

IV. HASIL DAN PEMBAHASAN

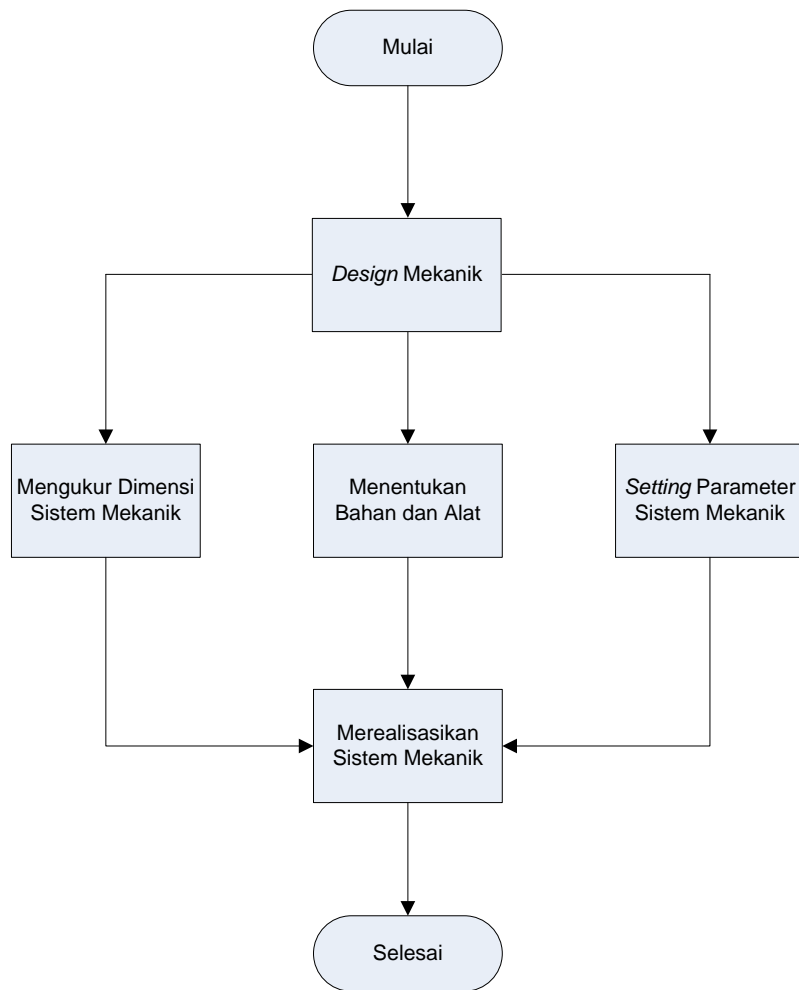
A. Perencanaan Dan Pembuatan Perangkat Keras Dan Perangkat Lunak

1. Sistem Mekanik

Sistem Pengendalian Posisi Model Panel Surya Berbasis Mikrokontroler ATmega8535 memiliki suatu perancangan sistem mekanik berupa sebuah sistem dudukan model sistem panel surya dengan kaca akrilik sebagai model panel surya. Sistem sendiri diletakkan secara simetris searah pasisi barat, timur, utara, dan selatan hal ini disebabkan karena sistem pergerakan model panel surya terbatas pada gerakan kanan kiri yang dapat diasumsikan sebagai gerak ke barat timur.

Perangkat sistem mekanik terdiri dari dudukan model panel surya berupa bangunan terbuat dari bahan kayu dan kaca akrilik. Semuanya terintegrasi sebagai suatu kesatuan sistem.

Berikut sistem kerja dari Sistem Mekanik Pengendalian Posisi Model Panel Surya Berbasis Mikrokontroler ATmega8535, sebagai poros gerak utama arah timur-barat (*horizontal*) terhubung dengan poros motor servo. Selain terhubung dengan model panel surya poros motor juga terhubung dengan potensiometer sebagai sensor. Gerak yang dihasilkan akan menghasilkan perubahan sudut poros motor yang sebanding juga dengan sudut dari potensiometer.



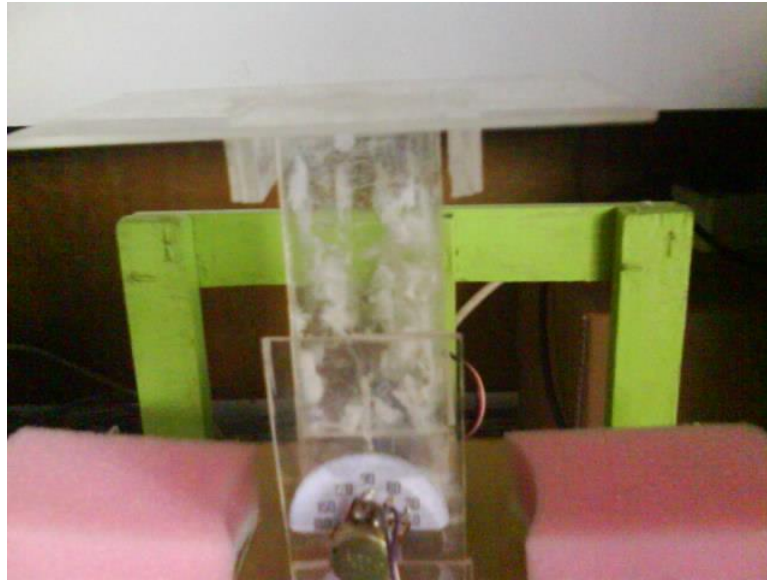
Gambar 4.1. Diagram Alir Sistem Mekanik

Sistem mekanik ini terdiri dari 3 bagian :

a. Sistem Model Panel Surya

Sistem ini terbuat dari bahan kaca akrilik berbentuk persegi panjang sebagai model panel surya yang terhubung dengan kaki berupa kaca akrilik juga dan kaki ini yang terhubung dengan poros motor. Menggunakan lem perekat dalam menyambung setiap bagian model panel surya ini. Dimensi dari model panel surya ini adalah :

Panjang : 11 cm
Lebar : 7,5 cm
Tinggi kaki : 10 cm



Gambar 4.2. Model panel surya

b. Dudukan sistem gerak

Terhubung dengan poros motor servo sehingga sudut posisi yang diinginkan sesuai dengan perubahan sudut servo. Motor servo sendiri terletak di atas sistem dudukan kaca akrilik berbentuk kotak persegi panjang terletak diatas bangunan mekanik dari kayu yang merupakan peletakan semua komponen-komponen elektronika yang digunakan. Dudukan servo motor berupa bidang kaca akrilik dengan dimensinya adalah :

Panjang : 15 cm
Lebar : 12 cm
Tebal : 0,3 mm

Sedangkan dudukan untuk sistem mekanik pada dudukan motor servo ini adalah dari bahan kaca akrilik transparan dengan dimensi :

Panjang : 22,5 cm

Lebar : 16 cm

Tinggi : 6 cm

c. Dudukan Rangkaian Elektronika

Terdiri dari tiga tingkat dengan ukuran dari dudukan rangkaian elektronika yaitu :

Tinggi : 18,5 cm

Lebar : 20,5 cm

Panjang : 25,5 cm

Tingkat I : 0,8 mm

Tingkat II : 9 cm



Gambar 4.3. Dudukan Rangkaian Elektronika

Tata letak dimaksudkan agar komponen-komponen elektronika yang digunakan memiliki posisi yang tetap, dan kokoh. Tingkat dasar terdapat rangkaian *power supply* dengan trafo untuk masukan dari sumber AC, di tingkat dua terdapat mikrokontroler sebagai pengendali utama dan di tingkat tiga terdapat tampilan LCD, *keypad* untuk isyarat masukan yang akan diproses mikrokontroler, dan dudukan sistem gerak dc motor servo yang terhubung langsung dengan kaki sistem model panel surya. Secara keseluruhan adalah seperti gambar berikut :



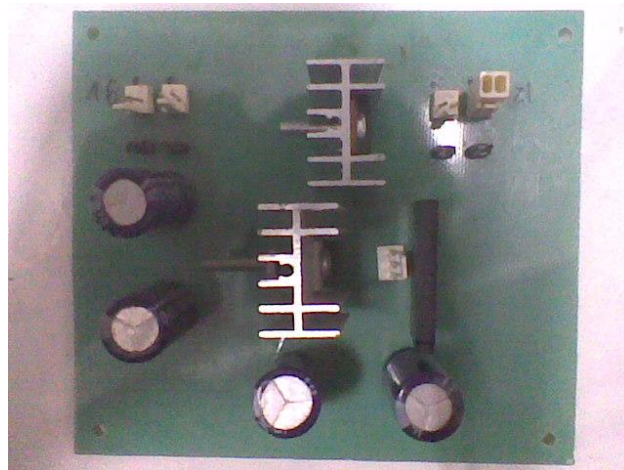
Gambar 4.4. Model Sistem Pengendalian Panel Surya

2. Rangkaian Elektronika

Terdiri dari rangkaian *power supply*, rangkaian pengendali dan rangkaian untuk sensor posisi berupa potensiometer. *Power supply* untuk pembangkit tegangan untuk komponen-komponen di dalam rangkaian pengendali. Rangkaian pengendali terdiri dari mikrokontroler ATmega8535 sebagai pengendali utama, mengendalikan putaran motor, sensor, tampilan di LCD, dan masukan melalui *keypad*.

a. *Power Supply*

Memiliki 9 V digunakan sebagai tegangan masukan untuk mikrokontroler ATmega 8535. Untuk mendapatkan tegangan 9 V yang stabil digunakan sebuah IC *regulator* LM7809. Karena pada modul DT-AVR terdapat IC *regulator* LM7805 yang menghasilkan tegangan 5 V, sehingga pada rangkaian *power supply* tidak lagi menggunakan IC *regulator* LM7805. Gambar *power supply* seperti berikut :



Gambar 4.5. Rangkain *Power Supply*

b. Rangkaian Pengendali Mikrokontroler ATmega8535

Tegangan sumber V_s 9 V dan *ground* didapat dari *power supply* melalui pin-pin konektor. Mikrokontroler ATmega8535 merupakan pengendali utama untuk mengatur gerak motor yaitu dari isyarat keluaran pin-pin *port B*, posisi panel surya melalui isyarat keluaran dari potensiometer, dan isyarat keluaran *keypad* serta isyarat keluaran untuk masukan LCD.

c. Rangkaian Tampilan dan *Keypad*

Keypad terhubung dengan *port C* dari Mikrokontroler ATmega8535, sebagai masukan untuk menjalankan sistem pengendalian model posisi panel surya. Tampilan berupa LCD 2x16 karakter memiliki 16 pin, pin 1 dan 2 berupa *ground* dan V_s 5 V_{dc} yang diambil dari keluaran mikrokontroler ATmega8535, pin 3 merupakan pengatur kontras tampilan yang terhubung dengan potensiometer, potensiometer memiliki 3 pin, 2 pin lainnya adalah V_s dan *ground* yang bernilai sama dengan V_s LCD, sehingga diambil pula dari keluaran pin V_s dan *ground* mikrokontroler ATmega8535 yang lain.

d. Rangkaian Sensor

Menggunakan potensiometer, potensiometer terhubung dengan poros dc motor servo pada sistem mekanik pengendalian posisi model panel surya. 3 kaki pin potensiometer, 2 pin adalah V_s 5 V dan *ground* sedangkan 1 pin lainnya ke pin *port A* Mikrokontroler ATmega8535.



Gambar 4.6. Potensiometer terhubung ke poros motor

3. Perangkat Lunak

Perangkat lunak dimaksudkan untuk menuliskan program yang akan di *download* ke mikrokontroler. Perangkat lunak yang digunakan adalah CodeVisionAVR. CodeVisionAVR merupakan *software C- cross compiler*, dimana program dapat ditulis menggunakan bahasa-C. Dengan menggunakan pemrograman bahasa-C diharapkan waktu disain (*developing time*) akan menjadi lebih singkat. Setelah program dalam bahasa-C ditulis dan dilakukan kompilasi tidak terdapat kesalahan (*error*) maka proses *download* dapat dilakukan. Mikrokontroler AVR mendukung sistem *download* secara *In-System Programming* (ISP).

Cara memulai *project* baru pada CodeVisionAVR adalah :

1. Jalankan *software* CodeVisionAVR.
2. Buat *project* baru pilih **File**→**New** pilih *project* lalu tekan tombol OK.
3. Kemudian muncul dialog apakah akan menggunakan CodeWizard AVR untuk mempermudah merancang kerangka program. Pilih YES.
4. *Board* yang digunakan menggunakan chip ATMega8535 dengan *clock* 4 MHz, kemudian pilih *tab Ports*.
5. Kemudian pilih **File**→**Generate, Save and Exit**.

B. Pengujian Perangkat Keras Dan Perangkat Lunak

1. Pengujian *Port* Mikrokontroler

Pengujian *Port* mikrokontroler dimaksudkan untuk mengecek apakah data yang dimasukkan (masukan) dan dikeluarkan (keluaran) mikrokontroler sesuai dengan deskripsi kerja sistem. Untuk simulasi awal pengecekan I/O menggunakan

simulasi nyala led dengan menggunakan program sederhana menyalakan led di *port A*. Berikut ini merupakan listing program menyalakan led di *port A*:

```
{
    while (1)
    {
        PORTA=PORTA-1;
        delay_ms(300);
    };
}
```

Berdasarkan program diatas maka tampilan yang didapatkan pada nyala led yaitu pada saat program pertama kali dijalankan maka led akan menyala yaitu led yang dihubungkan dengan P1.0 sampai dengan P1.3 kemudian setelah selang waktu yang telah ditentukan pada *delay* maka nyala led akan berjalan dari awal hingga akhir kemudian diulang lagi mulai awal hingga akhir begitu seterusnya.

Pengalamatan *Port* Mikrokontroler ATmega8535 yaitu sebagai berikut :

Tabel 4.1. Pengalamatan *Port* Mikrokontroler

Alamat	Keterangan
<i>Port A.2</i>	Sensor
<i>Port B.0</i>	RS (LCD Pin 4)
<i>Port B.1</i>	RD (LCD Pin 5)
<i>Port B.2</i>	EN (LCD Pin 6)
<i>Port B.3</i>	<i>Free</i>
<i>Port B.4</i>	DB4 (LCD Pin 11)
<i>Port B.5</i>	DB5 (LCD Pin 12)
<i>Port B.6</i>	DB6 (LCD Pin 13)
<i>Port B.7</i>	DB7 (LCD Pin 14)
<i>Port C.0</i>	<i>Keypad</i> (Kolom-1)
<i>Port C.1</i>	<i>Keypad</i> (Kolom-2)
<i>Port C.2</i>	<i>Keypad</i> (Kolom-3)
<i>Port C.3</i>	<i>Keypad</i> (Kolom-4)
<i>Port C.4</i>	<i>Keypad</i> (Baris-1)

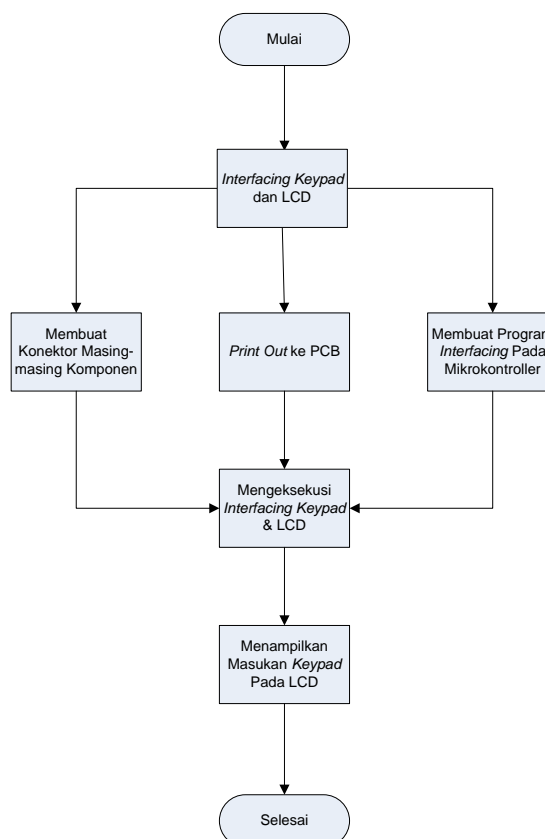
Tabel 4.1. Pengalamatan *Port* Mikrokontroler Lanjutan.

Alamat	Keterangan
<i>Port C.5</i>	<i>Keypad</i> (Baris-2)
<i>Port C.6</i>	<i>Keypad</i> (Baris-3)
<i>Port C.7</i>	<i>Keypad</i> (Baris-4)
<i>Port D.2</i>	Motor Servo

2. Pengujian *Keypad* Dan LCD

Pengujian *keypad* dan LCD dilakukan untuk dapat mengetahui baik atau tidaknya *keypad* dan LCD bekerja.

Diagram Alir pemrograman untuk menu masukan melalui *keypad* dan tampilan di LCD adalah sebagai berikut :

Gambar 4.7. Diagram Alir *Interfacing Keypad* dan LCD

Melalui program uji *keypad* dan LCD pada mikrokontroler ATmega8535 didapatkan hasil sebagai berikut :

Tabel 4.2. Uji *Keypad* dan LCD

Tekan Keypad	Tampilan LCD
A	A
B	B
C	C
D	D
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
*	*
#	#

Sedangkan untuk mengatur terang gelap LCD menggunakan potensiometer.

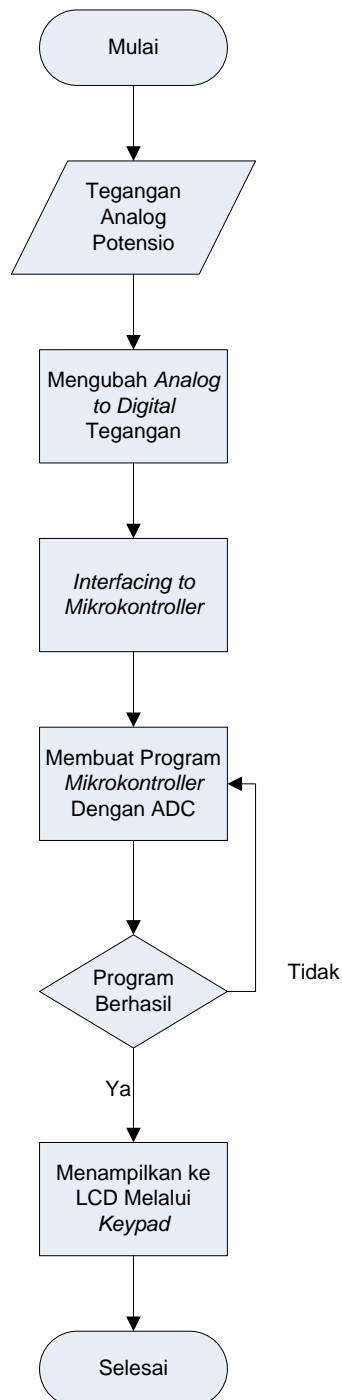
3. Pengujian Sensor Potensiometer

Hasil uji untuk potensiometer berdasarkan perubahan nilai resistansi terhadap terang dan gelapnya tampilan di LCD. Saat nilai resistansi rendah tampilan LCD sangat terang, begitu juga dengan sebaliknya.

Tabel 4.3. Hasil uji potensiometer

Putar Potensiometer	Tampilan LCD
Penuh Kekiri	Sangat Terang
Ketengah	Terang
Penuh Kekan	Agak Terang

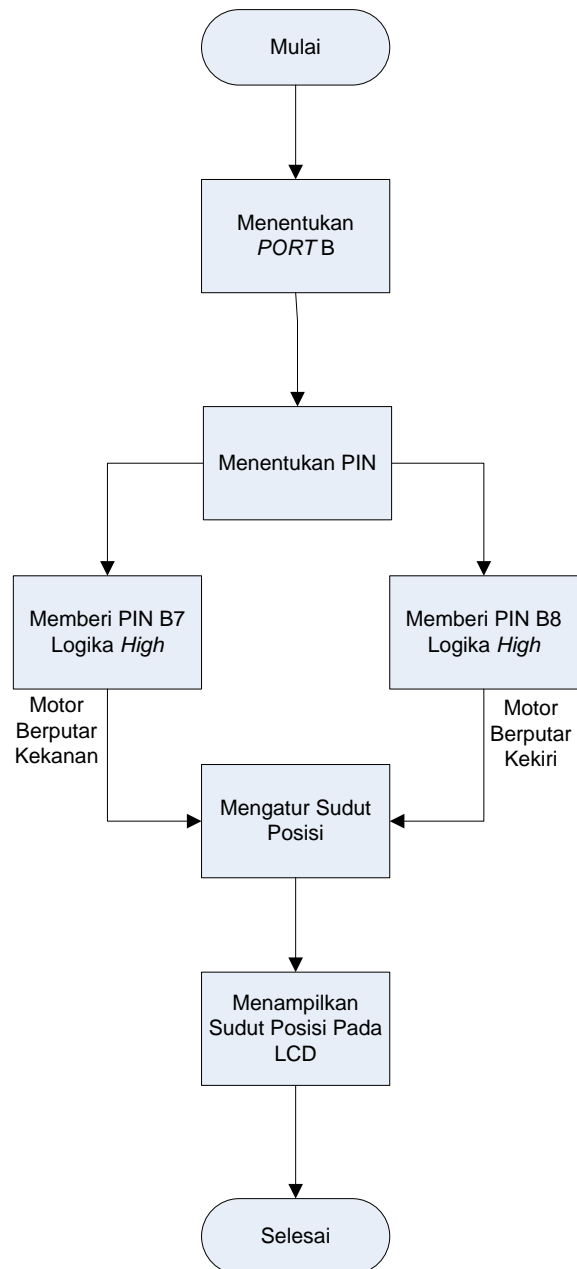
Diagram Alir pengujian pemrograman untuk sensor potensiometer ini adalah sebagai berikut :



Gambar 4.8. Diagram Alir Pengujian Sensor Potensiometer

4. Pengujian Motor

Untuk mengaktifkan motor keluaran diberikan logika 1, sedangkan untuk menonaktifkan diberikan logika 0.



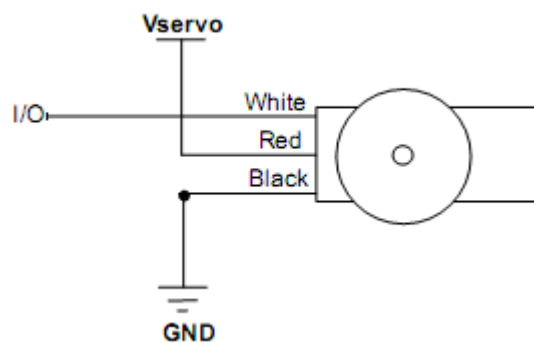
Gambar 4.9. Diagram Alir Pengujian Motor

C. Analisa dan Pembahasan

1. Model Alat

Pemodelan sistem dengan menggunakan dc motor servo *continuous* dengan sebidang bahan akrilik sebagai model panel sedangkan poros motor servo dihubungkan dengan potensiometer.

Untuk kaki-kaki pin motor servo dihubungkan ke mikrokontroler ATmega8535 melalui konektor-konektor, seperti pada gambar berikut :



Gambar 4.10. Pin Motor Servo

Tegangan yang dibutuhkan servo motor adalah 5 V dc, sehingga bisa diambil juga dari keluaran mikrokontroler V_{cc} dan *ground*. Begitu juga dengan potensiometer memiliki 3 pin dimana pin 2 terhubung dengan pin *Port A* pada mikrokontroler sedangkan 2 pin lainnya terhubung dengan V_{cc} dan *ground* dari mikrokontroler juga. Untuk pin potensiometer 1 maupun 3 dapat masing-masing bernilai positif ataupun negatif, hanya saja berbeda untuk keluaran nilai ADC yang terbaca potensiometer atau hanya berkebalikan saja.

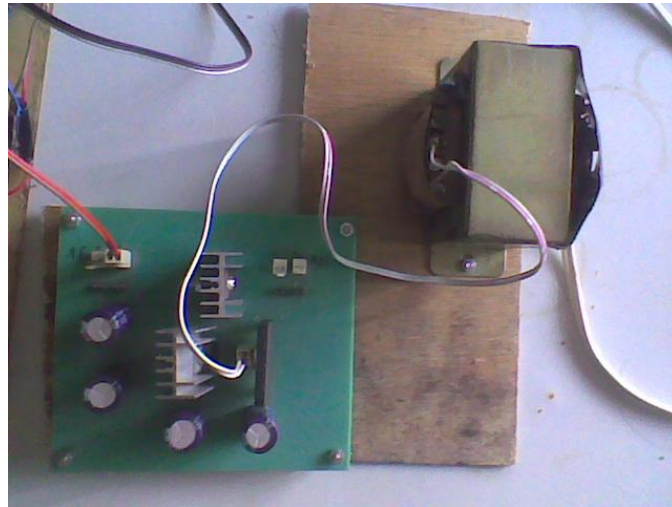
Selain terhubung ke servo motor dan potensiometer, mikrokontroler juga terhubung dengan *keypad* dan LCD. *Keypad* butuh 8 kaki pin, sesuai dengan jumlah pin I/O di satu *port* mikrokontroler, sedangkan untuk LCD memiliki 16 kaki pin. Pin 1 dan 2 adalah *ground* dan V_{cc} , pin 3 terhubung dengan potensiometer lain untuk pengaturan kontras tampilan di LCD.

Masih pada LCD, pin 4 – 6 terhubung dengan pin 1-3 dari pin I/O mikrokontroler, pin 7 – 10 kaki LCD merupakan pin bebas dan juga tidak terhubung ke mikrokontroler, pin 11 – 14 terhubung dengan pin 5 – 8 *Port B* mikrokontroler, 2 sisa pin lainnya merupakan V_{cc} dan *ground*.

Tabel 4.4. Pengalamatan LCD

Pin LCD	Fungsi	Keterangan	Port B
1	GND	0 V	
2	VCC	5 V	
3	VEE	Pengaturan <i>Contras</i> LCD (vr 10 k)	
4	RS	<i>Register Selec</i>	Port B.0
5	R/W	<i>Read/write</i>	Port B.1
6	E	<i>Enable clock</i>	Port B.2
7	Data 0	<i>Data Bus 0</i>	
8	Data 1	<i>Data Bus 1</i>	
9	Data 2	<i>Data Bus 2</i>	
10	Data 3	<i>Data Bus 3</i>	
11	Data 4	<i>Data Bus 4</i>	Port B.4
12	Data 5	<i>Data Bus 5</i>	Port B.5
13	Data 6	<i>Data Bus 6</i>	Port B.6
14	Data 7	<i>Data Bus 7</i>	Port B.7
15	VCC	5 V	
16	GND	0 V	

Mikrokontroler memerlukan tegangan masukan 9 V, untuk memenuhinya digunakan rangkaian *power supply* yang memiliki tegangan keluaran 9 V. Seperti pada gambar berikut :



Gambar 4.11. *Power Supply*

2. Pengujian Model Sistem

a. Hasil Pengukuran

Untuk penentuan sudut posisi ideal agar posisi model panel surya tegak lurus atau mendekati tegak lurus dengan surya, dipilih nilai-nilai sudut dengan tingkat ketelitian adalah setiap 1 jam dengan asumsi waktu terbit ideal bagi surya adalah pukul 06.00 dan waktu tenggelam surya adalah pukul 18.00. Sedangkan untuk range posisi sudut model panel surya sendiri adalah pada pukul 08.00 – 16.00 yaitu 30° – 150° . Dasar penentuan sudut posisi tersebut adalah bahwa posisi surya sepanjang tahun adalah sesuai atau mendekati waktu terbit dan tenggelam surya pada pukul 06.00 dan pukul 18.00 tersebut. Berikut data posisi surya menurut perhitungan pada posisi Laboratorium Teknik Elektro Universitas Lampung dengan letak astronomi

Latitude : 05:21,6916' *South* = 05:21':41,514" Lintang Selatan

Longitude : 105:14,5632 *East* = 105:14':30,84" Bujur Timur

Tabel 4.5. Posisi Surya Sepanjang Tahun

No	Tanggal	Posisi Surya	
		Pukul 06.00	Pukul 18.00
1	1 Januari 2009	1 ⁰	178 ⁰
2	15 Januari 2009	0 ⁰	177 ⁰
3	1 Februari 2009	1 ⁰	176 ⁰
4	15 Februari 2009	-2 ⁰	176 ⁰
5	1 Maret 2009	-2 ⁰	177 ⁰
6	15 Maret 2009	-1 ⁰	178 ⁰
7	1 April 2009	-1 ⁰	180 ⁰
8	15 April 2009	0 ⁰	181 ⁰
9	1 Mei 2009	0 ⁰	182 ⁰
10	15 Mei 2009	0 ⁰	182 ⁰
11	1 Juni 2009	-1 ⁰	182 ⁰
12	15 Juni 2009	-1 ⁰	182 ⁰
13	1 Juli 2009	-2 ⁰	181 ⁰
14	15 Juli 2009	-3 ⁰	180 ⁰
15	1 Agustus 2009	-2 ⁰	180 ⁰
16	15 Agustus 2009	-2 ⁰	180 ⁰
17	1 September 2009	0 ⁰	181 ⁰
18	15 September 2009	1 ⁰	181 ⁰
19	1 Oktober 2009	3 ⁰	182 ⁰
20	15 Oktober 2009	4 ⁰	182 ⁰
21	1 November 2009	5 ⁰	182 ⁰
22	15 November 2009	5 ⁰	182 ⁰
23	1 Desember 2009	4 ⁰	180 ⁰
24	15 Desember 2009	3 ⁰	180 ⁰

Dari tabel tersebut diperoleh sebuah pendekatan ideal untuk waktu terbit dan tenggelam surya yaitu pada pukul 06.00 dan pukul 18.00. Untuk pendekatan sudut posisi dengan ketelitian per jam yaitu sebesar 15⁰.

Sehingga diperoleh data posisi ideal untuk penentuan sudut posisi surya yaitu terbit pukul 06.00 dan tenggelam pada pukul 18.00. Hasil Pengukuran perbandingan nilai ADC potensiometer dengan sudut model sistem :

Tabel 4.6. Hasil Pengukuran ADC berdasarkan sudut

Sudut	ADC									
	1	2	3	4	5	6	7	8	9	10
30 ⁰	194	192	193	198	193	194	192	193	192	195
45 ⁰	173	173	174	173	171	172	173	173	174	173
60 ⁰	153	153	160	154	155	152	156	157	157	152
75 ⁰	132	132	131	132	135	132	129	128	133	128
90 ⁰	109	111	112	111	112	112	108	110	110	110
105 ⁰	87	88	90	88	87	90	89	86	84	87
120 ⁰	64	65	65	67	64	65	66	65	62	63
135 ⁰	46	42	46	48	47	48	47	44	45	45
150 ⁰	26	27	27	28	27	28	29	27	27	26

Penentuan waktu berdasarkan sudut :

Tabel 4.7. Penentuan Waktu Menurut Sudut

Waktu	Sudut
08.00	30 ⁰
09.00	45 ⁰
10.00	60 ⁰
11.00	75 ⁰
12.00	90 ⁰
13.00	105 ⁰
14.00	120 ⁰
15.00	135 ⁰
16.00	150 ⁰

b. Pengoperasian Alat

Masukan melalui *keypad*, seperti pada tampilan awal di LCD. Dengan tekan bintang (*) pada *keypad* maka pada LCD akan tampil menu panduan

pengoperasian alat. Servo *continuous* motor digerakkan ke posisi yang dimaksud yaitu pada sudut 150^0 atau nilai ADC sekitar 22, dengan menggunakan tombol angka 9. Pada pengoperasian alat ini mikrokontroler akan merespon setelah penekanan tombol *keypad*, sesuai instruksi tersebut motor servo akan bergerak ke posisi yang dimaksud. Sistem servo merupakan sistem kendali terbuka karena telah diperoleh posisi stabil untuk masing-masing posisi sudut. Penggunaan *keypad* dimaksudkan sebagai masukan untuk menjalankan *looping* program pada mikrokontroler, alasan penggunaan *keypad* ini agar sistem dapat dengan mudah dioperasikan dan dilakukan pengendalian posisi pada saat terdapat gangguan pada posisi sistem. Setelah posisi yang dimaksud tercapai, seperti halnya pada panduan, terdapat pilihan-pilihan posisi sudut yang bisa dituju, dapat dilihat dalam tabel berikut :

Tabel 4.8. Penjelasan Tombol *Keypad*

Tekan <i>Keypad</i>	Tampilan LCD
1	Sudut 30^0
2	Sudut 45^0
3	Sudut 60^0
4	Sudut 75^0
5	Sudut 90^0
6	Sudut 105^0
7	Sudut 120^0
8	Sudut 135^0

Pengoperasian alat sendiri, setelah diperoleh salah satu sudut yang diinginkan, maka posisi sudut servo diatur kembali seperti langkah awal yaitu memposisikan sudut ke 150^0 dengan menggunakan tombol angka 9, setelah itu baru memilih sudut posisi berapa yang akan dipilih.

Selain tombol-tombol diatas, berikut adalah penjelasan tombol-tombol *keypad* lainnya :

Tabel 4.9. Penjelasan Tombol Lain

Tekan Keypad	Tampilan LCD
0	Tampilan Awal
A	Tampilan ADC
B	Tampilan Sudut
C	Tampilan Waktu
D	Tampilan ADC, Sudut, dan Waktu

Poin-poin dalam penyelesaian tugas akhir ini, dibuat sebuah pemodelan sistem yang hampir mirip secara teknis maupun prinsip kerjanya berdasar pada tujuan tugas akhir ini, yaitu diperoleh :

1. Sistem sederhana pengendalian posisi.
2. Pembuatan rangkaian elektronika yang sesuai dengan tujuan tugas akhir.
3. Pemrograman mikrokontroler ATmega8535 sesuai dengan sistem sederhana pengendalian posisi.

Disamping itu banyak kekurangan juga dalam keakuratan posisi ideal yang diinginkan, yaitu posisi sudut yang selalu tegak lurus dengan cahaya surya.

Faktor-faktor itu antara lain :

1. Adanya tegangan bocor yang membuat servo *continous* motor berubah posisi sudut.
2. Pendekatan sudut posisi surya, karena surya setiap hari mengalami perbedaan waktu terbit dan tenggelam juga adanya akibat dari rotasi dan revolusi bumi yang menyebabkan pergeseran semu surya.

3. Keterbatasan kemampuan *programming* sehingga masih perlu banyak pembenahan dalam perancangan dan pembuatan program.

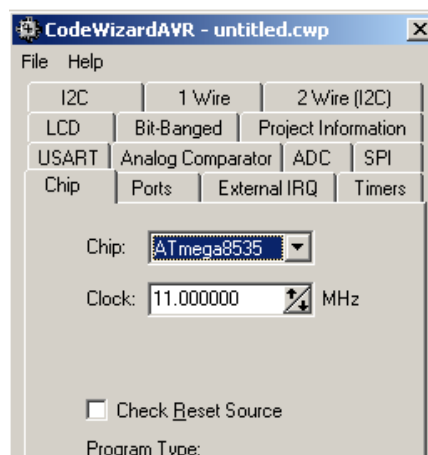
Dalam pembuatan program keseluruhan itu melalui tahap-tahap yaitu :

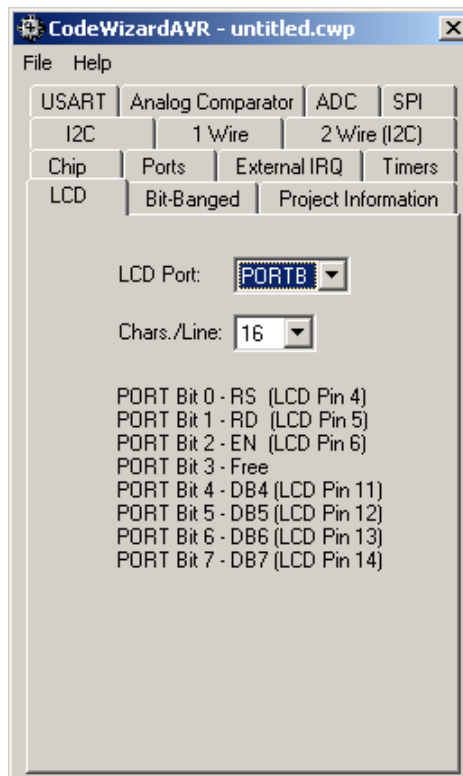
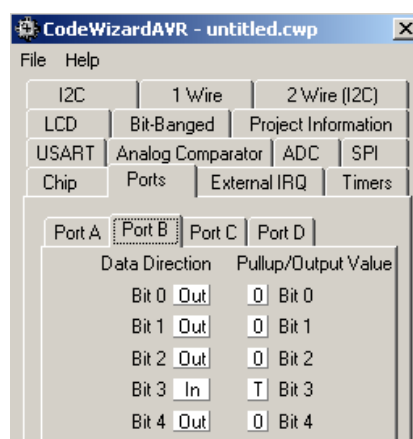
1. Menentukan *Port A* untuk potensiometer yang diatur sebagai sensor posisi.
2. Menentukan *Port B* untuk LCD.
3. Menentukan *Port C* untuk *keypad*.
4. Menentukan *Port D* untuk servo *continuous* motor.

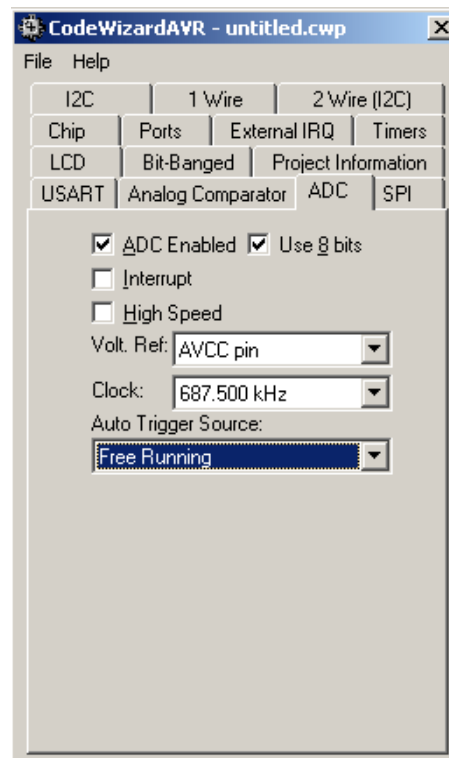
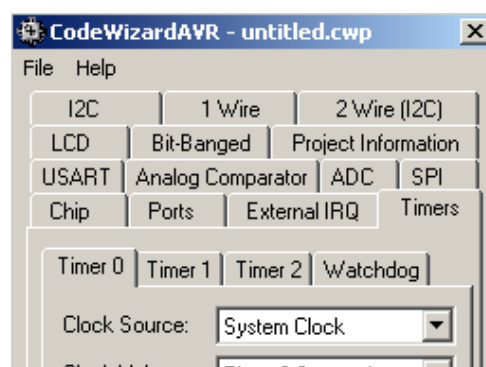
Memasukkan fungsi tambahan *keypad* ke *inc folder* cvavr sehingga dapat memanggil dan mengoperasikan *keypad*. Sedangkan untuk fungsi ADC, LCD, dan pwm sudah terdapat di cvavr.

3. Pemrograman

Pembuatan program dilakukan dengan menggunakan software codevision avr. Memilih *new project* dan menggunakan CodeWizardAVR. Menentukan parameter-parameter yang akan digunakan, yaitu *setting chip* ke ATmega8535, *setting* ADC dengan *Port A* sebagai keluaran ADC, *setting Port B* untuk LCD, dan *setting timer* seperti pada gambar berikut :



Gambar 4.12. *Setting chip ATmega8535 pada codevision avr*Gambar 4.13. *Setting LCD pada Port B*

Gambar 4.14. *Setting Port B*Gambar 4.15. *Setting ADC*

Gambar 4.16. *Setting Timer 0*

Pilih menu file, generate, save and exit kemudian simpan 3 kali dengan nama yang sama yaitu Kendali Posisi Panel Surya.

Menambahkan fungsi-fungsi utama yaitu :

```
#include <keypad.h>
#include <stdio.h>
#include <delay.h>
```

Kemudian mendefinisikan `pwm` pada *Port D.2* untuk perputaran motor servo ke kiri mencapai posisi awal untuk menentukan sudut yang diinginkan yaitu :

```
#define pwm PORTD.2
```

Mendeklarasikan variabel yaitu `pwmKi_val`, `pwmKa_val`, dan `x`, dengan menuliskan :

```
unsigned char pwmKi_val,pwmKa_val,x = 0;
```


Membuat *listing* program pada *interrupt timer* :

```
TCNT0 = 0xC0;
x++;
if (x>=pwmKa_val) {
    pwm=0;
} else {
    pwm=1;
}
if (x>=pwmKi_val) {
    pwm=0;
} else {
    pwm=1;
}
```

Untuk pengendalian motor servo berputar mencapai sudut yang diinginkan dengan menuliskan :

```
#define ctrl PORTD.2
```

Sedangkan *listing* program untuk menentukan posisi sudut yaitu :

```
void putar_servo30 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=12.5; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo45 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=12; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo60 (unsigned char input)
{
    unsigned char i,j;
```

```
    for(i=1; i<=11.999; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo75 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=11.8; i++)

    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo90 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=10.8; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo105 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=9.8; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
void putar_servo120 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=7; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}
```

```

void putar_servo135 (unsigned char input)
{
    unsigned char i,j;
    for(i=1; i<=4.2; i++)
    {
        ctrl=1;
        for (j=1; j<=input; j++)
            delay_us(10);
        ctrl=0;
        delay_ms(20);
    }
}

```

Terdapat 8 pilihan sudut yang dapat dicapai, yaitu : sudut 30° , 45° , 60° , 75° , 90° , 105° , 120° , dan 135° .

Untuk memanggil **void** penentuan sudut-sudut tersebut dilakukan melalui *keypad*, dengan masing-masing fungsi tombol pada *keypad*, yaitu :

```

void Push_1catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo30(data);
    data=20;
    if (data==20) data=200;
}
void Push_2catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo45(data);
    data=20;
    if (data==20) data=200;
}
void Push_3catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo60(data);
    data=20;
    if (data==20) data=200;
}
void Push_4catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
}

```

```

    putar_servo75(data);
    data=20;
    if (data==20) data=200;
void Push_5catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo90(data);
    data=20;
    if (data==20) data=200;
void Push_6catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo105(data);
    data=20;
    if (data==20) data=200;
void Push_7catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo120(data);
    data=20;
    if (data==20) data=200;
void Push_8catch(void)
{
    unsigned char data;
    PORTD=0x00;
    DDRD=0x04;
    data=200;
    putar_servo135(data);
    data=20;
    if (data==20) data=200;

```

Dalam *variable* deklarasi ADC membuat program *delay* agar tegangan masukan lebih stabil, dengan menuliskan :

```
delay_us(10);
```

Agar LCD dapat menampilkan nilai ADC menuliskan deklarasi *variable* :

```
char lcd_buffer[33];
```

Menampilkan kata-kata pembuka pada LCD, didalam module inisialisasi LCD ditulis:

```
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf(" KENDALI POSISI    PANEL SURYA");
```

Sehingga pada LCD akan tampil tulisan seperti gambar dibawah :



Gambar 4.17. Tampilan Pembuka LCD

Dalam *global interrupt* fungsi assembly menuliskan program untuk pwm, yaitu :

```
pwm=1;
```

Dalam fungsi **while**, memanggil fungsi *keypad*. *Software* codevision avr, tidak memiliki *library* fungsi untuk *keypad*, sehingga menambahkan fungsi keypad.h kedalam file **inc** codevision avr dengan nama keypad.h. Fungsi keypad.h ini disebut juga fungsi *compiler header* untuk *keypad*. *Listing* program keypad.h ini adalah sebagai berikut :

```
#ifndef keypad_INCLUDED_
#define keypad_INCLUDED_

void Push_0catch(void);
void Push_1catch(void);
void Push_2catch(void);
void Push_3catch(void);
```

```
void Push_4catch(void);
void Push_5catch(void);
void Push_6catch(void);
void Push_7catch(void);
void Push_8catch(void);
void Push_9catch(void);

void Push_1pulcatch(void);
void Push_2pulcatch(void);
void Push_3pulcatch(void);
void Push_4pulcatch(void);
void Push_5pulcatch(void);
void Push_6pulcatch(void);
void Push_7pulcatch(void);
void Push_8pulcatch(void);
void Push_9pulcatch(void);

void Push_0prcs(void);
void Push_1prcs(void);
void Push_2prcs(void);
void Push_3prcs(void);
void Push_4prcs(void);
void Push_5prcs(void);
void Push_6prcs(void);
void Push_7prcs(void);
void Push_8prcs(void);
void Push_9prcs(void);
void Push_10prcs(void);
void Push_11prcs(void);
void Push_12prcs(void);

void Push_bintang_prcs(void);
void Push_pagar_prcs(void);
void Push_A_prcs(void);
void Push_B_prcs(void);
void Push_C_prcs(void);
void Push_D_prcs(void);

void Push_0dsp(void);
void Push_1dsp(void);
void Push_2dsp(void);
void Push_3dsp(void);
void Push_4dsp(void);
void Push_5dsp(void);
void Push_6dsp(void);
void Push_7dsp(void);
void Push_8dsp(void);
void Push_9dsp(void);
void Push_10dsp(void);
void Push_11dsp(void);
void Push_12dsp(void);
void Push_bintang_dsp(void);
void Push_pagar_dsp(void);
void Push_A_dsp(void);
void Push_B_dsp(void);
void Push_C_dsp(void);
void Push_D_dsp(void);
```

```

void Keypad(void)
{

unsigned char dt;

PORTC=0xFF;
DDRC=0xF0;

PORTC.4=0;

dt=((~PINC)&0x0F);
switch(dt)
{
case 1 : Push_1catch();
break;

case 2 : Push_4catch();
break;

case 4 : Push_7catch();
break;

case 8 : Push_bintang_prca();
break;

};

PORTC.4=1;

PORTC.5=0;

dt=((~PINC)&0x0F);
switch(dt)
{
case 1 : Push_2catch();
break;

case 2 : Push_5catch();
break;

case 4 : Push_8catch();
break;

case 8 : Push_0catch();
break;

};

PORTC.5=1;

PORTC.6=0;

dt=((~PINC)&0x0F);
switch(dt)
{
case 1 : Push_3catch();
break;

```

```

case 2 : Push_6catch();
break;

case 4 : Push_9catch();
break;

case 8 : Push_pagar_prccs();
break;

};

PORTC.6=1;

PORTC.7=0;

dt=( (~PINC) &0x0F);
switch(dt)
{
case 1 : Push_A_prccs();
break;

case 2 : Push_B_prccs();
break;

case 4 : Push_C_prccs();
break;

case 8 : Push_D_prccs();
break;

};

PORTC.7=1;
}
#endif

```

Untuk *listing* deteksi *keypad* menggunakan fungsi **void**, harus sesuai dengan *compiler header keypad.h*. Setiap tombol pada *keypad* akan menjalankan fungsi-fungsi tertentu yang sudah dituliskan.

Sebagai contoh untuk menampilkan nilai ADC menggunakan tombol A, *listing* programnya adalah sebagai berikut :

```

void Push_A_prccs(void)
{
  lcd_clear();
  lcd_gotoxy(0,0);
  sprintf(lcd_buffer,"adc = %u ",read_adc(2));
  lcd_putsf(lcd_buffer);
}

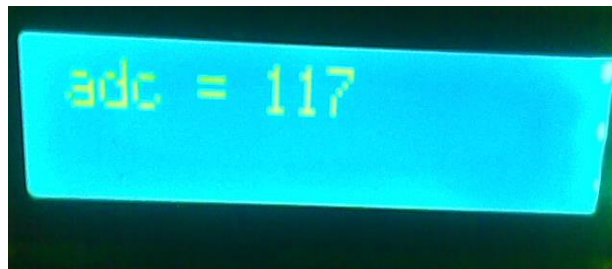
```



```
}

```

Potensiometer terhubung dengan *Port A.2*. Nilai ADC dari potensiometer akan tampil ketika tombol A ditekan. Nilai ADC ini akan berbeda-beda dari 0 – 255 sebatas putaran 300⁰ dari potensiometer. Sebagai contoh LCD menampilkan nilai ADC 117 seperti pada gambar berikut :



Gambar 4.18. Menampilkan Nilai ADC di LCD

Begitu juga dalam membuat menu-menu program, dapat ditampilkan pada LCD. Untuk perhitungan Sudut, potensiometer dapat berputar 300⁰ dengan nilai ADC yang dihasilkan adalah 0 – 255. Dari data yang diperoleh pada pengukuran perbandingan nilai ADC dan sudut, serta penentuan waktu menurut sudut dibuat *listing* program untuk menampilkan nilai ADC, sudut dan waktu yaitu :

```
delay_ms(100);
data_adc = read_adc(2);
if (data_adc <= 20) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut = 180");
} else if ((data_adc >= 20) && (data_adc <= 36)) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=150 16.00");
} else if ((data_adc >= 36) && (data_adc <= 56) ) {
    lcd_clear();

```

```

        lcd_gotoxy(0,0);
        sprintf(lcd_buffer,"adc = %u ",read_adc(2));
        lcd_puts(lcd_buffer);
        lcd_gotoxy(0,1);
        lcd_putsf("sudut=135  15.00");
} else if ( (data_adc >= 56) && (data_adc <= 76) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=120  14.00");
} else if ( (data_adc >= 76) && (data_adc <= 96) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=105  13.00");
} else if ( (data_adc >= 96) && (data_adc <= 120) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=90   12.00");
} else if ( (data_adc >= 120) && (data_adc <= 140) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=75   11.00");
} else if ( (data_adc >= 140) && (data_adc <= 160) ) {

    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=60   10.00");
} else if ( (data_adc >= 160) && (data_adc <= 180) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=45   09.00");
} else if ( (data_adc >= 180) && (data_adc <= 210) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));
    lcd_puts(lcd_buffer);
    lcd_gotoxy(0,1);
    lcd_putsf("sudut=30   08.00");
} else if ( (data_adc >= 210) && (data_adc <= 220) ) {
    lcd_clear();
    lcd_gotoxy(0,0);
    sprintf(lcd_buffer,"adc = %u ",read_adc(2));

```

```

        lcd_puts (lcd_buffer);
        lcd_gotoxy(0,1);
        lcd_putsf("sudut=15    07.00");
    } else if ( (data_adc >= 220) && (data_adc <= 255) ) {
        lcd_clear();
        lcd_gotoxy(0,0);

        sprintf(lcd_buffer,"adc = %u ",read_adc(2));
        lcd_puts (lcd_buffer);
        lcd_gotoxy(0,1);
        lcd_putsf("sudut=0    06.00");
    }
}

```

Hasil pengambilan data penentuan sudut posisi :

Tabel 4.10. Hasil Data Penentuan Sudut Posisi

Keypad	Sudut									
	1	2	3	4	5	6	7	8	9	10
1	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰
2	30 ⁰	30 ⁰	45 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰	30 ⁰
3	60 ⁰	30 ⁰	30 ⁰	60 ⁰	30 ⁰	30 ⁰	30 ⁰	60 ⁰	30 ⁰	30 ⁰
4	75 ⁰	30 ⁰	75 ⁰	75 ⁰	60 ⁰	30 ⁰	75 ⁰	30 ⁰	75 ⁰	75 ⁰
5	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰	90 ⁰
6	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰	105 ⁰
7	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰	120 ⁰
8	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰	135 ⁰

Pada pengambilan data penentuan posisi sudut, dilakukan sampling 10 kali untuk masing-masing sudut yaitu, 30⁰, 45⁰, 60⁰, 75⁰, 90⁰, 105⁰, 120⁰, dan 135⁰. Dari data yang diperoleh, untuk menentukan sudut 30⁰, 90⁰, 105⁰, 120⁰, serta 135⁰ diperoleh keberhasilan 100%, kemudian untuk sudut 75⁰ tingkat keberhasilan adalah 60%, sedangkan untuk sudut 60⁰ dan 45⁰, masing-masing adalah 30% dan 10%. Analisa penyebab tersebut yaitu adanya pembebanan pada motor servo, ketika motor bergerak dan saat akan berhenti ada gaya dorong yang menyebabkan servo motor tidak tepat berhenti di sudut yang diinginkan.

Simpangan Baku

Simpangan baku atau deviasi standar adalah ukuran sebaran statistik yang paling lazim. Singkatnya standar deviasi ini mengukur bagaimana nilai-nilai data tersebar. Simpangan baku didefinisikan sebagai akar kuadrat varians. Simpangan baku merupakan bilangan tak-negatif dan memiliki satuan yang sama dengan data.

Dari hasil pengukuran sudut terhadap nilai ADC pada tugas akhir ini dilakukan sampling sebanyak 10 kali, sehingga dapat dicari simpangan baku dari pengambilan data tersebut untuk mengetahui tingkat simpangan dari data yang terukur, semakin besar nilai nilai simpangan baku ini maka semakin besar pula nilai kesalahan hasil pengukuran data. Begitu juga sebaliknya.

Untuk mencari nilai simpangan baku ini digunakan rumus :

$$S = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}} \dots\dots\dots (1)$$

dimana,

S = Simpangan baku

X_i = Nilai X ke-i

\bar{X} = Nilai rata-rata X

n = Banyaknya jumlah *sample*

Dari pengukuran yang dilakukan sebanyak 10 kali untuk pengukuran ADC terhadap sudut diperoleh data seperti pada tabel 4.6. Dengan menggunakan program microsoft excel diperoleh :

Tabel 4.11. Simpangan Baku

Sudut	ADC										Standar Deviasi
	1	2	3	4	5	6	7	8	9	10	
30 ⁰	194	192	193	198	193	194	192	193	192	195	1.837873167
45 ⁰	173	173	174	173	171	172	173	173	174	173	0.875595036
60 ⁰	153	153	160	154	155	152	156	157	157	152	2.601281735
75 ⁰	132	132	131	132	135	132	129	128	133	128	2.250925735
90 ⁰	109	111	112	111	112	112	108	110	110	110	1.354006401
105 ⁰	87	88	90	88	87	90	89	86	84	87	1.837873167
120 ⁰	64	65	65	67	64	65	66	65	62	63	1.429840706
135 ⁰	46	42	46	48	47	48	47	44	45	45	1.87379591
150 ⁰	26	27	27	28	27	28	29	27	27	26	0.918936583

Kesalahan nilai ADC terhadap sudut posisi dihitung melalui rumus simpangan baku yang dilakukan *sample* sebanyak 10 kali. Dari data tersebut nilai simpangan baku memiliki range antara 0.875595036 - 2.601281735 yaitu pada nilai-nilai ADC untuk sudut 45⁰ dan 60⁰. Pada masing-masing sudut dengan nilai rata-rata simpangan baku 1.664458716.