	155	Terlalu terang
64.	157	Terlalu terang
65.	160	Terlalu terang
66.	162	Terlalu terang
67.	165	Terlalu terang
68.	167	Terlalu terang
69.	170	Terlalu terang
70.	172	Terlalu terang
71.	175	Terlalu terang
72.	177	Terlalu terang
73.	180	Terlalu terang
74.	182	Terlalu terang
75.	185	Terlalu terang
76.	187	Terlalu terang
77.	190	Terlalu terang
78.	192	Terlalu terang
79.	195	Terlalu terang

80.	197	Terlalu terang
81.	200	Terlalu terang
82.	202	Terlalu terang
83.	205	Terlalu terang
84.	207	Terlalu terang
85.	210	Terlalu terang
86.	212	Terlalu terang
87.	215	Terlalu terang
88.	217	Terlalu terang
89.	220	Terlalu terang
90.	222	Terlalu terang
91.	225	Terlalu terang
92.	227	Terlalu terang
93.	230	Terlalu terang
94.	232	Terlalu terang
95.	235	Terlalu terang
96.	237	Terlalu terang
97.	240	Terlalu terang
98.	242	Terlalu terang
99.	247	Terlalu terang
100.	250	Terlalu terang

Maka terlihat dari hasil diatas, nilai *thresholding* yang tepat adalah diantara 37 dan 62.

#### 4.4 Pengujian *Streaming* di PC

Pengujian ini dilakukan dengan mengambil gambar dari WebCam dan ditampilkan berupa gambar asli dan biner berikut keputusan menghadapi jalan yang di-*capture*.

#### 4.4.1 Tujuan

Tujuan dari pengujian ini adalah mengetahui hasil keputusan dari pengambilan gambar, apakah sesuai dengan program yang disusun sebelumnya dalam menghadapi jalan.

#### 4.4.2 Alat Yang Digunakan

1. PC (*Personal Computer*).
2. WebCam.
3. Software Visual C++ 2008 yang telah dilengkapi library OpenCV.

#### 4.4.3 Prosedur Pengujian

1. Sambungkan WebCam pada PC.
2. Buka *file project* ANYAR pada PC.
3. Klik “Run” pada toolbar menu.
4. Hadapkan kamera pada jalan robot.
5. Perhatikan respon dari command prompt.

#### 4.4.4 Hasil Pengujian

Pengujian ketika dihadapkan pada jalan robot, PC yang memperoleh input dari kamera dapat menetukan arah robot selanjutnya (kanan dan kiri). Selain itu PC memperoleh gambar asli dari kamera lalu gambar tersebut dibinerkan. Gambar-gambar berikut adalah gambar saat kamera mengambil gambar jalan

dengan berbagai kondisi beserta keputusan yang ditampilkan oleh *comand prompt*.



Gambar 4.2 Kondisi Belok Kiri dari gambar asli



Gambar 4.3 Kondisi Belok Kanan dari Gambar Asli



Gambar 4.4 Kondisi Jalan Lurus dari Gambar Asli

Gambar 4.2 , 4.3 dan 4.4 menunjukan bahwa pengambilan gambar secara *live streaming* telah berhasil dilakukan dan siap dilakukan proses selanjutnya.

#### 4.5 Pengujian Sistem Secara Keseluruhan

Pengujian secara keseluruhan dilakukan untuk menguji sistem dari awal. Pengujian secara keseluruhan dilakukan untuk menguji sistem dari awal hingga akhir program dijalankan. Pengujian dilakukan dengan menggerakkan *mobile robot* dengan pengaruh perubahan gambar jalan yang dicapture oleh WebCam.

##### 4.5.1 Tujuan

Pengujian secara keseluruhan ini mempunyai tujuan untuk mengetahui kinerja hardware dan software yang sesuai dengan harapan.

##### 4.5.2 Alat Yang Digunakan

1. PC (*Personal Computer*)
2. WebCam
3. *Minimum System ATMEGA8535*

4. *Relay Driver*

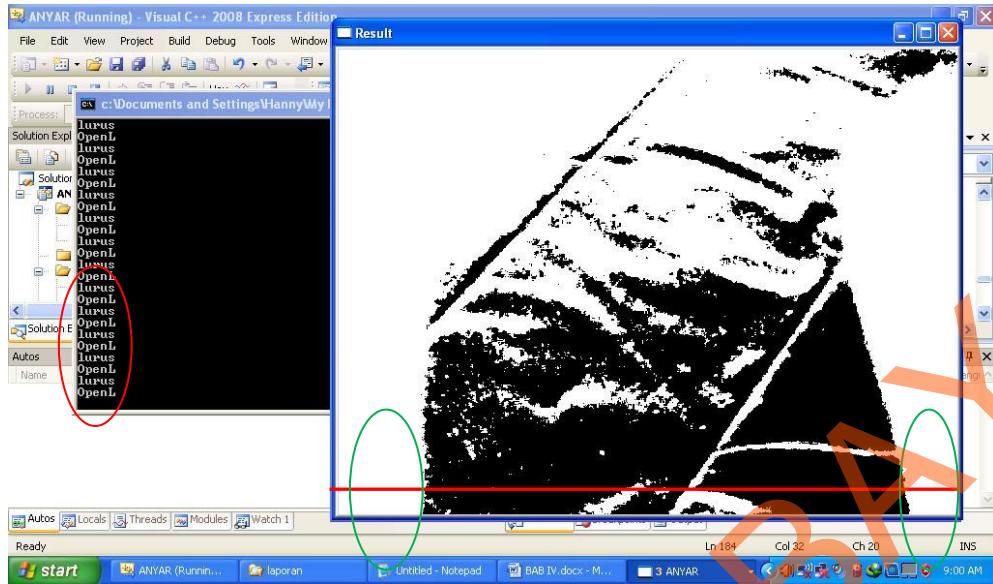
5. Motor DC 12V

#### 4.5.3 Prosedur Pengujian

1. Koneksikan webcam dengan PC.
2. Koneksikan minimum system dengan PC.
3. Buka *project file ANYAR*.
4. Jalankan Project ANYAR.
5. Nyalakan *minimum system*.
6. Perhatikan laju dari *mobile robot*.

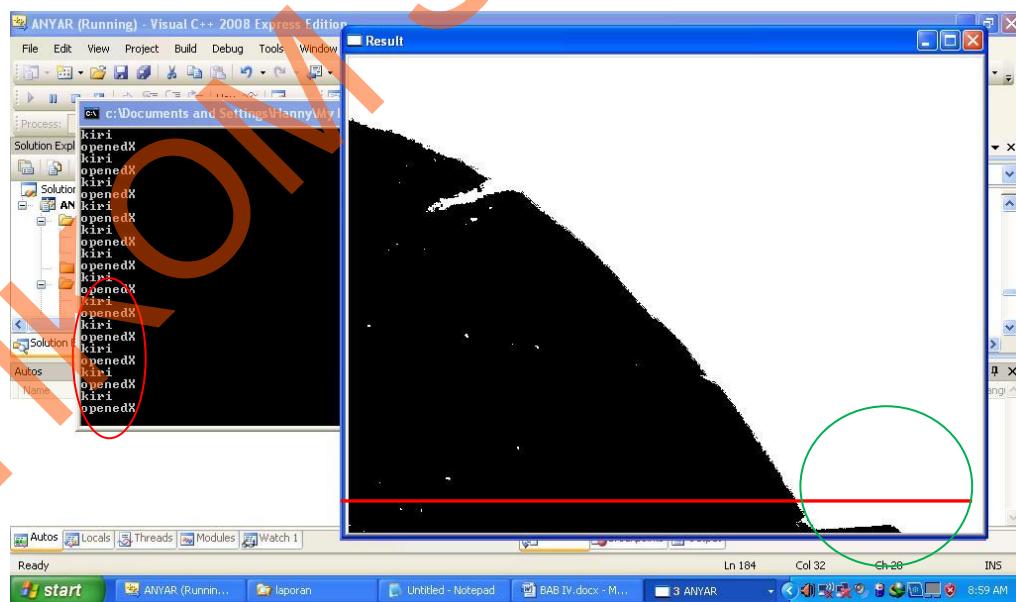
#### 4.5.4 Hasil Pengujian

Pengujian secara keseluruhan berjalan dengan baik karena terlihat dari koneksi serial PC dengan *microcontroller* berjalan baik dan respon dari PC terhadap kamera cukup cepat sehingga proses jalan dari *mobile robot* berjalan dengan lancar. Namun hasil *capture* dari kamera juga tergantung dengan pencahayaan, karena pencahayaan yang berlebihan atau kekurangan mempengaruhi proses dari pengolahan gambar. Sehingga pencahayaan yang cukup mempengaruhi kelancaran jalannya *mobile robot*. Gambar 4.5 , gambar 4.6 dan gambar 4.7 merupakan hasil dari respon *mobile robot*.



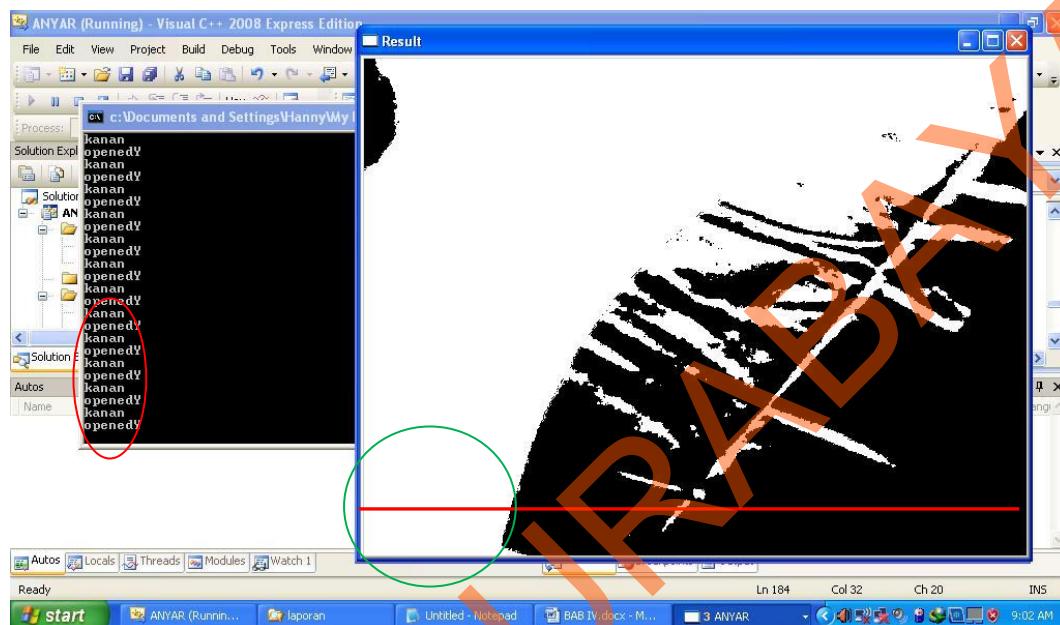
Gambar 4.5 Respone Lurus

Dari gambar 4.5 terlihat pada *Command Prompt* bahwa ketika mendapati gambar dengan ruang sisi kanan dan kiri hampir sama maka respon dari hasil pengolahan adalah lurus.



Gambar 4.6 Respon Kiri

Dari gambar 4.6 terlihat pada *Command Prompt* bahwa ketika mendapatkan gambar dengan ruang kanan sangat lebar sedangkan di sisi kiri tidak ada ruang kosong maka respon dari hasil pengolahan adalah belok kiri.



Gambar 4.7 Respon Kanan

Dari gambar 4.7 terlihat pada *Command Prompt* bahwa ketika mendapatkan gambar dengan ruang sisi kiri kosong sedangkan ruang sisi kanan tidak ada yang kosong maka respon dari hasil pengolahan adalah belok kanan.

Waktu yang digunakan oleh *mobile robot* untuk dapat menempuh satu putaran adalah 2 menit 20 detik dengan PWM 8-bit sebesar 40. Namun kondisi *mobile robot* tersebut mempunyai kelemahan lain selain ketergantungannya dengan cahaya, yaitu *mobile robot* ini hanya menjaga bumper agar tidak keluar dari jalur, namun terkadang bagian belakang dari *mobile robot* masih dapat keluar dari jalur.

Secara keseluruhan proses dari input kamera hingga output motor dapat disimpulkan dalam sebuah tabel dengan mengambil beberapa sampel dari beberapa tikungan pada lintasan tersebut. Tabel 4.2 merupakan hasil dari proses hubungan kamera dengan keadaan motor.

Tabel 4.4 Hubungan Data Kamera dengan keadaan motor

No	Kondisi	Pixel kanan	Pixel kiri	Motor kanan(PWM)	Motor kiri (PWM)
1	Kanan	<120	>120	Mundur (40)	Maju (40)
2	Kiri	<120	>120	Maju (40)	Mundur (40)
3	Lurus	$\pm 120$	$\pm 120$	Maju (40)	Maju (40)
4	Kiri	<120	>120	Maju (40)	Mundur (40)
5	kanan	<120	>120	Mundur (40)	Maju (40)

Dari hasil tabel tersebut terlihat bahwa 120 pixel untuk kanan dan kiri menjadi acuan dari gerak motor untuk membuat *mobile robot* tetap berada di lintasan.