

BAB 4

HASIL DAN PEMBAHASAN

Tahap pengujian sistem dilakukan dengan tujuan adalah untuk mengetahui hasil dari perancangan yang telah dibuat pada Bab 3. Pengujian sistem ini terdiri dari beberapa tahapan, dimulai dari pengujian terhadap tiap-tiap bagian pendukung sistem hingga pengujian sistem secara keseluruhan. Dari hasil pengujian maka dapat dianalisa kinerja-kinerja dari tiap-tiap bagian sistem yang saling berinteraksi sehingga terbentuklah sistem pintu air otomatis pengendali banjir. Pengujian terhadap keseluruhan sistem berguna untuk mengetahui bagaimana kinerja dan tingkat keberhasilan dari sistem tersebut. Dan hasil dari perancangan dan pembuatan tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1. Hasil Akhir Pembuatan Sistem Pengaturan Pintu Air Otomatis

4.1. Pengujian Rangkaian Sistem Minimum Mikrokontroler

Pada pengujian ini dilakukan program penyalan LED yang bertujuan untuk mengetahui apakah pin-pin *input* atau *output* (I/O) pada *port* berfungsi dengan baik atau tidak. Dengan *listing* program LED akan menyala sesuai dengan *input* yang diberikan, maka nantinya dapat dipastikan bahwa *port* I/O pada mikrokontroler sudah bekerja dengan baik atau tidak. Berikut ini adalah program yang akan digunakan untuk menyalakan LED sesuai input yang diberikan.

```

#include <mega8535.h>
#include <delay.h>

void inisialisasi_port()
{
  DDRA=0b11111111;
  DDRB=0b11111111;
  DDRB=0b11111111;
  DDRD=0b11111111;
}
void main(void)
{
  inisialisasi_port();

  while(1)
  {
    PORTA=0b11111111;
    PORTB=0b11111111;
    PORTC=0b11111111;
    PORTD=0b11111111;
    Delay_ms(200);
    PORTA=0b00000000;
    PORTB=0b00000000;
    PORTC=0b00000000;
    PORTD=0b00000000;
    Delay_ms(200);
  };
}

```

Program pengujian di atas menjelaskan bahwa saat inisialisasi PORTA, PORTB, PORTC, dan PORTD telah di set sebagai *output*. dengan kondisi awal semua pin adalah *high* (1) yang berarti keadaan awal LED akan hidup semua dan kemudian diberi *delay* 200 ms maka semua pin dalam keadaan *low* (0) yang berarti LED akan mati semua. Maka dengan hasil seperti ini dapat dikatakan bahwa rangkaian sistem minimum mikrokontroler tersebut dalam kondisi baik. Pengujian dapat pula dilakukan dengan menghubungkan *multimeter* pada jalur yang ada pada sistem minimum Atmega 8535.

4.2. Pengujian Internal ADC

Pengujian internal ADC dilakukan dengan melihat keluaran ADC yang diberikan oleh sensor untuk mengetahui nilai tegangan yang masuk pada mikrokontroler untuk setiap ketinggian air yang berbeda-beda. Tegangan keluaran dari sensor merupakan acuan untuk menentukan seberapa besar nilai dari bacaan

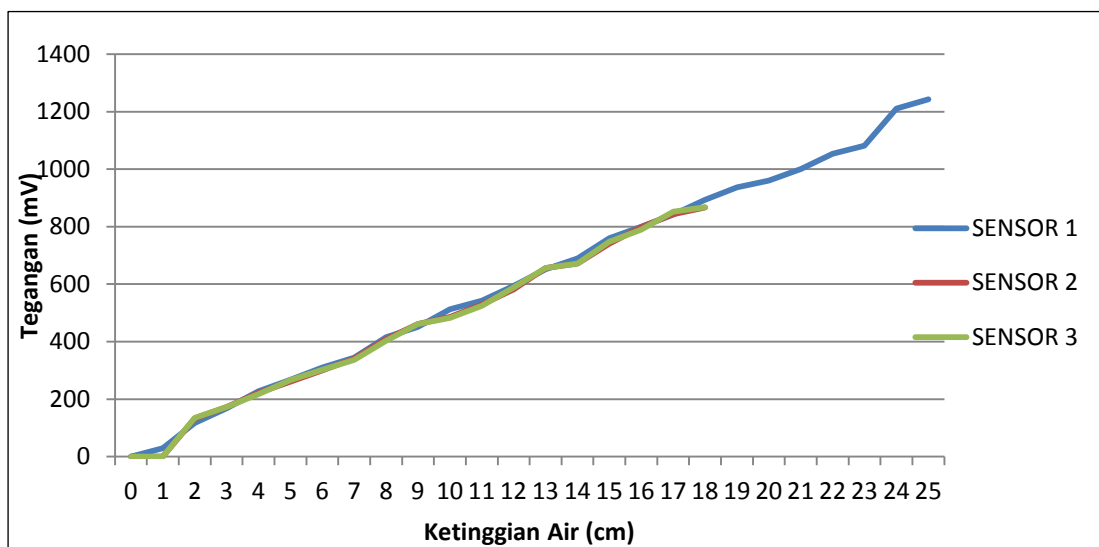
sensor. *Output* tegangan dari sensor yang berupa sinyal analog kemudian dikonversi menjadi sinyal digital menggunakan ADC (*Analog to Digital Converter*) yang terdapat pada mikrokontroler. Setelah data sudah berupa sinyal digital maka dengan menggunakan bahasa pemrograman data dari sensor tersebut dikalibrasi berdasarkan ketinggian air yang sesungguhnya dan kemudian ditampilkan pada LCD yang telah terhubung dengan mikrokontroler. Pengujian ini perlu dilakukan untuk memastikan sensor dapat bekerja dengan baik. Hasil dari pengujian keluaran ADC dari sensor dapat dilihat pada Tabel 4.1.

Tabel 4.1. Hasil Pengukuran *Output* ADC Sensor

KETINGGIAN AIR (cm)	OUTPUT ADC		
	SENSOR 1 (mV)	SENSOR 2 (mV)	SENSOR 3 (mV)
(1)	(2)	(3)	(4)
0	0	0	0
1	29	0	0
2	117	133	135
3	168	173	173
4	228	222	218
5	268	261	267
6	310	300	302
7	344	341	337
8	416	409	403
9	451	461	461
10	512	486	483
11	543	529	525
12	594	582	588
13	652	656	656
14	690	672	672
15	760	741	748
16	800	799	790
17	842	843	852
18	893	867	868

(1)	(2)	(3)	(4)
19	937	-	-
20	960	-	-
21	1000	-	-
22	1054	-	-
23	1081	-	-
24	1211	-	-
25	1243	-	-

Tabel 4.1 merupakan *output* dari 3 buah sensor ketinggian air yang digunakan pada penelitian ini. Dari Tabel 4.1 didapat berbagai macam nilai tegangan dari *output* sensor berdasarkan ketinggian air. *Output* dari ketiga sensor ini hampir sama hanya memiliki sedikit perbedaan, hal ini disebabkan karena kesalahan pembacaan dan juga kesalahan dari peralatan pengujian. Dari hasil pengujian ini dapat dilihat bahwa semakin tinggi air pada bak penampungan maka *output* tegangan dari sensor juga akan semakin besar. Untuk melihat lebih jelas dari *output* sensor berdasarkan ketinggian air pada dilihat Gambar 4.2 yang berisi grafik pengukuran ADC sensor.



Gambar 4.2. Grafik Hasil Pengukuran ADC Sensor

Grafik ADC sensor pada Gambar 4.2 merupakan suatu gambaran karakteristik dari sensor ketinggian air yang digunakan. Berdasarkan Gambar 4.2 tegangan dan ketinggian air berbanding searah, apabila ketinggian air naik maka *output* tegangan dari sensor pun jadi ikut naik. Dari grafik diatas menunjukkan bahwa *output* sensor linear terhadap ketinggian air. Selain itu dapat dilihat dengan jelas bahwa *output* tegangan dari ketiga sensor hampir sama, hanya memiliki sedikit perbedaan.

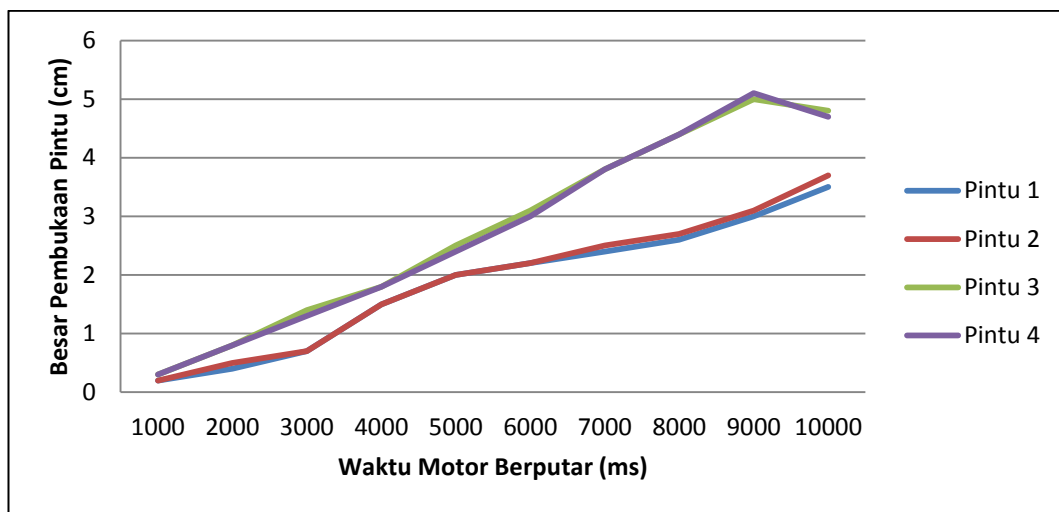
4.3. Pengujian Tinggi Pembukaan Pintu

Pengujian ini dilakukan untuk mengetahui seberapa besar pembukaan pintu ketika motor dihidupkan berdasarkan rentan waktu tertentu. Cara pengujiannya yaitu dengan cara motor pada pintu air dihidupkan sehingga pintu air pada bendungan akan membuka. Kemudian besar pembukaan dihitung berdasarkan berdasarkan lama waktu motor berputar. Pengujian ketinggian pintu ini dilakukan pada keempat buat pintu yang ada, pengujian ini dimaksudkan untuk menentukan seberapa tinggi pintu air naik dari posisi awal berdasarkan waktu motor berputar. Hasil dari pengujian pintu air dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil Pengujian Pembukaan Pintu Berdasarkan Waktu Motor Berputar

Waktu Motor Berputar (ms)	TINGGI PEMBUKAAN PINTU			
	Pintu 1 (cm)	Pintu 2 (cm)	Pintu 3 (cm)	Pintu 4 (cm)
0	0	0	0	0
1000	0,2	0,2	0,3	0,3
2000	0,4	0,5	0,8	0,8
3000	0,7	0,7	1,4	1,3
4000	1,5	1,5	1,8	1,8
5000	2	2	2,5	2,4
6000	2,2	2,2	3,1	3
7000	2,4	2,5	3,8	3,8
8000	2,6	2,7	4,4	4,4
9000	3	3,1	5	5,1
10000	3,5	3,7	4,8	4,7

Pada pengujian ini di dapatkan bahwa semakin sedikit waktu motor berputar maka semakin kecil pintu air itu naik dan apabila semakin banyak waktu motor berputar maka semakin besar pula pintu air itu akan naik. Besar hasil pengujian nilai dari pembukaan suatu pintu berbeda-beda antara pintu1, pintu2, pintu 3 dan pintu 4. Untuk melihat perbedaannya secara jelas dapat di lihat pada Gambar 4.3.



Gambar 4.3. Grafik Hasil Pengujian Besar Pembukaan Pintu

Pada grafik pembukaan pintu pada Gambar 4.3 terlihat jelas bahwa terdapat perbedaan pembukaan pintu air. Antara pintu 1 dengan pintu 2 hanya sedikit perbedaan, hal ini disebabkan karena konstruksi dari kedua pintu ini sama. Sedangkan pintu 3 dan 4 memiliki konstruksi pintu yang berbeda dan lebih kecil, ini dikarenakan antara bendungan utama dengan bendungan pembuangan (bendungan 1 dan 2) memiliki ukuran yang berbeda. Dari grafik juga dapat dilihat waktu motor berputar untuk membuka pintu air berbanding searah dengan besar pembukaan pintu, semakin lama motor berputar untuk membuka maka pembukaan pintu semakin besar.

4.4. Pengujian Debit Air

Pengujian debit air pada bendungan dilakukan untuk mempermudah alat pengontrol pintu air otomatis ini untuk menentukan besar pembukaan pintu dan waktu pintu menutup kembali pada saat air melebihi dari *set point*. Pengujian

debit air dibagi menjadi 2 cara pengujian. Cara pengujian yang pertama dilakukan dengan cara kedua pintu pada bendungan utama membuka dan cara pengujian yang kedua salah satu pintu saja yang terbuka. Besar pembukaan pintu dibagi menjadi 3 pembukaan, yaitu 0,5 cm (pembukaan kecil), 1,5 cm (pembukaan sedang) dan 2,5 cm (pembukaan besar). Sedangkan untuk ketinggian air pada pengujian ini digunakan ketinggian 25, 20, 15 dan 10 cm. Berikutnya dihitung volume air pada bendungan menggunakan Persamaan 2.2, dimana diketahui panjang bendungan 50 cm dan lebar bendungan sebesar 30 cm. Untuk menghitung debit air menggunakan Persamaan 2.3. Tabel 4.3 merupakan hasil pengujian debit air pada bendungan utama dengan kedua pintu terbuka.

Tabel 4.3. Pengujian Debit Air pada Bendungan Utama dengan Kedua Pintu Air Terbuka

Tinggi Air (cm)	Luas Bendungan (cm²)	Volume Air (cm³)	Tinggi Pembukaan Pintu Air (cm)	Waktu Pintu Terbuka Hingga Volume Air = 0 (s)	Debit Air (cm³/s)
25	1500	37500	0,5	230	163
25	1500	37500	1,5	133	281
25	1500	37500	2,5	85	438
20	1500	30000	0,5	152	161
20	1500	30000	1,5	108	277
20	1500	30000	2,5	69	431
15	1500	22500	0,5	128	159
15	1500	22500	1,5	82	273
15	1500	22500	2,5	53	424
10	1500	15000	0,5	96	155
10	1500	15000	1,5	55	269
10	1500	15000	2,5	36	411
5	1500	7500	0,5	53	140
5	1500	7500	1,5	29	257
5	1500	7500	2,5	19	401

Tabel 4.3 merupakan hasil pengujian debit aliran air pada bendungan utama dengan kedua pintunya terbuka. Pada tabel dapat dilihat tiap satu macam volume air menggunakan 3 macam pembukaan hal ini dilakukan untuk menentukan seberapa besar laju debit air yang mengalir pada saat pintu dibuka. Dari pengujian debit air yang keluar cukup besar karena kedua pintu terbuka, dapat dilihat apabila pintu air semakin besar membuka maka debit aliran air akan semakin besar pula dan ketinggian air juga berpengaruh terhadap debit aliran air karena adanya gaya gravitasi sehingga ada data tekan dari air. Sedangkan untuk hasil pengujian dengan salah satu pintu yang yang terbuka dapat dilihat pada Tabel 4.4.

Tabel 4.4. Pengujian Debit Air pada Bendungan Utama dengan Salah Satu Pintu Air Terbuka

Tinggi Air (cm)	Volume Air (cm²)	Tinggi Pembukaan Pintu Air (cm)	Waktu Pintu Terbuka Hingga Volume Air = 0 (s)	Debit Air (cm³/s)
25	37500	0,5	460	81
25	37500	1,5	266	141
25	37500	2,5	170	220
20	30000	0,5	304	98
20	30000	1,5	216	139
20	30000	2,5	138	217
15	22500	0,5	256	88
15	22500	1,5	164	137
15	22500	2,5	106	212
10	15000	0,5	192	78
10	15000	1,5	110	136
10	15000	2,5	72	208
5	7500	0,5	53	140
5	7500	1,5	29	257
5	7500	2,5	19	401

Tabel 4.4 merupakan hasil pengujian debit aliran air pada bendungan utama dengan salah satu pintunya terbuka. Cara pengujian hampir sama dengan pengujian debit air yang pertama bedanya hanya pada pengujian pertama kedua pintu terbuka sedangkan pada pengujian kedua ini hanya salah satu pintu yang terbuka. Pada hasil pengujian debit yang keluar lebih kecil dibandingkan ketika kedua pintu pada bendungan utama terbuka. Dari hasil pengujian debit air ini maka untuk selanjutnya kita dapat menentukan seberapa lama pintu membuka untuk membuang air pada saat air pada bendungan utama melebihi *set point* ketinggian air yang telah ditentukan. Untuk menghitung seberapa lama pintu air membuka dapat dilihat pada Persamaan 2.5.

Dari Persamaan 2.5 bisa dilakukan perhitungan untuk menentukan seberapa lama pintu membuka ketika air pada bendungan utama melebihi *set point*.

Tabel 4.5. Hasil Perhitungan Lama Waktu Pintu Air Membuka

Volume Air (cm³)	Debit Air (cm³/s)	Waktu Pintu Air Membuka (s)
(1)	(2)	(3)
1500	81	18,51
3000	81	37,03
4500	81	55,55
6000	81	74,07
7500	81	92,59
9000	141	63,82
10500	141	74,46
1200	141	85,10
13500	141	95,74
15000	141	106,38
16500	220	75
18000	220	81,81
19500	220	88,63

(1)	(2)	(3)
21000	220	95,45
22500	220	102,27
24000	220	109,09
25500	220	115,90
27000	220	122,72
28500	220	129,54
30000	220	136,36

Perhitungan pada Tabel 4.5 merupakan hasil perhitungan untuk lama waktu pintu air membuka. Volume air didapat dari luas bendungan utama, yaitu: 50 x 30 cm dengan ketinggian air dari ketinggian 1 cm sampai dengan 20 cm, sedangkan laju debit air didapat dari hasil pengujian sebelumnya. Dari volume dan debit air maka didapat berapa lama waktu membuka dengan menggunakan Persamaan 2.5.

3.5. Pengujian Pintu Air Otomatis untuk Membuang Air Ketika Ketinggian Air Melebihi Set Point

Seberapa besar pintu membuka dan berapa lama pintu akan menutup kembali dapat ditentukan dari hasil pengujian debit air, karena dari debit air yang mengalir pada bendungan dapat dihitung berapa waktu yang dibutuhkan untuk mengurangi air pada bendungan apabila air pada bendungan melebihi *set point*. Pengujian pintu air otomatis ini untuk pembukaan pintu dibagi menjadi 3 golongan, yaitu:

1. Pembukaan pintu 0,5 cm untuk *error* 1 – 5 cm.
2. Pembukaan pintu 1,5 cm untuk *error* 6 - 10 cm.
3. Pembukaan pintu 2,5 cm untuk *error* lebih dari 10 cm.

Setelah pintu terbuka dan dari pengujian debit bisa diperkirakan berapa waktu yang dibutuhkan untuk sistem kembali ke menutup pintu air agar air pada bendungan selalu berada pada posisi *set point*. Pengujian ini juga dilakukan pada

set point 5, 10 dan 15 cm. Hasil pengujian pintu air otomatis dengan *set point* 5 cm dapat dilihat pada Tabel 4.6.

Tabel 4.6. Hasil Pengujian Pintu Air Otomatis dengan *Set Point* 5 cm

Ketinggian Air Awal (cm)	Lama Pintu Membuka (s)	Ketinggian Air Setelah Pintu Tertutup (cm)	Error (%)
6	9	5	0
7	19	4	20
8	29	5	0
9	38	5	0
10	48	6	20
11	32	5	0
12	38	4	0
13	43	5	0
14	49	5	0
15	54	5	0
16	37	4	20
17	41	6	20
18	44	5	0
19	47	6	20
20	51	5	0
Error rata – rata			8

Hasil pengujian air otomatis dengan *set point* 5 cm pada Tabel 4.6, waktu tunggu yang diberikan berdasarkan Persamaan 2.5 mendapatkan hasil yang cukup baik dengan tepatnya pintu menutup sehingga pintu tidak mengalami perulangan dalam membuka dan menutup pintu air pada bendungan utama. Pada *set point* 5 cm memiliki *error* rata-rata sebesar 8%.

Tabel 4.7. Hasil Pengujian Pintu Air Otomatis dengan *Set Point* 10 cm

Ketinggian Air Awal (cm)	Lama Pintu Membuka (s)	Ketinggian Air Setelah Pintu Tertutup (cm)	Error (%)
11	11	10	0
12	21	10	0
13	30	11	10
14	40	10	0
15	51	11	10
16	34	9	10
17	42	10	0
18	45	10	0
19	52	10	0
20	56	11	10
Error rata – rata			4

Hasil pengujian pintu air otomatis dengan *set point* 10 cm pada Tabel 4.7. Hasil yang didapatkan sesuai dengan yang diharapkan walaupun terdapat sedikit data yang kurang tepat akan tetapi karena pada sistem ini diberikan toleransi 1 cm sehingga bisa dikatakan sistem berjalan dengan baik. Pada *set point* 10 *error* rata-ratanya sama dengan pengujian sebelumnya yaitu 4%.

Tabel 4.8. Hasil Pengujian Pintu Air Otomatis dengan *Set Point* 15 cm

Ketinggian Air Awal (cm)	Lama Pintu Membuka (s)	Ketinggian Air Setelah Pintu Tertutup (cm)	Error (%)
(1)	(2)	(3)	(4)
16	13	15	0
17	24	15	0
18	35	14	6,6

(1)	(2)	(3)	(4)
19	47	16	6,6
20	58	13	13,3
Error rata – rata			5,3

Hasil pengujian pintu air otomatis dengan *set point* 15 cm pada Tabel 4.8. Ketepatan pintu menutup pada pintu air sudah cukup baik dan dapat dikatakan percobaan berjalan dengan baik walaupun pada pintu air masih terdapat kebocoran ketika pintu ditutup. Untuk *set point* 5 cm *error* rata-ratanya lumayan kecil yaitu 5,3%.

Dari hasil pengujian pintu air otomatis sistem pintu air otomatis sebagai pengendali banjir berjalan dengan baik dengan *error* rata-rata 8% pada *set point* 5, pada *set point* 10 cm memiliki *error* rata-rata sebesar 4%, sedangkan pada *set point* 15 cm *error*-nya lebih kecil yaitu 5,3%. Dari hasil ini menunjukkan bahwa hasil pengujian sudah cukup baik.

BAB 5

PENUTUP

5.1 Kesimpulan

1. Pembuatan *prototype* sistem pengaturan pintu air otomatis sebagai pengendali banjir dapat membuka pintu air sesuai dengan *set point* yang telah ditentukan.
2. Pemilihan pembukaan pintu air pada bendungan utama dapat berjalan dengan baik dan dapat menentukan pintu mana yang harus dibuka terlebih dahulu berdasarkan *set point* dan pembacaan sensor pada bendungan utama dan 2 buah bendungan pembuangan.
3. Pembukaan pintu air otomatis ketika terjadi kenaikan ketinggian air berjalan dengan baik pada *set point* 5 cm setelah pintu membuka dan pintu air dapat menutup kembali pada saat air di ketinggian 5 cm, begitu pula pada *set point* 10 dan 15 cm pintu dapat menutup kembali pada saat ketinggian yang diinginkan walaupun pada beberapa pengujian ketinggian air masih berada pada ketinggian 4 dan 6 cm akan tetapi masih masuk dalam toleransi 1 cm.
4. Hasil pengujian keseluruhan diperoleh *error* rata-rata sebesar 8% pada saat *set point* 5 cm, 4% pada saat *set point* 10 cm dan saat *set point* 10 cm *error* rata-ratanya sebesar 5,3%.

5.2 Saran

1. Faktor tekanan air sangat berpengaruh pada pintu air pada saat membuka dan menutup, sehingga perlu diperhatikan agar tingkat kegagalan dalam pengendalian air menjadi kecil.
2. Perbandingan ukuran pintu terhadap ukuran bendungan harus tepat agar mudah dalam mengendalikan pembuangan air pada bendungan.

DAFTAR PUSTAKA

- [1] Utomo, Safrudin Budi. 2012. *Prototipe pintu bendungan otomatis berbasis Mikrokontroler atmega 16*. Yogyakarta: Universitas Negeri Yogyakarta.
- [2] Syahreza, Saumi. 2009. *Rancang Bangun sensor Ketinggian Air (Water Level) Menggunakan Transduser Ultra Sonik Berbasis Mikrokontroler MCS51*. Aceh: Jurnal Rekayasa Elektrika.
- [3] Indriyanto, Yogi. 2007. *Rancang Bangun Pintu Otomatis dengan Menggunakan Sensor Passive Infra Red Menggunakan Sensor Passive Infra Red KC7783R dan Mikrokontroler AT89S51*. Semarang: Jurusan Fisika Universitas Diponegoro.
- [4] Heryanto, M. Ary. 2008. *Pemrograman Bahasa C Mikrokontroler ATmega8535*. Yogyakarta: ANDI.
- [5] Winoto, Ardi. 2008. *Mikrokontroler AVR Atmega 8/32/16/8535 dan Pemrogramannya dengan bahasa C Pada WinAVR*. Jakarta: Informatika.
- [6] Rafiuddin, Rahmat. 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535*. Yogyakarta: ANDI.
- [7] Zuhail. 1988. *Dasar Teknik Tenaga Listrik dan Elektronika Daya*. Jakarta: Gramedia.
- [8] Maryanto, Hendra. 2010. *Pembuatan Prototipe Pintu Otomatis Satu Arah Berbasis Mikrokontroler Atmega 8535 Menggunakan Double IR*. Surakarta: Universitas Sebelas Maret.
- [9] Pratiwi, Nuryanti. 2013. *Pengaturan Flow Debit Air pada Kran Otomatis Menggunakan Sensor Infrared*. Bandung: UNIKOM.

LAMPIRAN

LAMPIRAN 1

LISTING PROGRAM PENGONTROLAN PINTU AIR OTOMATIS

```
#include <mega8535.h>
#include <delay.h>
#include <lcd.h>
#include <stdio.h>
#define motor1 OCR1A
#define motor2 OCR1B
#define motor3 OCR1C
#define motor4 OCR1D

#define m1a PORTB.0
#define m1b PORTB.1
#define en1 PORTB.4
#define m2a PORTB.2
#define m2b PORTB.3
#define en2 PORTB.5
#define m3a PORTD.0
#define m3b PORTD.1
#define en3 PORTD.4
#define m4a PORTD.2
#define m4b PORTD.3
#define en4 PORTD.5

#define ADC_VREF_TYPE 0x40
char lcd_buff[33];
#asm
.equ __lcd_port=0x15 ; //PORTC
#endasm

void buka1()
{m1a=1;m1b=0;en1=1;}

void buka2()
{m2a=0;m2b=1;en2=1;}

void buka3()
{m3a=0;m3b=1;en3=1;}

void buka4()
{m4a=1;m4b=0;en4=1;}
```

```
void tutup1()
{m1a=0;m1b=1;en1=1;}
```

```
void tutup2()
{m2a=1;m2b=0;en2=1;}
```

```
void tutup3()
{m3a=1;m3b=0;en3=1;}
```

```
void tutup4()
{m4a=0;m4b=1;en4=1;}
```

```
void stop1()
{en1=0;}
```

```
void stop2()
{en2=0;}
```

```
void stop3()
{en3=0;}
```

```
void stop4()
{en4=0;}
```

```
//tombol
```

```
#define sw_ok PINB.6
```

```
#define sw_cancel PINB.7
```

```
#define sw_down PIND.6
```

```
#define sw_up PIND.7
```

```
//====Variabel di eeprom=====
```

```
eeprom int Set_point1=0;
```

```
eeprom int Set_point2=0;
```

```
eeprom int Set_point3=0;
```

```
int pb,tunggu,tunggu1,tunggu2,tunggu3,error,error1,sensor1,sensor2,sensor3;
```

```
void setting_awal()
```

```
{
```

```
pin1:
```

```
lcd_clear();
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("SETTING PINTU 1");
delay_ms(150);
if(sw_up==0){buka1();delay_ms(100);stop1();}
if(sw_down==0){tutup1();delay_ms(100);stop1();}
if(sw_ok==0){delay_ms(150);goto pin2;}
else goto pin1;
```

```
pin2:
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("SETTING PINTU 2");
delay_ms(150);
if(sw_up==0){buka2();delay_ms(100);stop2();}
if(sw_down==0){tutup2();delay_ms(100);stop2();}
if(sw_ok==0){delay_ms(150);goto pin3;}
if(sw_cancel==0){delay_ms(150);goto pin1;}
else goto pin2;
```

```
pin3:
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("SETTING PINTU 3");
delay_ms(150);
if(sw_up==0){buka3();delay_ms(100);stop3();}
if(sw_down==0){tutup3();delay_ms(100);stop3();}
if(sw_ok==0){delay_ms(150);goto pin4;}
if(sw_cancel==0){delay_ms(150);goto pin2;}
else goto pin3;
```

```
pin4:
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("SETTING PINTU 4");
delay_ms(150);
if(sw_up==0){buka4();delay_ms(100);stop4();}
if(sw_down==0){tutup4();delay_ms(100);stop4();}
if(sw_ok==0){delay_ms(150);goto set1;}
if(sw_cancel==0){delay_ms(150);goto pin3;}
else goto pin4;
```

```
set1:
Set_point1=Set_point1;
if(sw_up==0) {Set_point1++;delay_ms(10);}
```

```

if(sw_down==0) {Set_point1=Set_point1-1;delay_ms(10);}
if(Set_point1>30){Set_point1=0;delay_ms(10);}
if(Set_point1<0) {Set_point1=30;delay_ms(10);}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Set Point1");
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Ketinggian:%i",Set_point1);
lcd_puts(lcd_buff);
delay_ms(100);
if(sw_ok==0) {delay_ms(150);goto set2;}
if(sw_cancel==0){delay_ms(150);goto pin4;}
else
goto set1;

set2:
Set_point2=Set_point2;
if(sw_up==0) {Set_point2++;delay_ms(10);}
if(sw_down==0) {Set_point2=Set_point2-1;delay_ms(10);}
if(Set_point2>30){Set_point2=0;delay_ms(10);}
if(Set_point2<0) {Set_point2=30;delay_ms(10);}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Set Point2");
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Ketinggian:%i",Set_point2);
lcd_puts(lcd_buff);
delay_ms(100);
if(sw_ok==0) {delay_ms(150);goto set3;}
if(sw_cancel==0){delay_ms(150);goto set1;}
else
goto set2;

set3:
Set_point3=Set_point3;
if(sw_up==0) {Set_point3++;delay_ms(10);}
if(sw_down==0) {Set_point3=Set_point3-1;delay_ms(10);}
if(Set_point3>30){Set_point3=0;delay_ms(10);}
if(Set_point3<0) {Set_point3=30;delay_ms(10);}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Set Point3");
lcd_gotoxy(0,1);

```

```
sprintf(lcd_buff,"Ketinggian:%i",Set_point3);  
lcd_puts(lcd_buff);  
delay_ms(100);  
if(sw_ok==0) {delay_ms(150);goto neutron;}  
if(sw_cancel==0){delay_ms(150);goto set2;}  
else  
goto set3;
```

```
neutron:  
lcd_clear();  
lcd_gotoxy(1,0);  
lcd_putsf("Save To EEPROM");  
delay_ms(200);
```

```
}
```

```
void tampilan_awal()
```

```
{  
lcd_init (16);  
lcd_gotoxy(0,0);  
lcd_putsf("#####");  
lcd_gotoxy(4,1);  
lcd_putsf("MENTARI");  
delay_ms(2000);  
lcd_clear();  
}
```

```
void inisialisasi_ADC()
```

```
{  
// ADC initialization  
// ADC Clock frequency: 750,000 kHz  
// ADC Voltage Reference: AVCC pin  
// ADC High Speed Mode: On  
// ADC Auto Trigger Source: None  
ADMUX=ADC_VREF_TYPE;  
ADCSRA=0x84;  
SFIOR&=0xEF;  
SFIOR|=0x10;  
}
```

```

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
  ADMUX=adc_input|ADC_VREF_TYPE;
  // Start the AD conversion
  ADCSRA|=0x40;
  // Wait for the AD conversion to complete
  while ((ADCSRA & 0x10)==0);
  ADCSRA|=0x10;
  return ADCW;
}

```

```

void inisialisasi_port()
{

```

```

  DDRA=0b00000000;
  PORTA=0b00000000;
  DDRB=0b00111111;
  DDRD=0b00111111;
  PORTB=0b11000000;
  PORTD=0b11000000;

```

```

}

```

```

void waktu()
{
  error=sensor1-Set_point1;
  pb=500;
  if(error>=6){pb=1000;}
  if(error>=11){pb=2000;}

```

```

  if(Set_point1>=0){tunggu=tunggu1;}
  if(Set_point1>=6){tunggu=tunggu2;}
  if(Set_point1>10){tunggu=tunggu3;}

```

```

  if(error<1){tunggu1=0;}
  else if(error==1){tunggu1=7000;}
  else if(error==2){tunggu1=17000;}
  else if(error==3){tunggu1=2600;}
  else if(error==4){tunggu1=36000;}

```

```
else if(error==5){tunggu1=45000;}
else if(error==6){tunggu1=29000;}
else if(error==7){tunggu1=35000;}
else if(error==8){tunggu1=40000;}
else if(error==9){tunggu1=46000;}
else if(error==10){tunggu1=50000;}
else if(error==11){tunggu1=35000;}
else if(error==12){tunggu1=38000;}
else if(error==13){tunggu1=40000;}
else if(error==14){tunggu1=45000;}
else if(error==15){tunggu1=48000;}
```

```
if(error<1){tunggu2=0;}
else if(error==1){tunggu2=10000;}
else if(error==2){tunggu2=18000;}
else if(error==3){tunggu2=28000;}
else if(error==4){tunggu2=38000;}
else if(error==5){tunggu2=48000;}
else if(error==6){tunggu2=31000;}
else if(error==8){tunggu2=39000;}
else if(error==9){tunggu2=41000;}
else if(error==10){tunggu2=49000;}
else if(error>10){tunggu2=53000;}
```

```
if(error<1){tunggu3=0;}
else if(error==1){tunggu3=13000;}
else if(error==2){tunggu3=23000;}
else if(error==3){tunggu3=33000;}
else if(error==4){tunggu3=44000;}
else if(error==5){tunggu3=55000;}
else if(error>6){tunggu3=60000;}
```

```
error1=error;
```

```
}
```

```
void pintu1()
```

```
{
if(sensor1<=(Set_point1+1)>=(Set_point1-1))
{stop1();stop2();}
```

```
if(sensor1>(Set_point1+1))
```

```

{if((sensor2-Set_point2)>(sensor3-Set_point3))
{buka2();delay_ms(pb);stop2();delay_ms(tunggu);tutup2();delay_ms(pb-
200);stop2();delay_ms(100);}
{if((sensor2-Set_point2)<(sensor3-Set_point3))
{buka1();delay_ms(pb);stop1();delay_ms(tunggu);tutup1();delay_ms(pb-
200);stop1();delay_ms(100);}
{if((sensor2-Set_point2)==(sensor3-Set_point3))
{buka1();buka2();delay_ms(pb);stop1();stop2();delay_ms(tunggu);tutup1();tutup2();d
elay_ms((pb)-200);stop1();stop2();delay_ms(100);}}}
}

```

```

void pintu2()
{
if(sensor2<=(Set_point2+2)>=(Set_point2-2))
{stop3();}

if(sensor2>(Set_point2+2))
{buka3();delay_ms(3000);stop3();delay_ms(3000);tutup3();delay_ms(3000);stop3();d
elay_ms(100);}

}

```

```

void pintu3()
{
if(sensor3<=(Set_point3+2)>=(Set_point3-2))
{stop4();}

if(sensor3>(Set_point3+2))
{buka4();delay_ms(3000);stop4();delay_ms(3000);tutup4();delay_ms(3000);stop4();d
elay_ms(100);}

}

```

```

void sensor()
{
sensor1=read_adc(0)/11; //baca adc0 10bit (0-1024)
if(sensor1<1) {sensor1=0;}
sensor2=read_adc(1)/21; //baca adc1 10bit (0-1024)
if(sensor2<1) {sensor2=0;}
sensor3=read_adc(2)/21; //baca adc2 10bit (0-1024)
if(sensor3<1) {sensor3=0;}
}

```



```
lcd_clear    ();
lcd_gotoxy   (0,0);
sprintf      (lcd_buff,"TA1=%i",sensor1);
lcd_puts     (lcd_buff);
lcd_gotoxy   (7,0);
sprintf      (lcd_buff,"PID=%i",tunggu);
lcd_puts     (lcd_buff);
lcd_gotoxy   (0,1);
sprintf      (lcd_buff,"TA2=%i",sensor2);
lcd_puts     (lcd_buff);
lcd_gotoxy   (7,1);
sprintf      (lcd_buff,"TA3=%i",sensor3);
lcd_puts     (lcd_buff);
delay_ms(100);
}
```

```
void main(void)
{
  inialisasi_port();
  inialisasi_ADC();
  tampilan_awal();
  setting_awal();
  while(1)
  {
    sensor();
    waktu();
    pintu1();
    sensor();
    waktu();
    pintu2();
    sensor();
    waktu();
    pintu3();
  };
}
```

LAMPIRAN 2

FOTO-FOTO KEGIATAN PENELITIAN

