

EMDEDEDDED ARRAY SENSOR UNTUK LINE FOLLOWING ROBOT

Akhmad Hendriawan

Jurusan Teknik Elektronika

Politeknik Elektronika Negeri Surabaya

Kampus ITS, Keputih Sukolilo Surabaya 60111

Telp. (+62) 31- 5947280 Fax (+62) 31- 5946114

E-mail: hendri@eepis-its.edu

ABSTRAK

Line following robot adalah sebuah robot yang bergerak secara otomatis mengikuti garis. Umumnya setiap kali akan melakukan running, pengaturan sensor pada linefollowing robot pada kondisi pencahayaan dan warna garis yang berbeda perlu dilakukan. Pengaturan sensor robot menggunakan potensiometer akan membutuhkan banyak waktu dan dapat mengurangi lamanya waktu pertandingan. Permasalahan yang lain adalah jalan robot tidak halus. Untuk mengatasi permasalahan tersebut dibuat suatu *sistem embedded array sensor* dengan menggunakan enam buah sensor photodiode. Pada *sistem embedded array sensor* digunakan AVR mikrokontroler yang dilengkapi dengan fasilitas ADC dan PWM internal. ADC digunakan untuk mengkonversi data analog dari sensor menjadi data digital. Pada kontroler robot digunakan suatu metode pentabelan kondisi sensor yang menghasilkan keluaran data PWM untuk mengatur kecepatan motor kanan dan motor kiri robot. Dengan adanya *sistem embedded array sensor* dihasilkan suatu robot yang dapat berjalan pada berbagai macam warna garis dengan background yang lebih gelap dari garis tanpa harus melakukan pengaturan secara manual menggunakan potensiometer dan tidak mudah terpengaruh oleh cahaya luar serta dapat berjalan secara halus.

Kata kunci: ADC, AVR mikrokontroler, metode pentabelan.

Pendahuluan

Pengaturan sensor dari suatu line following robot perlu dilakukan pada kondisi pencahayaan dan warna garis yang berbeda agar robot dapat berjalan mengikuti garis dengan baik. Metode pengaturan secara manual dengan menggunakan komparator telah berhasil dilakukan. Pada metode ini pengaturan dilakukan dengan cara mengubah nilai dari variabel resistor pada rangkaian komparator agar didapatkan suatu logika sensor '0' atau '1'. Pengaturan secara manual dengan menggunakan komponen variabel resistor ini memerlukan banyak waktu dan komponen yang lebih banyak

Pada paper ini disajikan metode pengaturan, untuk menghasilkan suatu robot yang secara otomatis mengikuti garis tanpa harus melakukan pengaturan secara manual menggunakan variabel resistor. Pengaturan dilakukan dengan memberikan fasilitas kalibrasi sensor pada robot. Kalibrasi dilakukan dengan cara mengambil data sensor pada garis dan mengambil data sensor pada *background* dari lintasan robot yang akan digunakan, kemudian pada setiap sensor dilakukan proses rata-rata antara data garis dan data *background*. Saat robot mulai melakukan start, data hasil kalibrasi yang sudah mengalami proses rata-rata dibandingkan dengan data pada setiap sensor sejumlah enam buah sensor. Hasil perbandingan antara data hasil rata-rata dengan data *current sensor* kemudian di ubah menjadi

data kecepatan motor dengan menggunakan metode pentabelan. Data hasil pentabelan ini digunakan untuk menggerakkan motor kiri dan motor kanan dengan menggunakan PWM controller.

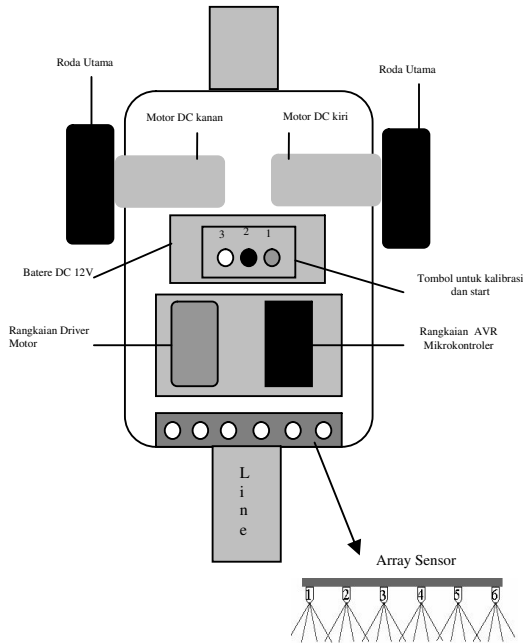
Perbedaan dengan metode pertama adalah pada metode kedua proses pengaturan dilakukan dengan cara pengambilan data analog sensor. Proses pengambilan data sensor analog mempunyai keuntungan hanya membutuhkan waktu beberapa detik untuk pengaturan sensor dan rangkaian yang digunakan sangat sedikit karena tidak memerlukan rangkaian komparator dan tidak memerlukan potensiometer.

Perancangan Sistem

- Array Sensor

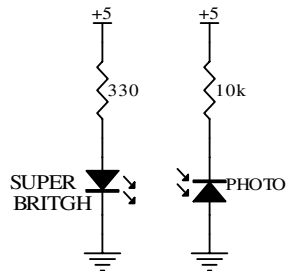
Array sensor adalah urutan sensor yang terdiri dari enam buah sensor photodiode sebagai *receiver* dan enam buah LED super bright sebagai *transmitter* [4]. Bentuk dan penempatan sensor pada robot dapat dilihat pada gambar 1. Peletakan sensor pada bagian depan robot dimaksudkan agar robot dapat berjalan secara dinamis mengikuti garis [1]. *Array sensor* digunakan untuk mengetahui posisi robot terhadap garis. Cara kerja dari *array sensor* adalah menghasilkan tegangan analog yang digunakan sebagai masukan *ADC*. Besar nilai tegangan analog yang dihasilkan setiap sensor digunakan sebagai acuan untuk mengetahui posisi masing-masing sensor terhadap garis. Jika nilai tegangan analog yang dihasilkan oleh sensor mendekati tegangan Vcc, berarti

posisi sensor berada pada garis dan jika nilai tegangan analog sensor mendekati 0 berarti posisi sensor berada diluar garis.



Gambar 1. Mekanik robot tampak atas

Rangkaian masing-masing sensor sensor seperti pada gambar 2.



Gambar 2. Rangkaian sensor photodiode

- ADC (Analog to Digital Converter)

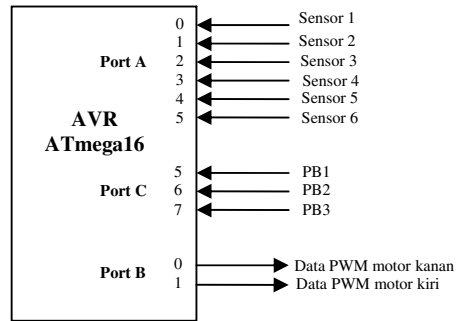
Komponen *ADC* digunakan untuk mengkonversi tegangan analog dari sensor menjadi data digital untuk selanjutnya diproses oleh mikrokontroler. *ADC* yang digunakan adalah *ADC internal* pada AVR mikrokontroler. *ADC* pada AVR mikrokontroler mempunyai 8 buah *channel* input dengan resolusi data untuk masing-masing *channel* sebesar 10-bit.

Enam buah sensor dihubungkan ke enam buah *channel* *ADC* pada mikrokontroler AVR. Proses pengambilan data pada array sensor dilakukan secara berurutan mulai dari *channel* 0 sampai dengan *channel* 5. Data hasil *konversi* *ADC* dapat diambil pada register `adc_data[i]` (*i* menunjukkan *ADC channel*). Data hasil *konversi* yang dihasilkan oleh *ADC* merupakan hasil

pembagian antara tegangan input dengan tegangan referensi dikalikan dengan 8-bit data / 255 (karena hanya menggunakan resolusi 8-bit).

- Mikrokontroler

Mikrokontroler yang digunakan adalah AVR ATmega16 yang mempunyai fasilitas *ADC* dan *PWM* internal [2]. AVR ATmega16 digunakan sebagai pemroses utama *line following* robot. Mikrokontroler AVR menerima input dari enam buah sensor garis dan menghasilkan output berupa *controller* *PWM* yang dihubungkan dengan dua buah motor DC sebagai pengatur pergerakan dan kecepatan motor kanan dan motor kiri. Penggunaan pin I/O AVR ATmega16 ditunjukkan pada gambar 3.



Gambar 3. Koneksi pin I/O AVR mikrokontroler dengan peralatan luar

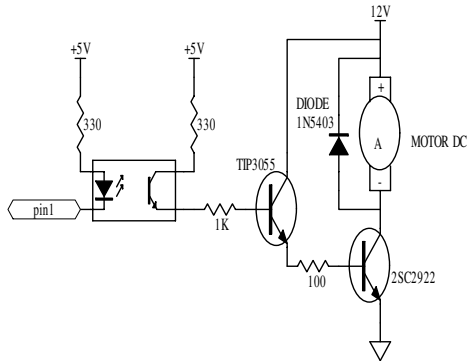
Berdasarkan gambar 3, dapat diketahui jumlah pin I/O yang dibutuhkan dalam membuat suatu sistem *embedded array sensor* pada *line following* robot. Port A bit 0 sampai dengan bit 5 digunakan sebagai *ADC channel* dari AVR ATmega16 yang dihubungkan dengan array sensor.

- Driver Motor

Rangkaian driver motor ditunjukkan pada gambar 4. Rangkaian driver motor menggunakan rangkaian *darlington* dari transistor. Penggunaan rangkaian driver motor adalah untuk memenuhi kebutuhan arus motor DC yang tidak bisa dipenuhi oleh mikrokontroler, karena mikrokontroler hanya mengeluarkan arus keluaran yang relatif kecil. Besar penguatan rangkaian driver motor merupakan perkalian antara transistor TIP3055 dengan penguatan transistor 2SC2922. Besar arus atau tegangan keluaran yang dihasilkan oleh rangkaian driver motor dibatasi oleh arus atau tegangan *Vcc* [1]. Port B bit 0 dan Port B bit 1 dihubungkan dengan rangkaian *driver motor* dan digunakan sebagai output data *PWM* untuk mengatur kecepatan motor kanan dan motor kiri.

- PWM kontroler

Prinsip kerja *PWM* adalah mengatur lebar pulsa on dan off dari sinyal yang mempunyai periode tertentu. Hubungan antara T_{high} dan T_{low} dinyatakan dengan *duty cycle*. Pembangkit *PWM* memanfaatkan fasilitas *interrupt timer* yang dimiliki oleh AVR mikrokontroler.



Gambar 4. rangkaian driver motor

Data maksimal yang digunakan untuk pengaturan PWM sebesar 25, Penentuan nilai data PWM tersebut berdasarkan pemilihan frekuensi PWM sebesar 200 Hz. Data PWM untuk mengatur kecepatan motor terdapat pada suatu variabel `pwmR` untuk data kecepatan motor kanan dan `pwmL` untuk data kecepatan motor kiri. Program untuk membentuk sinyal PWM adalah sebagai berikut :

```
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Place your code here
    r=r+1;
    if(r==25) //
    {
        r=0;
        PORTB.0=0; // motor kiri
        PORTB.1=0; // motor kanan
    }
    if(r==pwmL)PORTB.0=1;
    if(r==pwmR)PORTB.1=1;
}
```

- Kalibrasi

Proses kalibrasi bertujuan untuk mendapatkan nilai tengah dari masing-masing sensor. Port C dihubungkan dengan tombol *push button* yang terdiri dari PB1 sebagai tombol proses *learning* pada garis, PB2 sebagai tombol *learning* pada background dan PB3 sebagai tombol *start*.

Data kalibrasi akan digunakan sebagai data pembandingan dengan data current sensor pada saat *running*. Data kalibrasi dapat diperoleh dengan cara menjumlahkan data pada saat sensor berada di dalam garis dengan data sensor diluar garis dibagi dua. Proses kalibrasi ini dapat dirumuskan sebagai berikut :

$$\text{Nilai_tengah} = (\text{line} + \text{no_line}) / 2$$

Pengambilan data pada saat kalibrasi disimpan pada RAM dengan menggunakan variabel *white* untuk penyimpanan data pada saat sensor berada di dalam garis, variabel *black* untuk penyimpanan data pada saat sensor berada di luar garis dan variabel *rate* yang merupakan nilai rata-rata kedua data hasil kalibrasi Software proses kalibrasi adalah sebagai berikut :

```
// Global enable interrupts
#asm("sei")
while (1)
{
    // Place your code here
    if(PINC.7==0)
    {
        for(i=0;i<6;i++)
        {
            white[i]=adc_data[i];
            PORTD=white[i];
        }
    }
    else if(PINC.6==0) //
    {
        for(i=0;i<6;i++)
        {
            black[i]=adc_data[i];
            PORTD=black[i];
        }
    }
    else if(PINC.5==0) // tombol start
    {
        for(i=0;i<6;i++)
        {
            rate[i]=(white[i]+black[i])/2;
        }
    }
    else
        PORTD=0x00;
}
```

Jika tombol PB1 ditekan maka dilakukan proses pengambilan data sensor pada garis, jika tombol PB2 ditekan maka dilakukan proses pengambilan data sensor diluar garis (lapangan) dan jika tombol PB3 ditekan maka pada led akan tampak data hasil kalibrasi untuk masing-masing sensor. Proses kalibrasi ini diambil proses kalibrasi pada lima macam garis yang berbeda (putih, kuning, hijau, biru, merah) dengan background yang sama (hitam). Data hasil kalibrasi dapat dilihat pada tabel 1 :

Tabel 1 Data hasil kalibrasi

Warna line	Sensor	No line	Rate	Line
Putih	S1	1A	3D	60
	S2	13	35	58
	S3	2C	68	A5
	S4	29	5C	90
	S5	21	56	8C
Kuning	S6	25	55	86
	S1	1B	3F	63
	S2	13	3B	63
	S3	2E	6F	B0
	S4	2B	5D	8F
	S5	21	5B	96

	S6	26	5C	92
Biru	S1	1B	27	33
	S2	13	23	34
	S3	2E	3C	4A
	S4	2A	3A	4A
	S5	20	26	2C
	S6	26	37	48
Merah	S1	1C	3D	5E
	S2	12	32	52
	S3	2E	72	B7
	S4	2A	56	82
	S5	21	4D	7A
	S6	25	47	69
Hijau	S1	15	24	33
	S2	12	1D	28
	S3	2D	3E	50
	S4	24	2F	3B
	S5	21	23	26
	S6	25	36	47

Berdasarkan data pada table diatas dapat diketahui besar data masing-masing sensor pada saat berada pada garis dan diluar garis serta data hasil rata-rata untuk masing-masing sensor.

- Metode Pentabelan

Metode pentabelan dilakukan dengan cara mengambil beberapa pengkondisian array sensor pada robot terhadap garis. Banyaknya pengkondisian sensor yang dipakai sebanyak 15 kondisi. Pengkondisian berdasarkan posisi sensor mulai dari sisi kanan yaitu s1, s2, s3, s4, s5 dan s6, lebih jelas dapat dilihat pada gambar 1. Pengkondisian sensor menghasilkan keluaran data PWM untuk mengatur kecepatan motor kanan dan motor kiri robot. Data pentabelan sensor ditunjukkan pada tabel 2.

Tabel 2. Pengkondisian Sensor

No	Kondisi Sensor						Output Data PWM	
	s1	s2	s3	s4	s5	s6	Motor Kanan	Motor Kiri
1	1	0	0	0	0	0	3	7
2	1	1	0	0	0	0	4	8
3	0	1	0	0	0	0	5	8
4	0	1	1	0	0	0	6	8
5	0	0	1	0	0	0	7	8
6	0	0	1	1	0	0	10	10
7	0	0	0	1	0	0	8	7
8	0	0	0	1	1	0	8	6
9	0	0	0	0	1	0	8	5
10	0	0	0	0	1	1	8	4
11	0	0	0	0	0	1	7	3
12	1	1	1	0	0	0	3	7
13	0	1	1	1	0	0	5	7

14	0	0	1	1	1	0	7	5
15	0	0	0	1	1	1	7	3

Pengujian dan Analisa

Pengujian yang dilakukan adalah sebagai berikut:

a. Pengujian Sensor dan ADC

Pengujian sensor dilakukan dengan cara menghubungkan sensor garis dengan ADC yang sudah ada pada AVR mikrokontroler. Hasil pembacaan oleh mikrokontroler ini ditampilkan di LED sebanyak 8 buah yang dihubungkan dengan PORTC pada mikrokontroler yang sama. Dari hasil yang didapatkan pada tabel 3, diketahui bahwa sensor bekerja dengan baik, pada waktu yang sama didapatkan hasil yang tidak sama pada setiap sensor.

Tabel 3. Hasil tampilan data pada PORTC

Sensor	Data pada line(putih)	Data no_line(hitam)
1	60h	1Ch
2	0C2h	3fh
3	B0h	26h
4	0B8h	24h
5	0A0h	20h
6	80h	1Ah

b. Pengujian Intensitas Cahaya

Pengujian intensitas cahaya bertujuan untuk mengetahui pengaruh intensitas cahaya terhadap tingkat keberhasilan jalan robot terhadap garis. Pengujian dilakukan dengan cara memberikan suatu kondisi pencahayaan dengan intensitas 0, 100, 500, 1000, 1500 dan 2000 (diukur menggunakan luxmeter dalam satuan lux) pada saat robot akan mulai berjalan maupun pada saat robot sedang berjalan pada garis.

Berdasarkan percobaan yang dilakukan didapatkan suatu data tingkat keberhasilan robot terhadap intensitas cahaya yang diberikan seperti pada tabel 4.

Tabel 4. Tingkat keberhasilan robot terhadap intensitas cahaya

No	Intensitas Cahaya(lux)	Keterangan 10 kali percobaan	
		Berhasil	Gagal
1	0	10	0
2	100	10	0
3	500	10	0
4	1000	10	0
5	1500	9	1
6	2000	9	1

Berdasarkan data pada tabel 3 dapat diketahui bahwa intensitas cahaya yang diberikan tidak begitu mempengaruhi jalan robot karena terdapat fasilitas *learning* yang menghasilkan tegangan *threshol* sensor. Pada sensor robot juga diberikan penutup yang sangat

rapat sehingga cahaya luar hampir tidak dapat mempengaruhi kerja dari sensor dan sensor dapat bekerja dengan baik tanpa pengaruh cahaya luar.

c. Pengujian terhadap warna garis

Pengujian terhadap warna garis dilakukan bertujuan untuk mengetahui seberapa besar nilai perbedaan *grayscale* antara warna garis dengan warna background. Warna background adalah hitam. Pengujian dilakukan dengan cara menjalankan robot pada lintasan dengan warna garis merah, kuning, hijau, biru dan putih serta background warna hitam. Sebelum robot dijalankan terlebih dahulu dilakukan proses *learning* terhadap garis dan background.

Berdasarkan percobaan yang dilakukan didapatkan suatu data tingkat keberhasilan robot terhadap berbagai macam warna garis seperti pada tabel 5 :

Tabel 5. Tingkat keberhasilan robot terhadap warna garis dengan background warna hitam

No	Warna Garis	Data ADC	Keterangan 10 kali percobaan	
			Berhasil	Gagal
1	Putih	255	10	0
2	Merah	145	10	0
3	Kuning	195	10	0
4	Hijau	105	9	1
5	Biru	50	8	2

Berdasarkan data pada tabel 4 dapat diketahui bahwa semakin besar perbedaan nilai data ADC antara garis dengan background (hitam) maka tingkat keberhasilan robot dalam mengikuti garis menjadi semakin tinggi. Begitu juga sebaliknya semakin kecil perbedaan nilai nilai data antara garis dengan background maka tingkat keberhasilan robot dalam mengikuti garis juga semakin kecil. Tingkat keberhasilan terjadi karena besarnya nilai *grayscale* sangat menentukan besarnya data yang dihasilkan oleh sensor.

d. Pengujian sudut belokan

Pengujian terhadap sudut dilakukan bertujuan untuk mengetahui besar sudut yang dapat dilalui oleh robot tanpa harus keluar dari lintasan garis. Pengujian sudut belokan dilakukan dengan cara mencoba jalan robot pada sudut-sudut yang telah dibuat yaitu sudut 90°, 100°, 110°, 120°, 125°, 130°, 135°, 140°, 145° dan 150°.

Berdasarkan percobaan yang dilakukan didapat suatu data tingkat keberhasilan robot pada saat melewati sudut-sudut tersebut seperti ditunjukkan pada tabel 6.

Berdasarkan data pada tabel 6 dapat diketahui bahwa tingkat keberhasilan robot semakin tinggi jika sudut yang dilalui lebih besar dari sudut 110°.

Tabel 6 Tingkat keberhasilan robot terhadap sudut belokan

No	Sudut Belokan dalam derajat	Keterangan 10 kali percobaan	
		Berhasil	Gagal
1	90	1	9
2	100	2	8
3	110	3	7
4	120	8	2
5	125	9	1
6	130	9	1
7	135	10	0
8	140	10	0
9	145	10	0
10	150	10	0

Kesimpulan

Setelah melakukan perencanaan dan pembuatan sistem kemudian dilakukan pengujian dan analisisnya, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Robot dapat melalui belokan patah dengan baik jika sudut belokan lebih besar dari 110° yaitu 120°, 125°, 130°, 135°, 140°, 145°, 150° dan akan mengalami overshoot jika sudut belokan kurang dari 110° yaitu 90° dan 100°.
2. Robot dapat berjalan pada berbagai macam kondisi pencahayaan karena sensor tertutup rapat sehingga tidak terpengaruh oleh cahaya luar.

Daftar Pustaka

[1] Fred G.Martin, "Robotic Explorations: a hand-on introduction to engineering", Prentice Hall,2001
 [2] www.atmel.com., " Data Sheet AVR ATmega16 ", 15 Desember 2004.
 [3] www.robotroom.com, 12 Januari 2005.
 [4] www.elm-chan.org, 14 Februari 2005

